School of Computing Science,
University of Newcastle upon Tyne

An Algebra of Timed-Arc Petri Nets

A. Niaouris and M. Koutny

Technical Report Series

CS-TR-895

March 2005

Published by the University of Newcastle upon Tyne,
School of Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, UK.

An Algebra of Timed-Arc Petri Nets

Apostolos Niaouris and Maciej Koutny

School of Computing Science, University of Newcastle
Newcastle upon Tyne NE1 7RU, U.K.
email:{apostolos.niaouris,maciej.koutny}@ncl.ac.uk

Abstract. In this paper we present and investigate two algebras, one based on term re-writing and the other on Petri nets, aimed at the specification and analysis of concurrent systems with timing information. The former is based on process expressions (at-expressions) and employs a set of SOS rules providing their operational semantics. The latter is based on a class of Petri nets with time restrictions associated with their arcs, called at-boxes, and the corresponding transition firing rule. We relate the two algebras through a compositionally defined mapping which for a given at-expression returns an at-box with behaviourally equivalent transition system. The resulting model, called the Arc Time Petri Box Calculus (atPBC), extends the existing approach of the Petri Box Calculus (PBC).
Keywords: Net-based algebraic calculi; arc-based time Petri nets; relationships between net theory and other approaches; process algebras; box algebra; SOS semantics.

1 Introduction

Process algebras, e.g., ACP [2], CCS [15] and CSP[9], provide a formal framework for dealing with large and complex concurrent computing systems by employing specific operators corresponding to commonly used programming constructs. The way of representing system's structure is given through suitably defined set of process expressions, and their behaviour is typically captured by a (structured) set of sequences of executed actions. Another way of modelling concurrent systems is provided by Petri nets [16, 21], which support a graphical representation of concurrent systems and, through their being based on a theory of partial orders (capturing explicit asynchrony), an additional means of verifying their correctness efficiently, and a way of expressing properties related to causality and concurrency in system behaviour.

These two kinds of formalisms treat the structure and semantics of concurrent systems in different ways, which in the past meant that it was almost impossible to take full advantage of their relative advantages (i.e., *compositionality* and *explicit asynchrony*) when used in isolation. To a significant extent, this problem was addressed by the Box Algebra [5, 6] and its precursor, the Petri Box Calculus (PBC) [4]. Both models provided a framework where Petri nets and process algebras could co-exist, and thus established a bridge between these two approaches.

Since its conception, the PBC has been extended to cover, in particular, concurrent systems with timing restrictions [12, 13], where the timing restrictions were associated with transitions, effectively specifying for how long an enabled action (or transition) can delay/prolong its execution as well as what is a minimum delay or execution time. Another way in which timing assumption could be introduced is to associate clocks (or age) with the resources (or tokens). More precisely, one can specify how old/young a given resource consumed by an action must be. This approach has been extensively studied in the past, see, e.g., [1, 8, 18], both as a model for dealing with complex concurrent systems such as communication protocols, and as a framework for verifying their properties. It is precisely this kind of time modelling which has been adopted in this paper.

We will introduce and investigate two different models for the specification of concurrent systems including explicit timing information. Both models have an algebraic structure based on operators present in the standard PBC. The first algebra is based on process expressions, called at-expressions, and a system of rewriting rules providing structural operational semantics of at-expressions in the style of [19]. The second algebra is based on a class of Petri nets with arc-based timing restrictions, called at-boxes, and their execution rules. This means, in particular, that: (i) each arc from a place p to a transition is given two time bounds, e and l, representing the *earliest consuming time* and the *latest consuming time*, respectively, for a token which has arrived at place p; (ii) the local clock of a token is started at the very moment it has been created; and (iii) time is discrete. It is important to point out that property (i) suits particularly well the intended compositional setting we are aiming at since the handshake synchronisation of two transitions basically amounts to gluing them together, and no special consideration of their timing restrictions is needed. On the other hand, gluing two transitions in the other time framework we mentioned requires combining their timing intervals which can be done in several different ways.

The two algebras are related through a compositionally defined mapping which, for at-expression returns a corresponding at-box (its denotational semantics). The main result is that the denotational and operational semantics of an at-expression are behaviourally equivalent. The resulting framework, first reported in [17], consisting of two consistent algebras is called the Arc-Based Time Petri Box Calculus, or *atPBC*.

The paper is organised as follows. Section 2 recalls some basic notions used throughout the paper, section 3 introduces at-boxes, section 4 provides the syntax and semantics of at-expressions, and section 5 develops a compositional net model based on at-boxes.

Throughout the paper, we assume that the reader is familiar with the basic concepts of PBC and the Box Algebra [5] on which the compositional treatment of nets is based. In addition, the appendix contains all the relevant definitions and results. It also contains the proofs of the key properties of atPBC, as well as the definitions of other related algebras of expressions and nets.

2 Basic notions

Multisets Throughout the paper, $\mathbb{N}$ denotes the set of non-negative integers and $\mathbb{N}^\infty \stackrel{\text{df}}{=} \mathbb{N} \cup \{\infty\}$. A *multiset* over a set X is a function $\mu : X \to \mathbb{N}$. We will write $\mu \leq \mu'$ if the domain X of μ is included in that of the multiset μ', and $\mu(x) \leq \mu'(x)$, for all $x \in X$. An element $x \in X$ belongs to μ, denoted $x \in \mu$, if $\mu(x) > 0$. The sum and difference of multisets, and the multiplication by a non-negative integer are respectively denoted by $+$, $-$ and $\cdot$ (the difference will only be applied when the second argument is smaller or equal to the first one). A subset of X may be treated as a multiset over X, by identifying it with its characteristic function, and a singleton set can be identified with its sole element. A multiset μ over X may be denoted as

$$\sum_{x \in X} \mu(x) \cdot \{x\}\,,$$

as well as written in extended set notation, e.g., $\{a, a, b\}$ denotes a multiset μ such that, for every $x \in X$,

$$\mu(x) = \begin{cases} 2 & \text{if } x = a \\ 1 & \text{if } x = b \\ 0 & \text{otherwise}\,. \end{cases}$$

Nets A triple (P, T, F) is a *net* if P is a finite set of *places*, T is a finite set of *transitions* disjoint from P, and $F \subseteq (T \times P) \cup (P \times T)$ is a *flow relation.* In what follows, for every $x \in P \cup T$, ${}^\bullet x \stackrel{\text{df}}{=} \{y \mid (y, x) \in F\}$ is the *preset* of x and $x^\bullet \stackrel{\text{df}}{=} \{y \mid (x, y) \in F\}$ is the *postset* of x; we assume that, for any transition x, both sets are always non-empty. The dot-notation extends in the usual way to sets of places and/or transitions.

A tuple $N = (P, T, F, M)$ is a *Place/Transition net* (or PT-net) if (P, T, F) is a net and $M : P \to \mathbb{N}$ is the *initial marking* (in general, any mapping from P to $\mathbb{N}$ is a marking of N.

A finite set of transitions U, called a *step*, is *enabled* at a marking M if, for all $p \in P$,

$$M(p) \geq \sum_{t \in U} F(p, t) \cdot U(t)\ ,$$

where we use the symbol F to denote the characteristic function of F. Such a step may *fire* leading to a *follower* marking M' given, for every place $p \in P$, by

$$M'(p) \stackrel{\text{df}}{=} M(p) - \sum_{t \in U} F(p, t) \cdot U(t) + \sum_{t \in U} F(t, p) \cdot U(t).$$

We denote this by $M[U\rangle M'$, and call M' *reachable* from M (in general, a marking can be reachable through a possibly empty sequence of intermediate markings). The net is *safe* if for every marking M reachable from the initial one, it is the case that $M(p) \leq 1$, for all $p \in P$.

Action labels and time restrictions To label transitions in nets considered in this paper, we use a fixed set of *communication* actions $\mathcal{A}$ such that for every $a \in \mathcal{A}$, there exists its *conjugate*, $\widehat{a} \in \mathcal{A}$, satisfying $a \neq \widehat{a}$ and $\widehat{\widehat{a}} = a$. Also, there is a *silent* (or internal) action $\imath \notin \mathcal{A}$. In the algebra of nets (as well as in the process algebra), it will be assumed that a *synchronisation* of two conjugate communication actions always gives rise to the silent action $\imath$.

To model timing restrictions we use the following notation:

$$\begin{array}{ll} \mathbb{D}^\infty & \stackrel{\text{df}}{=} \{el \mid e \in \mathbb{N} \wedge l \in \mathbb{N}^\infty \wedge e \leq l\} \\ \mathbb{D} & \stackrel{\text{df}}{=} \{el \in \mathbb{D}^\infty \mid l \neq \infty\} \\ \mathbb{D}^\perp & \stackrel{\text{df}}{=} \mathbb{D} \cup \{\perp\} \\ \mathbb{N}^\perp & \stackrel{\text{df}}{=} \mathbb{N} \cup \{\perp\}\,. \end{array}$$

Let $n \in \mathbb{N}$, $\mathbb{EL} \in \mathbb{D}$ and $el \in \mathbb{D}^\infty$. Then n satisfies the timing restriction el if $e \leq n \leq l$, and $\mathbb{EL}$ satisfies the timing restriction el if $e \leq \mathbb{E}$ and $\mathbb{L} \leq l$. We denote this by n tsat el and $\mathbb{EL}$ tsat el, respectively. Moreover, for every pair $\xi, \nu \in \mathbb{D}^\perp$, we denote

$$\xi \oplus \nu \stackrel{\text{df}}{=} \begin{cases} \perp & \text{if } \ \xi = \nu = \perp \\ \mathbb{EL} & \text{if } \ \{\xi,\nu\} = \{\perp, \mathbb{EL}\} \\ \min\{\mathbb{E},\mathbb{E}'\}\max\{\mathbb{L},\mathbb{L}'\} & \text{if } \ \xi = \mathbb{EL} \wedge \nu = \mathbb{E}'\mathbb{L}'\,. \end{cases}$$

3 Boxes with arc-based time restrictions

An *arc-time box* (or at-box) is a tuple $\mathbf{\Theta} \stackrel{\text{df}}{=} (P, T, F, \lambda, \mu)$ such that:[1]

- P, T and F are as in the definition of a PT-net.
- λ is a mapping with the domain $P \cup T \cup ((P \times T) \cap F)$. For every place $p \in P$ and transition $t \in T$, we have the following: $\lambda(p)$ is a symbol in $\{\mathsf{e}, \mathsf{i}, \mathsf{x}\}$; $\lambda(t)$ is an action in $\mathcal{A} \cup \{\imath\}$; and if $(p,t) \in F$ then $\lambda(p,t) \in \mathbb{D}^\infty$.
- $\mu : P \to \mathbb{N}^\perp$ is the *initial* token timing mapping of $\mathbf{\Theta}$ (in general, any such mapping is a token timing of $\mathbf{\Theta}$).

Note that token timing mappings of at-boxes are interpreted differently from markings of PT-nets, namely, $\mu(p) = k$ means that p holds a single token which is k units of time old, and $\mu(p) = \perp$ means that p is empty.

We adopt the standard rules concerning the drawing of diagrams. In the diagrams, the empty local state $\perp$ will not be represented, and otherwise $\mu(p)$ will be displayed. Other drawing conventions are the same as for the standard Petri nets.

[1] To improve the readability of the main part of the paper, we have slightly simplified the definition of at-boxes and other related notions, which are given in full in the appendix.

The 'time-less' version of $\mathbf{\Theta}$ is defined as a PT-net $[\![\mathbf{\Theta}]\!] \stackrel{\mathrm{df}}{=} (P, T, F, [\![\mu]\!])$ such that, for every $p \in P$,

$$[\![\mu]\!](p) \stackrel{\mathrm{df}}{=} \begin{cases} 1 & \text{if } \mu(p) \in \mathbb{N} \\ 0 & \text{if } \mu(p) = \bot\,. \end{cases}$$

In what follows, $[\![\mathbf{\Theta}]\!]$ will be called the *underlying* net of $\mathbf{\Theta}$, and we will assume that it is always safe.

In the at-box model, time restrictions are associated with the arcs incoming to transitions. For example, if $\lambda(p,t) = (e,l)$, then the interval (e,l) gives the waiting time for the tokens flowing from place p to transition t. This interval identifies the time for which a token has to wait in place p before it can be used to fire transition t on this occasion. The left bound, e, is called the *minimum* waiting time and the right bound, l, the *maximum* waiting time. A token on p cannot be used to fire t when it is younger than the minimum waiting time and must be used to fire an enabled transition before the maximum waiting time has finished (unless the transition has been disabled in the meantime). If t is not enabled and the maximum waiting time has passed, the token can no longer be used to fire transition t. The age of tokens is represented through a token timing which returns, for each place containing a token, its age ($\bot$ is returned if a given place is empty). When a token arrives to a place, its age is set to zero. After that the age can be increased due to the passage of time. It should be emphasized that a token does not need to enable any transition in order for its clock to start 'ticking'.

A finite set of transitions $U = \{t_1, \ldots, t_k\}$, called a *step*, is *enabled* by a token timing μ if it is enabled at the marking $[\![\mu]\!]$ in the safe underlying PT-net and, moreover, if $t \in U$ and $p \in {}^\bullet t$ then $\mu(p)$ tsat $\lambda(p,t)$. Such a step may *fire* leading to a *follower* token timing ν such that, for every place $p \in P$,

$$\nu(p) \stackrel{\mathrm{df}}{=} \begin{cases} \bot & \text{if } p \in {}^\bullet U \setminus U^\bullet \\ 0 & \text{if } p \in U^\bullet \\ \mu(p) & \text{otherwise}\ . \end{cases}$$

We denote this by $\mu[U\rangle\nu$.

Another kind of dynamic changes is effected by time moves. A token timing μ can change into token timing ν by the passage of one time unit if, for transition t enabled at μ and for every place $p \in {}^\bullet t$ we have $\mu(p) < l$, where $el = \lambda(p,t)$. The change results in a new token timing ν such that, for every place $p \in P$,

$$\nu(p) \stackrel{\mathrm{df}}{=} \begin{cases} \mu(p) + 1 & \text{if } \mu(p) \in \mathbb{N} \\ \mu(p) & \text{otherwise}\ . \end{cases}$$

We denote this by $\mu[\surd\rangle\nu$. Intuitively, at-boxes' time deadlines are assumed to be *hard*, i.e., when a transition is ready to fire and even if only one of its input tokens has reached the maximum waiting time, then this transition must fire (or become disabled) before further passage of time.

The overall behaviour of $\mathbf{\Theta}$ is captured by its *reachability tree* with nodes labelled by token timings and arcs annotated by labelled moves, denoted by

$\mathsf{RT}_{\mathbf{\Theta}}$. More precisely, the root node is labelled by the initial token timing and, if a node is labelled by μ, then for every move $\mu[x\rangle\nu$ there is a unique descendant labelled by ν; the arc leading to it is labelled by $\surd$ if $x = \surd$, and by the multiset of communication labels

$$\lambda(U) \stackrel{\mathrm{df}}{=} \sum_{t \in U} U(t) \cdot \{\lambda(t)\}$$

if $x = U$ is an executed transition step. Figure 1 shows an at-box $\mathbf{\Theta}$ and the corresponding reachability tree $\mathsf{RT}_{\mathbf{\Theta}}$. The use of reachability trees instead of reachability graphs may be quite surprising at the moment but will be explained later in this paper together with the considerations that led to this decision.

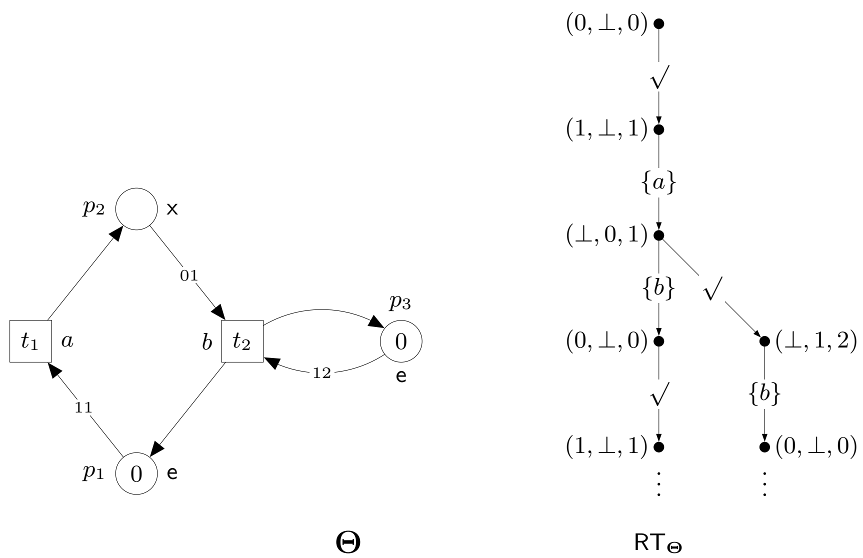


Fig. 1. An at-box $\mathbf{\Theta}$ and a part of its reachability tree $\mathsf{RT}_{\mathbf{\Theta}}$.

4 An algebra of process expressions

In this section, we define the syntax and then operational semantics of process expressions corresponding to at-boxes.

4.1 Static at-expressions

The following is the syntax for the *static arc-based time box expressions* (or static at-expressions), E, which correspond to at-boxes without tokens (below A is a finite subset of $\mathcal{A}$, and Z is an auxiliary subset of static expressions which is needed to ensure that the nets corresponding to at-expressions are safe).

$$\begin{aligned} E &::= \alpha el \quad \mid E \,\mathsf{sc}\, A \mid E \Box E \mid E \| E \mid E \,\mathbf{;}\, E \mid \langle\!\langle E \circledast Z \circledast E \rangle\!\rangle \\ Z &::= Z \,\mathsf{sc}\, A \mid Z \Box Z \quad \mid E \,\mathbf{;}\, E \mid \langle\!\langle E \circledast Z \circledast E \rangle\!\rangle \;. \end{aligned} \tag{1}$$

The only real modification, when compared with the standard PBC syntax, is that a different type of constant expression is used, viz. αel where $\alpha \in \mathcal{A} \cup \{\imath\}$ is a basic action $el \in \mathbb{D}^\infty$ is a timing restriction. Moreover, the actions employed by the syntax allow two-way rather than multi-way synchronisation. Similarly as in the case of at-boxes, e denotes the *minimum*, and l the *maximum* waiting time.

Sequence $E\,\mathbf{;}\,F$ and choice $E \Box F$ compositions are standard; the $\Box$ is used to denote what is essentially the + in CCS [15] and the comma (,) in COSY [10]. The iterative construct $\langle\!\langle D \circledast E \circledast F \rangle\!\rangle$ means 'perform D once, then perform zero or more repetitions of E, then perform F once'. The basic expression αel means 'upon its activation, execute a single action with communication capabilities α and terminate, waiting at least e units of time and no more than l units of time to do so'. The concurrent composition operator is basically a disjoint union and hence differs from its counterparts in CCS and COSY, and is similar to the $\|_\emptyset$ in TCSP [22]. For instance, $a00 \| \widehat{a}00$ can perform the $\{a\}$ and $\{\widehat{a}\}$ actions individually (as well as a two-action step $\{a, \widehat{a}\}$), but no synchronised action (in contrast to $a.nil | \widehat{a}.nil$ in CCS). Finally, scoping $E \,\mathsf{sc}\, A$ implements a combination of synchronisation and restriction. In essence, it applies the CCS synchronisation mechanism over all the concurrently enabled pairs $(a, \widehat{a})$, for $a \in A$, of conjugate action names but it prevents the individual actions a and $\widehat{a}$ from occurring.

Static expressions describe structural characteristics of concurrent systems. Their behaviour will be modelled using dynamic at-expressions, introduced next.

4.2 Dynamic at-expressions

The syntax of (standard) dynamic PBC expressions is changed by adding time related annotations to the over- and underbars. Each such annotation is a pair of two non-negative integers that correspond to the *age* of the 'youngest' and 'oldest' token that might be consumed. For example, $\overline{E}^{00}$ is an expression E which is in its initial state and all tokens present are zero time units old. Another example, $\underline{E}_{35}\,\mathbf{;}\,F$, is a sequential composition where the first component has terminated, and produced some tokens. The exact number (and clock values) of these tokens is not represented by the annotation, but what is represented is the age of the youngest token (3 time units), and the age of of the oldest one (5 time units). Effectively, this means that the annotation gives an *age range* for the tokens in the state which is represented by the expression. This, in general, provides less information than that conveyed by the token timings provided by at-boxes. However, it will turned out that this reduced (or abstracted) view is sufficient to reason about the behaviour. We will re-visit this issue later on.

The *dynamic at-expressions*, G, are defined below, where E and Z denote static at-expression as in the syntax (1) and $\mathbb{EL} \in \mathbb{D}$.

$$
\begin{aligned}
G \;::=\;& \overline{E}^{\mathbb{EL}} \mid E\,;G \mid G\Box E \mid G\,\mathsf{sc}\,A \mid \langle\!\langle E \circledast Z \circledast G\rangle\!\rangle \mid \langle\!\langle E \circledast K \circledast E\rangle\!\rangle \mid \\
& \underline{E}_{\mathbb{EL}} \mid G\,;E \mid E\Box G \mid G\|G \mid \langle\!\langle G \circledast Z \circledast E\rangle\!\rangle \\
K \;::=\;& \overline{Z}^{\mathbb{EL}} \mid G\,;E \mid K\Box Z \mid \langle\!\langle G \circledast Z \circledast E\rangle\!\rangle \mid \langle\!\langle E \circledast Z \circledast G\rangle\!\rangle \mid \\
& \underline{Z}_{\mathbb{EL}} \mid E\,;G \mid Z\Box K \mid \langle\!\langle E \circledast K \circledast E\rangle\!\rangle \mid K\,\mathsf{sc}\,A
\end{aligned}
\qquad (2)
$$

Given that we are primarily interested in at-expressions that can be derived from expressions of the form $\overline{E}^{00}$, the above syntax may appear to be too permissive. For example, it admits expressions like $\overline{a03}^{55}$ which has an inconsistent timing information (the enabled action cannot wait for more than 3 time units before being executed, yet the age of the enabling tokens is already 5). However, such an expression may be a part of another, fully consistent expression, e.g., $(\overline{a03}^{55})\,\mathsf{sc}\{a\}$, and thus cannot be excluded.

$\overline{E\parallel F}^{\mathbb{EL}} \equiv \overline{E}^{\mathbb{EL}}\parallel\overline{F}^{\mathbb{EL}}$	$\underline{E}_{\mathbb{EL}}\parallel\underline{F}_{\mathbb{E'L'}} \equiv \underline{E\parallel F}_{\min\{\mathbb{E},\mathbb{E}'\}\,\max\{\mathbb{L},\mathbb{L}'\}}$
$\overline{E\Box F}^{\mathbb{EL}} \equiv \overline{E}^{\mathbb{EL}}\Box F$	$\underline{E}_{\mathbb{EL}}\Box F \equiv \underline{E\Box F}_{\mathbb{EL}}$
$\overline{E\Box F}^{\mathbb{EL}} \equiv E\Box\overline{F}^{\mathbb{EL}}$	$E\Box\underline{F}_{\mathbb{EL}} \equiv \underline{E\Box F}_{\mathbb{EL}}$
$\overline{E\,\mathsf{sc}\,A}^{\mathbb{EL}} \equiv \overline{E}^{\mathbb{EL}}\,\mathsf{sc}\,A$	$\underline{E}_{\mathbb{EL}}\,\mathsf{sc}\,A \equiv \underline{E\,\mathsf{sc}\,A}_{\mathbb{EL}}$
$\overline{E\,;F}^{\mathbb{EL}} \equiv \overline{E}^{\mathbb{EL}}\,;F$	$E\,;\underline{F}_{\mathbb{EL}} \equiv \underline{E\,;F}_{\mathbb{EL}}$
$\underline{E}_{\mathbb{EL}}\,;F \equiv E\,;\overline{F}^{\mathbb{EL}}$	$\overline{\langle\!\langle D\circledast E\circledast F\rangle\!\rangle}^{\mathbb{EL}} \equiv \langle\!\langle\overline{D}^{\mathbb{EL}}\circledast E\circledast F\rangle\!\rangle$
$\langle\!\langle\underline{D}_{\mathbb{EL}}\circledast E\circledast F\rangle\!\rangle \equiv \langle\!\langle D\circledast\overline{E}^{\mathbb{EL}}\circledast F\rangle\!\rangle$	$\langle\!\langle D\circledast\underline{E}_{\mathbb{EL}}\circledast F\rangle\!\rangle \equiv \langle\!\langle D\circledast E\circledast\overline{F}^{\mathbb{EL}}\rangle\!\rangle$
$\langle\!\langle D\circledast\overline{E}^{\mathbb{EL}}\circledast F\rangle\!\rangle \equiv \langle\!\langle D\circledast\underline{E}_{\mathbb{EL}}\circledast F\rangle\!\rangle$	$\langle\!\langle D\circledast F\circledast\underline{F}_{\mathbb{EL}}\rangle\!\rangle \equiv \underline{\langle\!\langle D\circledast E\circledast F\rangle\!\rangle}_{\mathbb{EL}}$

Table 1. Rules of the structural equivalence for at-expressions.

4.3 Operational semantics of at-expressions

We follow the way through which the semantics of the standard PBC was defined, with appropriate modifications in order to address timing restrictions. We first define a structural equivalence relation on at-expressions which aims to capture the most fundamental correspondence between expressions. For example, $\underline{E}_{\mathbb{EL}}\,;F \equiv E\,;\overline{F}^{\mathbb{EL}}$ states that a sequential system in which its first component has terminated is the same as the system in which the second component is ready to begin its operation. The time annotations are not changed since the entire state produced by the first component is passed to the second one. Formally, $\equiv$ is the least equivalence relation on dynamic at-expressions such that the rules in table 1 are satisfied. Note that we do not give any rule for $\overline{E}^{\mathbb{EL}} \| \overline{F}^{\mathbb{E}'\mathbb{L}'}$ with $\mathbb{EL} \neq \mathbb{E}'\mathbb{L}'$ as such an expression can never be derived from initially marked static expressions, which are the only at-expressions we are interested in.

Proposition 1. *Assuming that we treat the rules in table 1 as term rewriting rules, if $G \equiv H$ and G is an at-expression, then so is H.*

Proof. Follows from a similar result in the standard box algebra. □

4.4 SOS rules

Similarly as at-boxes, at-expressions can perform two kinds of operational semantics moves, namely *action* moves and *time* moves. A time move has the form

$$G \xrightarrow{\checkmark} H$$

and an action move has the form

$$G \xrightarrow{\Gamma} H$$

where Γ is a finite multiset of communication actions. We now define various types of moves of the structural operational semantics of dynamic at-expressions.

Empty moves The following rules deal with the empty action moves.

$$\frac{G \equiv H}{G \xrightarrow{\varnothing} H} \qquad \frac{G \xrightarrow{\varnothing} J \xrightarrow{\Gamma} H}{G \xrightarrow{\Gamma} H} \qquad \frac{G \xrightarrow{\Gamma} J \xrightarrow{\varnothing} H}{G \xrightarrow{\Gamma} H}$$

Basic action A basic action can occur if its timing restrictions are satisfied by the age range of its overbar:

$$\frac{\mathbb{EL} \ \mathsf{tsat} \ el}{\overline{\alpha el}^{\mathbb{EL}} \xrightarrow{\{\alpha\}} \underline{\alpha el}_{00}}$$

Note that the age range of a newly created underbar is always set to (00).

Scoping There is a single rule for scoping:

$$\frac{G \xrightarrow{\{a_1,\widehat{a}_1\}+\cdots+\{a_k,\widehat{a}_k\}+\Gamma} H \ , \ (A \cup \widehat{A}) \cap \Gamma = \varnothing \ , \ a_1,\ldots,a_k \in A}{G \,\mathsf{sc}\, A \xrightarrow{k\cdot\{\imath\}+\Gamma} H \,\mathsf{sc}\, A}$$

Other operators There is no real difference in the rules for the remaining operators when compared with the standard PBC [5, 6].

$$\frac{G \xrightarrow{\Gamma} G' \;,\; H \xrightarrow{\Gamma'} H'}{G \| H \xrightarrow{\Gamma + \Gamma'} G' \| H'} \qquad \frac{G \xrightarrow{\Gamma} H}{\begin{array}{c}\langle\!\langle G \circledast E \circledast F \rangle\!\rangle \xrightarrow{\Gamma} \langle\!\langle H \circledast E \circledast F \rangle\!\rangle \\ \langle\!\langle E \circledast G \circledast F \rangle\!\rangle \xrightarrow{\Gamma} \langle\!\langle E \circledast H \circledast F \rangle\!\rangle \\ \langle\!\langle E \circledast F \circledast G \rangle\!\rangle \xrightarrow{\Gamma} \langle\!\langle E \circledast F \circledast H \rangle\!\rangle\end{array}}$$

$$\frac{G \xrightarrow{\Gamma} H}{\begin{array}{c} E \Box G \xrightarrow{\Gamma} E \Box H \\ G \Box E \xrightarrow{\Gamma} H \Box E \end{array}} \qquad \frac{G \xrightarrow{\Gamma} H}{\begin{array}{c} G \,;\, E \xrightarrow{\Gamma} H \,;\, E \\ E \,;\, G \xrightarrow{\Gamma} E \,;\, H \end{array}}$$

Urgent labels of at-expressions To identify cases when time moves can be applied, we need the notion of *urgent* labels which can be executed by an at-expression. Urgent labels of dynamic at-expressions are defined by

$$\mathsf{urgent}_{lab}(H) \stackrel{\mathrm{df}}{=} \{\alpha \mid \alpha^0 \in \mathsf{enabled}_{aux}(H)\},$$

where $\mathsf{enabled}_{aux}(H)$ is a set defined by induction on the structure of H. There are two kinds of objects which $\mathsf{enabled}_{aux}(H)$ can contain, namely α^δ and a, where $\alpha \in \mathcal{A} \cup \{\imath\}$, $a \in \mathcal{A}$ and $\delta \in \{0,1\}$. Intuitively, α^0 means that the label α is enabled and urgent in the expression H, α^1 means that the label α is enabled but non-urgent, and a means that there is a pair of conjugate labels $(a, \widehat{a})$ enabled simultaneously and at least one of these labels is urgent. In more detail, for the base case, we have:

$$\mathsf{enabled}_{aux}(\overline{\alpha el}^{\mathbb{EL}}) \stackrel{\mathrm{df}}{=} \begin{cases} \{\alpha^0\} & \text{if } \quad \mathbb{EL} \;\mathsf{tsat}\; el \quad \text{and} \quad l = \mathbb{L} \\ \{\alpha^1\} & \text{if } \quad \mathbb{EL} \;\mathsf{tsat}\; el \quad \text{and} \quad l < \mathbb{L} \\ \varnothing & \text{otherwise .} \end{cases}$$

$$\mathsf{enabled}_{aux}(\underline{\alpha el}_{\mathbb{EL}}) \stackrel{\mathrm{df}}{=} \varnothing$$

For more complicated expressions H, we define $\mathsf{urgent}_{lab}(H)$ as the smallest set such that, whenever $H \equiv G$ then

$$\mathsf{enabled}_{aux}(G) = \mathsf{enabled}_{aux}(H)$$

and then the following hold for individual cases of composition operators. For scoping, if $a \in \mathsf{enabled}_{aux}(G)$ and $a \in (A \cup \widehat{A})$ then:

$$\imath^0 \in \mathsf{enabled}_{aux}(G \;\mathsf{sc}\; A) \;,$$

as well as

$$\{\alpha^\delta \in \mathsf{enabled}_{aux}(G) \mid \alpha \notin (A \cup \widehat{A})\} \subseteq \mathsf{enabled}_{aux}(G \,\mathsf{sc}\, A)$$
$$\{a \in \mathsf{enabled}_{aux}(G) \mid a \notin (A \cup \widehat{A})\} \subseteq \mathsf{enabled}_{aux}(G \,\mathsf{sc}\, A)\ .$$

For concurrent composition,

$$\mathsf{enabled}_{aux}(G) \cup \mathsf{enabled}_{aux}(J) \subseteq \mathsf{enabled}_{aux}(G\|J)$$
$$\{a \mid a^\delta \in \mathsf{enabled}_{aux}(G) \wedge \widehat{a}^{\delta'} \in \mathsf{enabled}_{aux}(J) \wedge \delta \cdot \delta' = 0\} \subseteq \mathsf{enabled}_{aux}(G\|J)\ .$$

For the remaining operators, we have that:

$$\mathsf{enabled}_{aux}(G) \subseteq \mathsf{enabled}_{aux}(\langle\!\langle G \circledast E \circledast F \rangle\!\rangle) \cap \mathsf{enabled}_{aux}(\langle\!\langle E \circledast G \circledast F \rangle\!\rangle) \cap \mathsf{enabled}_{aux}(\langle\!\langle E \circledast F \circledast G \rangle\!\rangle)$$
$$\mathsf{enabled}_{aux}(G) \subseteq \mathsf{enabled}_{aux}(G \Box E) \cap \mathsf{enabled}_{aux}(E \Box G)$$
$$\mathsf{enabled}_{aux}(G) \subseteq \mathsf{enabled}_{aux}(G\,;E) \cap \mathsf{enabled}_{aux}(E\,;G)\ .$$

Time moves There is a single time rule:

$$\frac{\mathsf{urgent}_{lab}(G) = \varnothing}{G \xrightarrow{\checkmark} G^{\checkmark}}$$

where $G^{\checkmark}$ is G with each time annotation $\mathbb{EL}$ at an over- or underbar changed to $(\mathbb{E}+1)(\mathbb{L}+1)$. Notice that a time move can only be applied at the topmost level of an expression as it cannot be 'propagated' through the expression using action rules. This ensures that time progresses uniformly.

Note also that to capture the urgency of enabled label, one cannot use a definition of the following kind: $\alpha \in \mathsf{urgent}_{lab}(G)$ if α is enabled by G but not by $G^{\checkmark}$. The reason is that enabling alone cannot find out precisely which action cannot wait any longer. Take for, instance, the following at-expression: $\overline{a00 \Box a01}^{00}$. We have here two possible occurrences of a leading to the the same expression $\underline{a00 \Box a01}_{00}$. However, one of them should be considered urgent, even though we still have that a is enabled by $\overline{a00 \Box a01}^{11}$.

It can be seen that the rules of the operational semantics do not lead outside the set of dynamic at-expressions.

Proposition 2. *Assuming that we treat the rules of the operational semantics as term rewriting rules, and H has been derived from an at-expression, then H is also an at-expression.*

Proof. Follows from a similar result in the standard box algebra. □

Reachability trees of at-expressions As already mentioned, we are ultimately interested in those at-expressions that can be reached, through the rules of the structural operational semantics, from static at-expressions started at zero time, i.e., we are interested in at-expressions of the form $G = \overline{E}^{00}$ executed using the operational semantics rules defined earlier in this section. The representation that we will use to capture the behaviour of G will again be a *reachability tree*, denoted by RT_G. Its nodes are labelled by equivalence classes of dynamic expressions reachable from G, and arcs are labelled by multisets over $\mathcal{A} \cup \{\imath\}$ or the $\surd$ symbol. The root node is labelled by $[G]_\equiv$ and, if a node is labelled by $[H]_\equiv$, then: for every move

$$H \xrightarrow{\Gamma} J\,,$$

there is a unique descendant labelled by $[J]_\equiv$ and the arc leading to it is labelled by Γ, and if the time move is possible for H then there is a unique descendant labelled by $[H^{\surd}]_\equiv$ and the arc leading to it is labelled by $\surd$.

Note that we base reachability trees (and later transition systems) of at-expressions on the equivalence classes of $\equiv$, rather than on at-expressions themselves, since we may have $G \xrightarrow{\varnothing} G'$ for two different expressions G and G', whereas in the domain of at-boxes, $\mathbf{\Theta}[\varnothing\rangle\mathbf{\Xi}$ always implies $\mathbf{\Theta} = \mathbf{\Xi}$.

4.5 Examples

Our first example, in figure 2, shows an at-expression with two sequential actions a, c in parallel with two other sequential actions $b, \widehat{c}$ and scoping on action c. Different execution scenarios can be followed. We choose, in line (2), to execute action a followed by a time move in line (3) which is the only possible move at this stage. Action b becomes then urgent and in line (4) b is executed. After three time moves, in line (6), the c part of enabled synchronisation action is urgent, and so time move is disallowed. Synchronisation takes place in line (7), by executing the silent synchronisation action $\imath$.

The second example, in figure 3, shows an at-expression consisting of an action a in parallel with two sequential actions $b, \widehat{a}$ and scoping on action a. In line (2), we cannot execute a due to the restriction imposed by the scoping operator (as well as the timing age) and b is not ready to fire. In line (3), after one time move, action b is urgent and must be executed immediately. In line (5), action a is urgent, but its counterpart $\widehat{a}$ is not enabled due to the time restrictions. As a result, the synchronisation action of the scoping operator is not possible and there are no other possible action moves after that.

5 An algebra of arc-time boxes

We now extend the box algebra to at-boxes, by defining compositionally a mapping Box which, for static at-expressions, returns at-boxes. The net algebra employs operators directly corresponding to (and denoted as) those used in the algebra of static at-expressions. All the net operators are similar to those in the

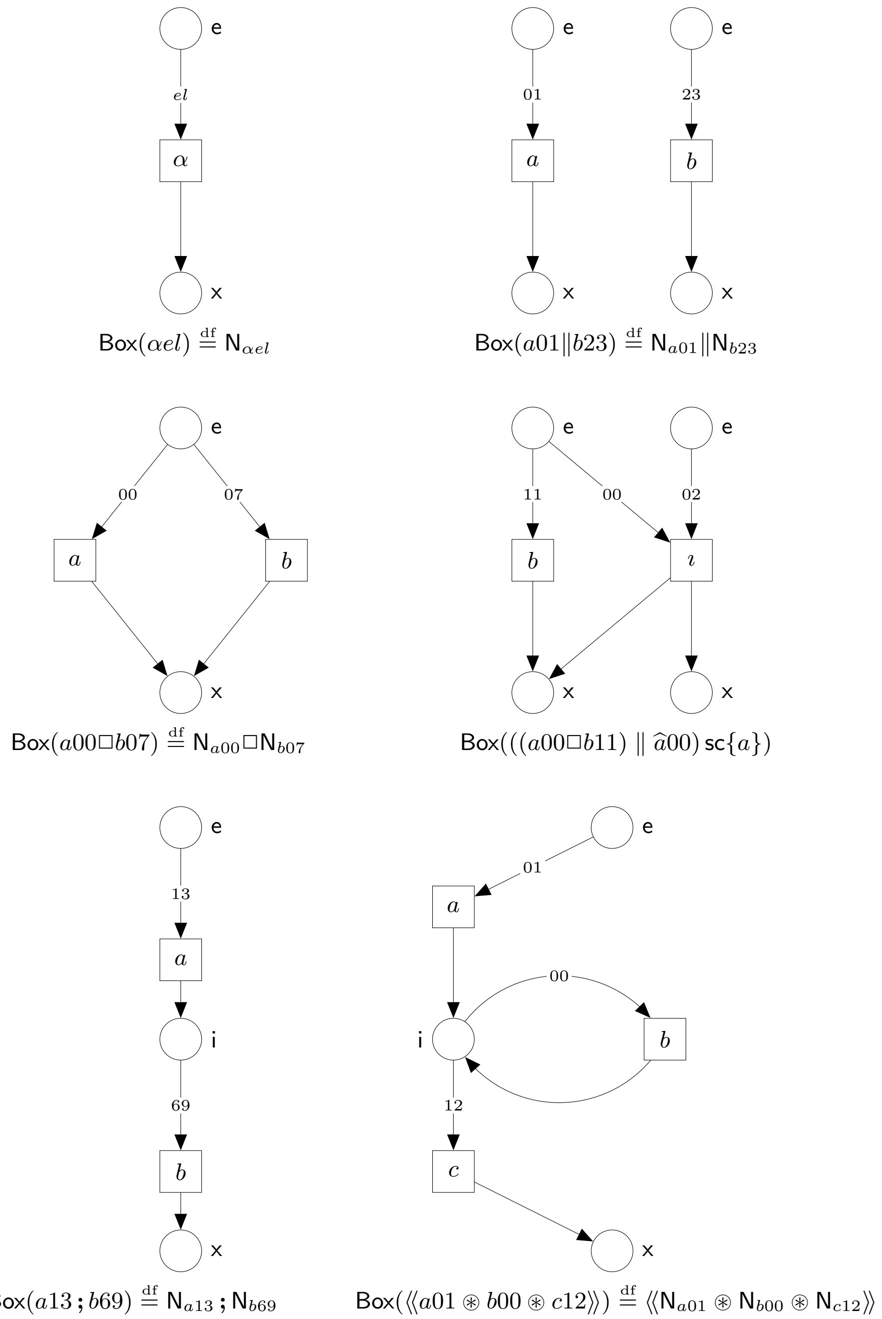
(1)	$\overline{((a02\,;c44)\parallel(b11\,;\widehat{c}14))\,\mathsf{sc}\{c\}}^{00}$	$\equiv$
(2)	$((\overline{a02}^{00}\,;c44)\parallel(\overline{b11}^{00}\,;\widehat{c}14))\,\mathsf{sc}\{c\}$	$\xrightarrow{\{a\}}$
(3)	$((\underline{a02}_{00}\,;c44)\parallel(\overline{b11}^{00}\,;\widehat{c}14))\,\mathsf{sc}\{c\}$	$\xrightarrow{\surd}$
(4)	$((\underline{a02}_{11}\,;c44)\parallel(\overline{b11}^{11}\,;\widehat{c}14))\,\mathsf{sc}\{c\}$	$\xrightarrow{\{b\}}$
(5)	$((\underline{a02}_{11}\,;c44)\parallel(\underline{b11}_{00}\,;\widehat{c}14))\,\mathsf{sc}\{c\}$	$\equiv$
(6)	$((a02\,;\overline{c44}^{11})\parallel(b11\,;\overline{\widehat{c}14}^{00}))\,\mathsf{sc}\{c\}$	$\xrightarrow{\surd\surd\surd}$
(7)	$((a02\,;\overline{c44}^{44})\parallel(b11\,;\overline{\widehat{c}14}^{33}))\,\mathsf{sc}\{c\}$	$\xrightarrow{\{\imath\}}$
(8)	$((a02\,;\underline{c44}_{00})\parallel(b11\,;\underline{\widehat{c}14}_{00}))\,\mathsf{sc}\{c\}$	$\equiv$
(9)	$\underline{((a02\,;c44)\parallel(b11\,;\widehat{c}14))\,\mathsf{sc}\{c\}}_{00}$	

Fig. 2. An evolution of the expression $\overline{((a02\,;c44)\parallel(b11\,;\widehat{c}14))\,\mathsf{sc}\{c\}}^{00}$.

(1)	$\overline{(a11\parallel(b11\,;\widehat{a}11))\,\mathsf{sc}\{a\}}^{00}$	$\equiv$
(2)	$(\overline{a11}^{00}\parallel(\overline{b11}^{00}\,;\widehat{a}11))\,\mathsf{sc}\{a\}$	$\xrightarrow{\surd}$
(3)	$(\overline{a11}^{11}\parallel(\overline{b11}^{11}\,;\widehat{a}11))\,\mathsf{sc}\{a\}$	$\xrightarrow{\{b\}}$
(4)	$(\overline{a11}^{11}\parallel(\underline{b11}_{00}\,;\widehat{a}11))\,\mathsf{sc}\{a\}$	$\equiv$
(5)	$(\overline{a11}^{11}\parallel(b11\,;\overline{\widehat{a}11}^{00}))\,\mathsf{sc}\{a\}$	

Fig. 3. An evolution of the expression $\overline{(a00\parallel(b11\,;\widehat{a}01))\,\mathsf{sc}\{a\}}^{00}$.

standard PBC with two important modifications: (i) changing the definition of the basic net corresponding to a single action, and (ii) taking care of the time restrictions associated with transition input arcs. Essentially, the latter means that if p and t are a place and transition which are 'carried forward' by a net operator, then the associated time constraint $\lambda(p,t)$ is also carried forward. Moreover, in the scoping operation, if t and t' are fused together to yield a $\imath$-labelled synchronisation transition u, then we assume that ${}^\bullet t \cap {}^\bullet t' = \varnothing$ and $t^\bullet \cap t'^\bullet = \varnothing$. We omit here a full definition of the composition operators (they all can be found in the appendix), and instead provide in figure 4 a number of examples involving the operators used in the algebra of at-boxes.



$\mathsf{Box}(\alpha el) \stackrel{\mathrm{df}}{=} \mathsf{N}_{\alpha el}$

$\mathsf{Box}(a01\|b23) \stackrel{\mathrm{df}}{=} \mathsf{N}_{a01}\|\mathsf{N}_{b23}$

$\mathsf{Box}(a00\Box b07) \stackrel{\mathrm{df}}{=} \mathsf{N}_{a00}\Box\mathsf{N}_{b07}$

$\mathsf{Box}(((a00\Box b11)\ \|\ \widehat{a}00)\,\mathsf{sc}\{a\})$

$\mathsf{Box}(a13\,;b69) \stackrel{\mathrm{df}}{=} \mathsf{N}_{a13}\,;\mathsf{N}_{b69}$

$\mathsf{Box}(\langle\langle a01 \circledast b00 \circledast c12\rangle\rangle) \stackrel{\mathrm{df}}{=} \langle\langle \mathsf{N}_{a01} \circledast \mathsf{N}_{b00} \circledast \mathsf{N}_{c12}\rangle\rangle$

Fig. 4. Examples of nets defined in the algebra of at-boxes.

Formally, we introduce a denotational semantics of at-expressions through the semantical mapping Box from static at-expressions to at-boxes so that

$$\mathsf{Box}(\alpha el) \stackrel{\mathrm{df}}{=} \mathsf{N}_{\alpha el}$$

where $\mathsf{N}_{\alpha el}$ is shown in figure 4, and for other static at-expressions:

$$\mathsf{Box}(E \,\mathsf{sc}\, A) \stackrel{\mathrm{df}}{=} \mathsf{Box}(E) \,\mathsf{sc}\, A$$

$$\mathsf{Box}(E \Box F) \stackrel{\mathrm{df}}{=} \mathsf{Box}(E) \Box \mathsf{Box}(F)$$

$$\mathsf{Box}(E \| F) \stackrel{\mathrm{df}}{=} \mathsf{Box}(E) \| \mathsf{Box}(F)$$

$$\mathsf{Box}(E \,;\, F) \stackrel{\mathrm{df}}{=} \mathsf{Box}(E) \,;\, \mathsf{Box}(F)$$

$$\mathsf{Box}(\langle\!\langle D \circledast E \circledast F \rangle\!\rangle) \stackrel{\mathrm{df}}{=} \langle\!\langle \mathsf{Box}(D) \circledast \mathsf{Box}(E) \circledast \mathsf{Box}(F) \rangle\!\rangle \,.$$

Since we are interested in the behaviour of systems starting from their initial state, we also need to describe $\mathsf{Box}(G)$, for any dynamic at-expression of the form $G = \overline{E}^{00}$. The appropriate at-box is defined as $\mathsf{Box}(E)$ with $\mu(p)$ changed to 0, for every entry place p.

In order to guarantee the safeness of the underlying PT-net, we followed the standard treatment of the PBC and restricted slightly the syntax of the second component of the iterative construct $\langle\!\langle D \circledast E \circledast F \rangle\!\rangle$, by stipulating that each application of parallel composition is within the scope of some sequential composition or iteration. And, as shown in the appendix,

Proposition 3. *For every dynamic at-expression $G = \overline{E}^{00}$, the mapping* Box *returns an at-box.*

Consistency between denotational and operational semantics We now formulate the central result of this paper which states that the two semantics of at-expressions are equivalent. The following result, proved in the appendix, extends that for the standard PBC where the transition systems of corresponding expressions and boxes are isomorphic [6].

Theorem 1. *For every dynamic at-expression $G = \overline{E}^{00}$, the reachability trees RT_G and $\mathsf{RT}_{\mathsf{Box}(G)}$ are isomorphic.*

The first comment about the above theorem is that the result is not formulated in terms of *transition systems* of G and $\mathsf{Box}(G)$, as in the standard PBC, but rather in terms of their reachability trees. The reason is that the latter are not isomorphic (though they are strongly bisimilar [15]). Isomorphism of reachability graphs fails to hold because, in general, there is no one-to-one correspondence between the expressions reachable from G and the token timings reachable from the initial token timing of $\mathsf{Box}(G)$. To illustrate this, we consider the at-expression $G = \overline{((a00 \parallel b01) \parallel c11) \,;\, d01}^{00}$ and the corresponding at-box $\mathsf{Box}(G)$ shown in figure 5.

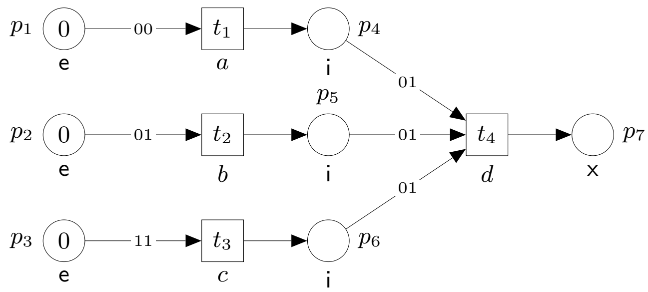


Fig. 5. An at-box corresponding to the expression $\overline{((a00 \parallel b01) \parallel c11)\,;d01}^{\,00}$.

It may be easily checked that this at-box allows the following two sequences of moves, both starting from the initial token timing:

	scenario1	*scenario2*
(1)	$(0,0,0,\bot,\bot,\bot,\bot)$ $[\{t_1,t_2\}\rangle$	$(0,0,0,\bot,\bot,\bot,\bot)$ $[\{t_1\}\rangle$
(2)	$(\bot,\bot,0,0,0,\bot,\bot)$ $[\surd\rangle$	$(\bot,0,0,0,\bot,\bot,\bot)$ $[\surd\rangle$
(3)	$(\bot,\bot,1,1,1,\bot,\bot)$ $[\{t_3\}\rangle$	$(\bot,1,1,1,\bot,\bot,\bot)$ $[\{t_2,t_3\}\rangle$
(4)	$(\bot,\bot,\bot,1,1,0,\bot)$ $[\{t_4\}\rangle$	$(\bot,\bot,\bot,1,0,0,\bot)$ $[\{t_4\}\rangle$
(5)	$(\bot,\bot,\bot,\bot,\bot,\bot,0)$	$(\bot,\bot,\bot,\bot,\bot,\bot,0)$

The two corresponding execution sequences for the expression G are shown in figure 6. One may further observe that the left marking in line (4) above corresponds to the expressions in lines $(4')$ and $(4a')$, and that the right marking in line (4) above corresponds to the expressions in lines $(4'')$ and $(4a'')$. However, the two markings are different yet we have $(4') \equiv (4a') = (4a'') \equiv (4'')$, which indicates that the expressions in lines $(4', 4a', 4'', 4a'')$ represent the same state of the system. It is therefore impossible to show that the reachability graphs of G and $\mathsf{Box}(G)$ are isomorphic. This should not be treated as a cause for concern since theorem 1 above still establishes very strong relationship between the behaviours of the at-expressions and the corresponding at-boxes. The above discussions also shows that, in general, there can be no direct translation from dynamic at-expressions to at-boxes since, informally, there are fewer of the former than of the latter. In a way, as we already mentioned, at-expressions are more *abstract* than the corresponding at-boxes. This, as we expect, can be used to improve model-checking of behaviours specified by at-expressions, by providing an equivalence relation between reachable token timings of at-boxes which could be used to improve the efficiency of the unfolding of at-boxes (with the resulting unfoldings being smaller). This hypothesis is at the present moment investigated in the context of the general scheme for generating net unfoldings in [11] and the corresponding tool support.

5.1 Cluster based evolutions

The above discussion also means that a proof of theorem 1 cannot be obtained by a simple adaptation of that used in [6] since dynamic at-expressions cannot be unambiguously mapped to at-boxes. To explain how we cope with this problem, assume that we have an at-expression $G = \overline{E}^{00}$ not involving action scoping, like that considered above. One can then make a crucial observation that for each transition t in $\mathsf{Box}(G)$, the annotations of its input arcs are *exactly the same*, say *el* (this is, clearly, not true of at-boxes in general). This specific property implies that to check the enabledness of t it suffices to check that each input place to t has a token, and that the age of the oldest and the youngest token in such places lies between e and l. And this is strictly less information than we require in the general case.

For every at-expression G, we define *clusters* $CL(G) = \{\mathsf{cl}_1, \ldots, \mathsf{cl}_n\}$ which are sets of places of $\mathsf{Box}(G)$ corresponding to the entry/exit interfaces of $\mathsf{Box}(G)$ as well as the input places of all individual transitions. This allows one to express the evolutions of $\mathsf{Box}(G)$ in terms of changing the 'state' of clusters rather than the state of individual places. More precisely, the *cluster filling* $\mathcal{M}$ of $\mathsf{Box}(G)$ (see appendix D) is a mapping which associates with each cluster either $\perp$ (meaning the cluster is empty), or $\mathbb{EL}$ (meaning the cluster has at least one token and the age of the youngest token in it is $\mathbb{E}$, and the age of the oldest is $\mathbb{L}$). We then define enabledness of steps and dynamic changes of the net w.r.t. cluster based states, in a way quite similar to that used in the usual semantics, including the notion of a reachability tree. It can then be proven that *cluster-based at-boxes* are behaviorally equivalent to normal at-boxes (more precisely, their reachability trees are isomorphic).

For the example considered above, there are six clusters: $\mathsf{cl}_1 \stackrel{\text{df}}{=} \{p_1, p_2, p_3\}$ $\mathsf{cl}_2 \stackrel{\text{df}}{=} \{p_1\}$, $\mathsf{cl}_3 \stackrel{\text{df}}{=} \{p_2\}$, $\mathsf{cl}_4 \stackrel{\text{df}}{=} \{p_3\}$, $\mathsf{cl}_5 \stackrel{\text{df}}{=} \{p_4, p_5, p_6\}$ and $\mathsf{cl}_6 \stackrel{\text{df}}{=} \{p_7\}$. Assuming this ordering of clusters, our two scenarios can be re-written as follows:

	scenario1	*scenario2*
$(1''')$	$(00,00,00,00,\perp,\perp)\ [\{t_1,t_2\}\rangle$	$(00,00,00,00,\perp,\perp)\ [\{t_1\}\rangle$
$(2''')$	$(00,\perp,\perp,00,00,\perp)\ [\surd\rangle$	$(00,\perp,00,00,00,\perp)\ [\surd\rangle$
$(3''')$	$(11,\perp,\perp,11,11,\perp)\ [\{t_3\}\rangle$	$(11,\perp,11,11,11,\perp)\ [\{t_2,t_3\}\rangle$
$(4''')$	$(\perp,\perp,\perp,\perp,01,\perp)\ [\{t_4\}\rangle$	$(\perp,\perp,\perp,\perp,01,\perp)\ [\{t_4\}\rangle$
$(5''')$	$(\perp,\perp,\perp,\perp,\perp,00)$	$(\perp,\perp,\perp,\perp,\perp,00)$

Note that the problem encountered before with line (4) is no longer present in line $(4''')$. Effectively, this means that we can suitably adopt the proof technique used in, e.g., [6], to justify theorem 1 (as far as synchronised transitions are concerned, they will have two clusters which are responsible for their enabling).

scenario1

$$(1') \qquad \overline{((a00 \parallel b01) \parallel c11)\,;d01}^{\,00} \qquad \xrightarrow{\{a,b\}}$$

$$(2') \qquad ((\underline{a00}_{\,00} \parallel \underline{b01}_{\,00}) \parallel \overline{c11}^{\,00})\,;d01 \qquad \xrightarrow{\surd}$$

$$(3') \qquad ((\underline{a00}_{\,11} \parallel \underline{b01}_{\,11}) \parallel \overline{c11}^{\,11})\,;d01 \qquad \xrightarrow{\{c\}}$$

$$(4') \qquad ((\underline{a00}_{\,11} \parallel \underline{b01}_{\,11}) \parallel \underline{c11}_{\,00})\,;d01 \qquad \equiv$$

$$(4a') \qquad ((a00 \parallel b01) \parallel c11)\,;\overline{d01}^{\,01} \qquad \xrightarrow{\{d\}}$$

$$(5') \qquad \underline{((a00 \parallel b01) \parallel c11)\,;d01}_{\,00}$$

scenario2

$$(1'') \qquad \overline{((a00 \parallel b01) \parallel c11)\,;d01}^{\,00} \qquad \xrightarrow{\{a\}}$$

$$(2'') \qquad ((\underline{a00}_{\,00} \parallel \overline{b01}^{\,00}) \parallel \overline{c11}^{\,00})\,;d01 \qquad \xrightarrow{\surd}$$

$$(3'') \qquad ((\underline{a00}_{\,11} \parallel \overline{b01}^{\,11}) \parallel \overline{c11}^{\,11})\,;d01 \qquad \xrightarrow{\{b,c\}}$$

$$(4'') \qquad ((\underline{a00}_{\,11} \parallel \underline{b01}_{\,00}) \parallel \underline{c11}_{\,00})\,;d01 \qquad \equiv$$

$$(4a'') \qquad ((a00 \parallel b01) \parallel c11)\,;\overline{d01}^{\,01} \qquad \xrightarrow{\{d\}}$$

$$(5'') \qquad \underline{((a00 \parallel b01) \parallel c11)\,;d01}_{\,00}$$

Fig. 6. Two execution sequences corresponding to scenario 1 and 2.

6 Concluding remarks

In this paper, we introduced a new compositional model of arc-based time Petri nets, and a corresponding process algebra of time expressions. We have explained the nature of the correspondence between the two algebras, in terms of their respective reachability trees, and outlined an intermediate (cluster based) representation used in the proof of this correspondence. In particular, these results make it possible to combine the verification techniques developed independently for process algebra and Petri nets with timing, and to give a syntax oriented semantics of real-time specification languages. We also plan to explore more efficient model checking technique based on the observations made in this paper.

Acknowledgement

This research was supported by the EC IST grant 511599 (Rodin).

References

1. P. A. Abdulla and N. Aletta: Timed Petri Nets and BQOs. Proc. of *ICATPN'01*, J.-M. Colom, M. Koutny (Eds.). Springer-Verlag, Lecture Notes in Computer Science 2075 (2001) 53-70.
2. J. Baeten and W. P. Weijland: *Process Algebra*. Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press (1990).
3. B. Berthomieu and M. Diaz: Modelling and verification of Time Dependent Systems Using Time Petri Nets. *IEEE Trans. on Soft. Eng.* 17 (1991) 259-273.
4. E. Best, R. Devillers and J. Hall: The Petri Box Calculus: a New Causal Algebra with Multilabel Communication. In: *Advances in Petri Nets*, G. Rozenberg (Ed.). Springer-Verlag, Lecture Notes in Computer Science 609 (1992) 21-69.
5. E. Best, R. Devillers and M. Koutny: A Unified Model for Nets and Process Algebras. In: *Handbook of Process Algebra*, J. A. Bergstra, A. Ponse, S. A. Smolka, (Eds.). Elsevier (2001) 873–944.
6. E. Best, R. Devillers and M. Koutny: *Petri Net Algebra*. EATCS Monographs on TCS, Springer (2001).
7. B. Bieber and H. Fleischhack: Model Checking of Time Petri Nets Based on Partial Order Semantics. Proc. of *CONCUR'99*, J. Baeten and S. Mauw (Eds.). Springer-Verlag, Lecture Notes in Computer Science 1664 (1999) 210-225.
8. T. Bolognesi, F. Lucidi and S. Trigila: From Timed Petri Nets to timed LOTOS. Proc. of *PSTV'90*, L. Logrippo et al. (Eds.). North-Holland (1990) 395-408.
9. C. A. R. Hoare: *Communicating Sequential Processes*. Prentice Hall (1985).
10. R. Janicki and P. E. Lauer: *Specification and Analysis of Concurrent Systems - the COSY Approach*. EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1992).
11. V. Khomenko, M. Koutny and W. Vogler: Canonical Prefixes of Petri Net Unfoldings. *Acta Informatica* 40 (2003) 95-118.
12. M. Koutny: A Compositional Model of Time Petri Nets. Proc. of *ICATPN'00*, M. Nielsen and D. Simpson (Eds.). Springer-Verlag, Lecture Notes in Computer Science 1825 (2000) 303-322.
13. O. Marroquin Alonso and D. de Frutos-Escrig: Computing a Finite Prefix of a Time Petri Net. Proc. of *ICATPN'01*, J.-M. Colom and M. Koutny (Eds.). Springer-Verlag, Lecture Notes in Computer Science 2075 (2001) 303-322.
14. P. Merlin and D. Farber: Recoverability of Communication Protocols - Implication of a Theoretical Study. *IEEE Trans. on Soft. Comm.* 24 (1976) 1036-1043.
15. R. Milner: *Communication and Concurrency*. Prentice Hall (1989).
16. T. Murata: Petri Nets: Properties, Analysis and Applications. *Proc. of IEEE* 77 (1989) 541-580.
17. A. Niaouris: An Algebra of Petri Nets with Arc-Based Time Restrictions. Proc. of *ICTAC'04*, Z. Liu and K. Araki (Eds.). Springer-Verlag, Lecture Notes in Computer Science 3407 (2005) 447-462.
18. M. Nielsen, V. Sassone and J. Srba: Properties of Distributed Timed-Arc Petri Nets. Proc. of *FSTTCS'01*, R. Hariharan, M. Mukund, V. Vinay (Eds.). Springer-Verlag, Lecture Notes in Computer Science 2245 (2001) 280-285.
19. G. D. Plotkin: A Structural Approach to Operational Semantics. Teachnical Report FN-19, Computer Science Department, University of Aarhus (1981).
20. L. Popova: Time Petri Nets. *Journal of Information Processing and Cybernetics EIK* 27 (1991) 227-244.
21. W. Reisig: *Petri Nets. An Introduction*. EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1985).
22. A. W. Roscoe: *The Theory and practice of Concurrency*. Prentice-Hall (1998).

A An algebra of at-boxes

In what now follows, we re-trace at much greater level of detail and extend the development presented in the main body of the paper.

A.1 Net substitution

The identities of places and transitions will play a key role, especially when defining the transition based SOS semantics of process expressions. As in the standard box algebra, place and transition identities will come in the form of finite labelled trees retracing the operators used to construct a box.

We shall assume that there are two disjoint sets of *basic* place and transition names, $\mathsf{P}_{\mathsf{root}}$ and $\mathsf{T}_{\mathsf{root}}$. Each name $\eta \in \mathsf{P}_{\mathsf{root}} \cup \mathsf{T}_{\mathsf{root}}$ can be viewed as a special tree with a single node labelled with η, which is both a root and a leaf. (All the transitions in figure 7 are assumed to be of that kind.) 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 $x \lhd \mathsf{T}$, where x is a basic name in $\mathsf{P}_{\mathsf{root}} \cup \mathsf{T}_{\mathsf{root}}$, and T is a set of trees, denotes a tree where the trees of the multiset are appended to an x-labelled root. Moreover,

- if $\mathsf{T} = \{\mathsf{t}\}$ is a singleton then $x \lhd \mathsf{T}$ will be denoted by $x \lhd \mathsf{t}$.
- $x \blacktriangleleft \mathsf{T}$ denotes the set of trees $\{x \lhd \mathsf{t} \mid \mathsf{t} \in \mathsf{T}\}$.
- $x \lhd (v_1 \blacktriangleleft \mathsf{T}_1, \ldots, v_k \blacktriangleleft \mathsf{T}_k)$ denotes the set of trees

$$\{x \lhd \{v_1 \lhd \mathsf{t}_1, \ldots, v_k \lhd \mathsf{t}_k\} \mid \mathsf{t}_1 \in \mathsf{T}_1 \wedge \ldots \mathsf{t}_k \in \mathsf{T}_k\} \ .$$

B Petri nets with arc-based time restrictions

An *arc-time Petri net* (or at-net) is a tuple $\Sigma \overset{\mathrm{df}}{=} (P, T, F, \lambda, M)$ such that:

- (P, T, F) is a net.
- λ is a mapping with the domain $P \cup T \cup ((P \times T) \cap F)$ such that, for every place p and transition t, $\lambda(p)$ is a symbol in $\{\mathsf{e}, \mathsf{i}, \mathsf{x}\}$, $\lambda(t)$ is an action in $\mathcal{A} \cup \{\imath\}$, and if $(p, t) \in F$ then $\lambda(p, t)$ is a time constraint in $\mathbb{D}^\infty$.
- $M : P \to \mathbb{N}$ is a marking.

Places labelled by e, i and x are respectively called entry, internal and exit, and their sets are denoted by ${}^\circ\Sigma$, $\ddot{\Sigma}$ and Σ°. We use the standard step sequence semantics for at-nets, in particular, the set of enabled steps will be denoted by $\mathsf{enabled}(\Sigma)$, and the firing of an enabled step $U \in \mathsf{enabled}(\Sigma)$ is denoted by $\Sigma[U\rangle\Sigma'$, where Σ' is defined as usual. (It is worth stressing that at-nets are nothing but ordinary nets with time annotations on the input arcs which are simply ignored at this point.) Following the standard box algebra terminology, we say that Σ is:

- *ex-directed* if ${}^\bullet p = q^\bullet = \varnothing$, for every entry place p and exit place q.
- *ex-restricted* if there is at least one entry place and at least one exit place.

- *static* if $M_\Sigma = \varnothing$ and every marking reachable from ${}^\circ\Sigma$ is safe and clean (the latter means that if all the entry or all the exit places are marked then all the remaining places are empty).
- *dynamic* if $M_\Sigma \neq \varnothing$ and every marking reachable from M_Σ or ${}^\circ\Sigma$ is safe and clean.
- *entry* (or *exit*) at-net if $M_\Sigma = {}^\circ\Sigma$ (resp. $M_\Sigma = \Sigma^\circ$).
- *ex-exclusive* if, for every marking M reachable from M_Σ or ${}^\circ\Sigma$, it is the case that $M \cap {}^\circ\Sigma = \varnothing$ or $M \cap \Sigma^\circ = \varnothing$, i.e., it is not possible to mark simultaneously an entry and an exit place.
- *initial* (or *final*) if it contains one token in each entry (resp. exit) place and tokens elswhere.

If $M_\Sigma = \varnothing$ then $\overline{\Sigma}$ (and $\underline{\Sigma}$) is obtained by changing the marking to ${}^\circ\Sigma$ (resp. Σ°). And $\lfloor\Sigma\rfloor$ denotes Σ with its marking set to the empty one.

The above notions will be transferred, whenever it makes sense, to other kinds of Petri nets based on at-nets.

B.1 Net refinement

We now give definitions of the composition operators for at-nets; these are basically the same as for the standard box algebra with the addition that we need to give the time annotations for all arcs from places to transitions. The relevant operator boxes are shown in figure 7.

Scoping Let $A \subseteq \mathcal{A}$ and Σ be an ex-restricted and ex-directed at-net. The result of a substitution of the transition $v_{\mathsf{sc}\,A}$ in $\Omega_{\mathsf{sc}\,A}$ by Σ is an at-net $\Phi = \Omega_{\mathsf{sc}\,A}(\Sigma)$ whose components are defined as follows.

Places. There are three kinds of places in Φ:

- For every entry place p in Σ, $q = e_{\mathsf{sc}\,A} \lhd v_{\mathsf{sc}\,A} \lhd p$ is an entry place in Φ with the marking $M_\Sigma(p)$.
- For every exit place p in Σ, $q = x_{\mathsf{sc}\,A} \lhd v_{\mathsf{sc}\,A} \lhd p$ is an exit place in Φ with the marking $M_\Sigma(p)$.
- For every internal place p in Σ, $q = v_{\mathsf{sc}\,A} \lhd p$ is an internal place in Φ with the marking $M_\Sigma(p)$.

Transitions, arcs and timing constraints. There are two kinds of transitions in Φ:

- For every transition t in Σ with a label not belonging to $A \cup \widehat{A}$, $w = v_{\mathsf{sc}\,A} \lhd t$ is a transition in Φ with the same label as t.
  There is an arc from a place q to w iff there was an arc from p to t; moreover, in such a case, $\lambda_\Phi(q, w) = \lambda_\Sigma(p, t)$.
  There is an arc from w to a place q iff there was an arc from t to p.

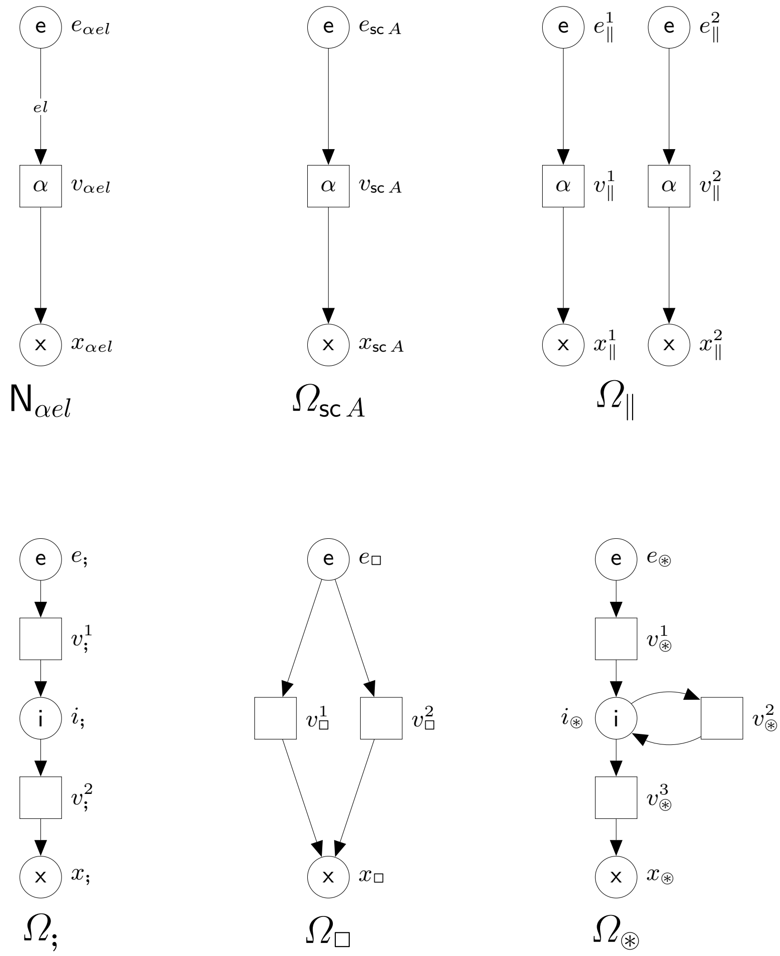


Fig. 7. An at-net $\mathsf{N}_{\alpha el}$ and five operator boxes.

– For all pairs of transitions t, u in Σ, one with a label $a \in A$ and the other with the label $\widehat{a}$, as well as with disjoint sets of pre- and post-places, $w = v_{\mathsf{sc}\,A} \lhd \{t, u\}$ is a transition in Φ with the label $\imath$.
There is an arc from a place q to w iff there was an arc from p to t (or u); moreover, in such a case, $\lambda_{\Phi}(q, w) = \lambda_{\Sigma}(p, t)$ (or $\lambda_{\Phi}(q, w) = \lambda_{\Sigma}(p, u)$).[2]
There is an arc from w to a place q iff there was an arc from t or u to p.

Other operators Let $\Omega_{op} \in \{\Omega_{\Box}, \Omega_{\circledast}, \Omega_{;}, \Omega_{\|}\}$ be any n-unary ($n \geq 2$) operator box and $\mathbf{\Sigma} = (\Sigma_1, \ldots, \Sigma_n) = (\Sigma_{v^1_{op}}, \ldots, \Sigma_{v^n_{op}})$ be an n-tuple of ex-restricted and ex-directed at-nets. The result of a simultaneous substitution of the transitions

[2] Note that the definition is well-formed since the pre-sets of t and u are disjoint.

v^i_{op} in Ω_{op} by the at-nets $\Sigma_{v^i_{op}}$ is a net $\Omega_{op}(\mathbf{\Sigma}) = \Phi$ whose components are defined as follows.

Transitions. There is one kind of transition in Φ:

- For all transitions v in Ω_{op} and t in Σ_v, $w = v \triangleleft t$ is a transition in Φ with the same label as t.

Places, arcs and timing constraints. There are two kinds of places in Φ:

- For every transition z in Ω_{op} and every internal place p in Σ_z, $q = z \triangleleft p$ is an internal place in Φ with the marking $M_{\Sigma_z}(p)$.
  There is an arc from q to a transition w iff $v = z$ and there was an arc from p to t; moreover, in such a case, $\lambda_\Phi(q, w) = \lambda_{\Sigma_z}(p, t)$.
  There is an arc from a transition w to q iff $v = z$ and there was an arc from t to p.
- For every place s in Ω_{op} with ${}^\bullet s = \{u_1, \ldots, u_k\}$ and $s^\bullet = \{u_{k+1}, \ldots, u_{k+m}\}$, we construct in Φ all the places

  $$q = s \triangleleft (u_1 \triangleleft p_1, \ldots, u_{k+m} \triangleleft p_{k+m}) \ ,$$

  where each p_i (for $i \leq k$) is an exit place of Σ_{u_i}, and each p_j (for $j > k$) is an entry place of Σ_{u_j}.
  The label of q is that of s and the marking is equal to

  $$M_{\Sigma_{u_1}}(p_1) + \ldots + M_{\Sigma_{u_{k+m}}}(p_{k+m}) \ .$$

  There is an arc from q to a transition w iff $w = u_j$ (for some j) and there was an arc from p_j to t; moreover, in such a case, $\lambda_\Phi(q, w) = \lambda_{\Sigma_w}(p_j, t)$.[3]
  There is an arc from a transition w to q iff $w = u_j$ (for some j) and there was an arc from t to p_j.

As in the standard box algebra, we will use the following notations:

$$\begin{aligned}
\Omega_{\mathsf{sc}\,A}(\Sigma) &= \Sigma\ \mathsf{sc}\ A \\
\Omega_{\|}(\Sigma, \Sigma') &= \Sigma \| \Sigma' \\
\Omega_{;}(\Sigma, \Sigma') &= \Sigma\,\mathbf{;}\,\Sigma' \\
\Omega_{\Box}(\Sigma, \Sigma') &= \Sigma \Box \Sigma' \\
\Omega_{\circledast}(\Sigma, \Sigma', \Sigma'') &= \langle\!\langle \Sigma \circledast \Sigma' \circledast \Sigma'' \rangle\!\rangle \ .
\end{aligned}$$

B.2 Algebra of at-nets

The syntax (1) for *static arc-based time box expressions* (or static at-expressions), E, which correspond to at-nets with empty markings was given in the main body of the paper. Note that Z captures an auxiliary set of at-expressions such that

[3] Note that the definition is well-formed since the operand at-nets are ex-directed.

the corresponding at-nets are always ex-exclusive, and due to the standard box algebra theory, the at-nets corresponding to the expressions E are safe and clean (see proposition 4). The syntax for dynamic at-expressions (2) was also given in the main body of the paper. In addition, for any dynamic at-expression H, we denote by $\lfloor H \rfloor$, the static at expression obtained from H by removing all the overbars and underbars. And, any dynamic expression of the form $\overline{E}^{00}$ will be called *initial*.

Composite at-nets To be able to take advantage of the results developed for the standard box algebra, we introduce semantics of at-expressions into at-nets which are the same as that in the standard box algebra if we ignore all time annotations. The mapping Box from at-expressions to at-nets is defined so that

$$\mathsf{Box}(\alpha el) \stackrel{\mathrm{df}}{=} \mathsf{N}_{\alpha el}$$

where $\mathsf{N}_{\alpha el}$ is shown in figure 7,

$$\mathsf{Box}(\overline{E}^{\mathbb{EL}}) \stackrel{\mathrm{df}}{=} \overline{\mathsf{Box}(E)}$$
$$\mathsf{Box}(\underline{E}_{\mathbb{EL}}) \stackrel{\mathrm{df}}{=} \underline{\mathsf{Box}(E)}$$

and for remaining static of dynamic at-expressions:

$$\begin{aligned}
\mathsf{Box}(H \,\mathsf{sc}\, A) &\stackrel{\mathrm{df}}{=} \mathsf{Box}(H) \,\mathsf{sc}\, A \\
\mathsf{Box}(H \square J) &\stackrel{\mathrm{df}}{=} \mathsf{Box}(H) \square \mathsf{Box}(J) \\
\mathsf{Box}(H \| J) &\stackrel{\mathrm{df}}{=} \mathsf{Box}(H) \| \mathsf{Box}(J) \\
\mathsf{Box}(H \,;\, J) &\stackrel{\mathrm{df}}{=} \mathsf{Box}(H) \,;\, \mathsf{Box}(J) \\
\mathsf{Box}(\langle\!\langle H \circledast J \circledast I \rangle\!\rangle) &\stackrel{\mathrm{df}}{=} \langle\!\langle \mathsf{Box}(H) \circledast \mathsf{Box}(J) \circledast \mathsf{Box}(I) \rangle\!\rangle \ .
\end{aligned}$$

Any at-net obtained through the $\mathsf{Box}()$ from some at-expression will be called *composite*. Note that the above at-nets semantics of at-expressions are the standard black token semantics, with all time constraint being simply ignored.

Proposition 4. *For every static (or dynamic) at-expression H, $\mathsf{Box}(H)$ is a static (resp. dynamic) at-net which is both ex-directed and ex-restricted. Moreover, if H conforms to the syntax for Z or K then $\mathsf{Box}(H)$ is ex-exclusive.*

Proof. Follows from similar results in the standard box algebra. □

Transition based operational semantics of at-expressions To prove our main results, we will need another semantics of at-expressions, based on the transitions present in the corresponding composite at-nets. More precisely, at-expressions can perform two kinds of operational semantics moves, namely *transition* moves and *time* moves. A time move has the form

$$G \xrightarrow{\surd} H$$

and an action move has the form

$$G \xrightarrow{U} H$$

where $U = \{t_1, \ldots, t_k\}$ $(k \geq 0)$ is a set of transitions in the composite at-net $\mathsf{Box}(E)$, where E is obtained from G by deleting all overbars and underbars.

We now define various types of moves of the structural operational semantics of dynamic at-expressions (note that the relation $\equiv$ below is defined as in table 1).

Empty moves The following rules deal with the empty action moves.

$$\frac{G \equiv H}{G \xrightarrow{\varnothing} H} \qquad \frac{G \xrightarrow{\varnothing} J \xrightarrow{U} H}{G \xrightarrow{U} H} \qquad \frac{G \xrightarrow{U} J \xrightarrow{\varnothing} H}{G \xrightarrow{\Gamma} H}$$

Basic action A basic action can occur if its timing restrictions are satisfied by the age range of its overbar:

$$\frac{\mathbb{EL}\ \mathsf{tsat}\ el}{\overline{\alpha el}^{\mathbb{EL}} \xrightarrow{\{v_{\alpha el}\}} \underline{\alpha el}_{00}}$$

Scoping There is a single rule for scoping:

$$\frac{G \xrightarrow{\{t_1,u_1\}\uplus\cdots\uplus\{t_k,u_k\}\uplus U} H\ ,\ (\forall i) a_i = \widehat{c_i} \in A\ ,\ (A \cup \widehat{A}) \cap L = \varnothing}{G\,\mathsf{sc}\,A \xrightarrow{\{v_{\mathsf{sc}\,A} \lhd \{t_1,u_1\},\ldots,v_{\mathsf{sc}\,A} \lhd \{t_k,u_k\}\} \cup v_{\mathsf{sc}\,A} \blacktriangleleft U} H\,\mathsf{sc}\,A}$$

where $L = \lambda_{\mathsf{Box}(\lfloor G \rfloor)}(U)$, $a_i = \lambda_{\mathsf{Box}(\lfloor G \rfloor)}(t_i)$ and $c_i = \lambda_{\mathsf{Box}(\lfloor G \rfloor)}(u_i)$, for $i = 1, \ldots, k$.

Other operators There is no real difference in the rules for the remaining operators when compared with the standard box algebra [5, 6].

$$\frac{G \xrightarrow{U} G'\ ,\ H \xrightarrow{U'} H'}{G \| H \xrightarrow{v_{\|}^1 \blacktriangleleft U \cup v_{\|}^2 \blacktriangleleft U'} G' \| H'}$$

$$\frac{G \xrightarrow{U} H}{\begin{array}{c} \langle\!\langle G \circledast E \circledast F \rangle\!\rangle \xrightarrow{v_{\circledast}^1 \blacktriangleleft U} \langle\!\langle H \circledast E \circledast F \rangle\!\rangle \\ \langle\!\langle E \circledast G \circledast F \rangle\!\rangle \xrightarrow{v_{\circledast}^2 \blacktriangleleft U} \langle\!\langle E \circledast H \circledast F \rangle\!\rangle \\ \langle\!\langle E \circledast F \circledast G \rangle\!\rangle \xrightarrow{v_{\circledast}^3 \blacktriangleleft U} \langle\!\langle E \circledast F \circledast H \rangle\!\rangle \end{array}}$$

$$\frac{G \xrightarrow{U} H}{\begin{array}{c} G \Box E \xrightarrow{v_{\Box}^1 \blacktriangleleft U} H \Box E \\ E \Box G \xrightarrow{v_{\Box}^2 \blacktriangleleft U} E \Box H \end{array}}$$

$$\frac{G \xrightarrow{\Gamma} H}{\begin{array}{c} G\,;E \xrightarrow{v_{;}^1 \blacktriangleleft U} H\,;E \\ E\,;G \xrightarrow{v_{;}^2 \blacktriangleleft U} E\,;H \end{array}}$$

Urgent transitions of at-expressions Urgent transitions of dynamic at-expressions are defined by induction on their structure, as follows. For the base case, we have:

$$\begin{aligned}\mathsf{urgent}(\overline{\alpha el}^{\mathbb{EL}}) &\stackrel{\text{df}}{=} \begin{cases}\{v_{\alpha el}\} & \text{if } \mathbb{EL}\ \mathsf{tsat}\ el \text{ and } l = \mathbb{L}\\ \varnothing & \text{otherwise .}\end{cases}\\ \mathsf{urgent}(\underline{\alpha el}_{\mathbb{EL}}) &\stackrel{\text{df}}{=} \varnothing\end{aligned}$$

For more complicated expressions H, we define $\mathsf{urgent}(H)$ as the smallest set such that, whenever $H \equiv G$ then

$$\mathsf{urgent}(G) = \mathsf{urgent}(H)$$

and then the following hold for individual cases of composition operators. For scoping, if $v_{\mathsf{sc}\,A} \lhd U \in \mathsf{enabled}(G)$ and $U \cap \mathsf{urgent}(G) \neq \varnothing$ then:

$$v_{\mathsf{sc}\,A} \lhd U \in \mathsf{urgent}(G\,\mathsf{sc}\,A)\ .$$

Note: $\mathsf{enabled}(H)$ comprises all t such that there is an at-expression J satisfying

$$H \xrightarrow{\{t\}} J\ .$$

For the remaining operators, if $t \in \mathsf{urgent}(G)$ then:

$$\begin{aligned}v^1_{\|} \lhd t &\in \mathsf{urgent}(G\|J)\\ v^2_{\|} \lhd t &\in \mathsf{urgent}(J\|G)\\ v^1_{\circledast} \lhd t &\in \mathsf{urgent}(\langle\!\langle G \circledast E \circledast F\rangle\!\rangle)\\ v^2_{\circledast} \lhd t &\in \mathsf{urgent}(\langle\!\langle E \circledast G \circledast F\rangle\!\rangle)\\ v^3_{\circledast} \lhd t &\in \mathsf{urgent}(\langle\!\langle E \circledast F \circledast G\rangle\!\rangle)\\ v^1_{\Box} \lhd t &\in \mathsf{urgent}(G \Box E)\\ v^2_{\Box} \lhd t &\in \mathsf{urgent}(E \Box G)\\ v^1_{;} \lhd t &\in \mathsf{urgent}(G\,;E)\\ v^2_{;} \lhd t &\in \mathsf{urgent}(E\,;G)\ .\end{aligned}$$

Time moves There is a single time rule:

$$\boxed{\frac{\mathsf{urgent}(G) = \varnothing}{G \xrightarrow{\surd} G^{\surd}}}$$

Note that $\mathsf{urgent}(G)$ is the set of all transitions enabled by G but not by $G^{\surd}$ and, in fact, it could be defined like that. However, we preferred to give a definition closer to that used in the label based presentation in the main body of the paper. Note also that the example motivating a rather complicated definition of urgent labels there, $\overline{a00\Box a01}^{11}$, no longer works. The reason is that in case of the transition based semantics, the two a labels correspond to executing $v^1_\Box \lhd v_{a00}$ and $v^2_\Box \lhd v_{a01}$, respectively, and so they can be distinguished by the enabling relation.

It can be seen that the rules of the operational semantics do not lead outside the set of dynamic at-expressions.

Proposition 5. *Assuming that we treat the rules of the transition based operational semantics as term rewriting rules, and H has been derived from an at-expression, then H is also an at-expression.*

Proof. Follows from a similar result in the standard box algebra. □

Representing global behaviour of at-expressions There are different, though closely related, representation capturing the overall behaviour of an at-expression H. The first one we already introduced is that of reachability tree, RT_H. We will also need the following.

- A *full reachability tree* of a dynamic at-expression H, denoted by fRT_H, is a tree whose nodes are labelled by equivalence classes of dynamic expressions reachable from H using the rules defined in this section, and arcs are labelled by steps of transitions or the $\surd$ symbol. The root node is labelled by $[H]_\equiv$ and, if a node is labelled by $[G]_\equiv$, then: for every move

$$G \xrightarrow{U} J\,,$$

  there is a unique descendant labelled by $[J]_\equiv$ and the arc leading to it is labelled by U, and if the time move is possible for G then there is a unique descendant labelled by $[G^\surd]_\equiv$ and the arc leading to it is labelled by $\surd$. For a static at-expression H, $\mathsf{fRT}_H \stackrel{\mathrm{df}}{=} \mathsf{fRT}_{\overline{H}^{00}}$.
- Let H be a dynamic at-expression. We will use $[H\rangle$ to denote all the at-expressions derivable from H using the operational semantics defined in this section, i.e., the least set of expressions containing H such that if $H' \in [H\rangle$ and $H' \xrightarrow{U} H''$, for some step U of transitions in $[\![\mathsf{cBox}(H)]\!]$, then $H'' \in [H\rangle$. Moreover, $[H]_\equiv$ will denote the equivalence class of $\equiv$ containing H.
  The *full transition system* of H is then defined as $\mathsf{fTS}_H \stackrel{\mathrm{df}}{=} (V, \mathit{Arcs}, \mathit{init})$, where $V \stackrel{\mathrm{df}}{=} \{[H']_\equiv \mid H' \in [H\rangle\}$ is the set of states with $\mathit{init} \stackrel{\mathrm{df}}{=} [H]_\equiv$ being the initial state, and Arcs is the set of labelled arcs of the form $([H']_\equiv, U, [H'']_\equiv)$ such that $H', H'' \in [H\rangle$ and $H' \xrightarrow{U} H''$.
  For a static at-expression H, $\mathsf{fTS}_H \stackrel{\mathrm{df}}{=} \mathsf{fTS}_{\overline{H}^{00}}$.
- Let H be a dynamic at-expression. We will use $[H\rangle_{lab}$ to denote all the at-expressions derivable from H using the operational semantics introduced in the main body of the paper, i.e., the least set of expressions containing H such that if $H' \in [H\rangle_{lab}$ and $H' \xrightarrow{\Gamma} H''$, for some multiset of communication labels Γ, then $H'' \in [H\rangle_{lab}$.

The *transition system* of H is then defined as $\mathsf{TS}_H \stackrel{\mathrm{df}}{=} (V, \mathit{Arcs}, \mathit{init})$, where $V \stackrel{\mathrm{df}}{=} \{[H']_\equiv \mid H' \in [H\rangle_{lab}\}$ is the set of states with $\mathit{init} \stackrel{\mathrm{df}}{=} [H]_\equiv$ being the initial state, and Arcs is the set of labelled arcs of the form $([H']_\equiv, \Gamma, [H'']_\equiv)$ such that $H', H'' \in [H\rangle_{lab}$ and $H' \stackrel{\Gamma}{\longrightarrow} H''$.
For a static at-expression H, $\mathsf{TS}_H \stackrel{\mathrm{df}}{=} \mathsf{TS}_{\overline{H}^{\circ\circ}}$.

B.3 Interface regions

The standard boxes have quite regular internal structure which then has a significant impact on their behaviour. We will capture some aspects of this structure through the notion of interface regions, which will form a partition of the set of internal places.

The set of *interface regions* $\mathbb{IR}(\Sigma)$ of a composite at-net Σ is defined by induction on the structure of the at-net, in the following way.

Basic Net: $\Sigma = \mathsf{N}_{\alpha el}$. Then $\mathbb{IR}(\Sigma) \stackrel{\mathrm{df}}{=} \varnothing$.

Parallel composition: $\Sigma = \Sigma_1 \| \Sigma_2$. Then

$$\mathbb{IR}(\Sigma) \stackrel{\mathrm{df}}{=} \bigcup_{k=1}^{2} \{v_{\|}^{k} \blacktriangleleft Q \mid Q \in \mathbb{IR}(\Sigma_k)\} .$$

Sequential composition: $\Sigma = \Sigma_1 \,;\, \Sigma_2$. Then

$$\mathbb{IR}(\Sigma) \stackrel{\mathrm{df}}{=} \{i_{;} \triangleleft (v_{;}^{1} \blacktriangleleft \Sigma_1^{\circ}, v_{;}^{2} \blacktriangleleft {}^{\circ}\Sigma_2)\} \cup \bigcup_{k=1}^{2} \{v_{;}^{k} \blacktriangleleft Q \mid Q \in \mathbb{IR}(\Sigma_k)\} .$$

Choice operator: $\Sigma = \Sigma_1 \Box \Sigma_2$. Then

$$\mathbb{IR}(\Sigma) \stackrel{\mathrm{df}}{=} \bigcup_{k=1}^{2} \{v_{\Box}^{k} \blacktriangleleft Q \mid Q \in \mathbb{IR}(\Sigma_k)\} .$$

Iteration: $\Sigma = \langle\!\langle \Sigma_1 \circledast \Sigma_2 \circledast \Sigma_3 \rangle\!\rangle$. Then

$$\mathbb{IR}(\Sigma) \stackrel{\mathrm{df}}{=} \{i_{\circledast} \triangleleft (v_{\circledast}^{1} \blacktriangleleft \Sigma_1^{\circ}, v_{\circledast}^{2} \blacktriangleleft {}^{\circ}\Sigma_2, v_{\circledast}^{2} \blacktriangleleft \Sigma_2^{\circ}, v_{\circledast}^{3} \blacktriangleleft {}^{\circ}\Sigma_3)\} \;\cup$$

$$\bigcup_{k=1}^{3} \{v_{\circledast}^{k} \blacktriangleleft Q \mid Q \in \mathbb{IR}(\Sigma_k)\} .$$

Scoping: $\Sigma = \Sigma_1 \, \mathsf{sc} \, A$. Then $\mathbb{IR}(\Sigma) \stackrel{\mathrm{df}}{=} \{v_{\mathsf{sc}\, A} \blacktriangleleft Q \mid Q \in \mathbb{IR}(\Sigma_1)\}$.

Proposition 6. *Let Σ be a composite at-net. Then*

$$\ddot{\Sigma} = \biguplus_{Q \in \mathbb{IR}(\Sigma)} Q \; .$$

Proof. Follows by a straightforward induction on the way Σ has been constructed. □

A crucial property of an interface region is that its marking behaves in a monotone way, as captured by the following result.

Proposition 7. *Let Σ be an initial composite at-net, $Q \in \mathbb{IR}(\Sigma)$ one of its interface regions, and $M_1 U_1 M_2 U_2 \ldots M_n U_n M_{n+1}$ be a sequence of markings and steps such that $M_1 = M_\Sigma = {}^\circ\Sigma$ and $M_i[U_i\rangle M_{i+1}$, for $i = 1, \ldots, n$. Moreover, let $M'_i = M_i \cap Q$, for $i = 1, \ldots, n+1$.*

1. *There are indices $k_1 < k_2 < \ldots < k_m$ such that $k_1 = 1$, $k_m = n+2$ and, for each $j < m$, one of the following holds:*

$$\begin{array}{ll} \textit{Case 1:} & \varnothing = M'_{k_j} \subseteq M'_{k_j+1} \subseteq \cdots \subseteq M'_{k_{j+1}-1} \; . \\ \textit{Case 2:} & Q = M'_{k_j} \supseteq M'_{k_j+1} \supseteq \cdots \supseteq M'_{k_{j+1}-1} \; . \end{array}$$

   Moreover the two cases strictly alternate, beginning with Case 1.
2. *If M'_i occurs in Case 1 sequence then ${}^\bullet U_i \cap Q = \varnothing$, and otherwise $U_i^\bullet \cap Q = \varnothing$.*

Proof. (1) This is a property of the standard box algebra. It can be shown, for instance, by considering the isomorphism between the reachability graphs of such boxes and the the corresponding process expressions. One also needs the following property $\varnothing \in \{{}^\bullet U_i \cap Q, U_i^\bullet \cap Q\}$, for all i, which holds due to the syntaxes (1,2); in particular, since the way in which the syntax for Z was given guarantees that the corresponding at-net is ex-exclusive.

(2) Follows directly from part (1) and the above property. □

B.4 Clusters

For every composite at-net Σ, its *clusters* are defined as:

$$CL(\Sigma) \stackrel{\mathrm{df}}{=} \{{}^\circ\Sigma, \Sigma^\circ\} \cup \mathsf{cl_e}(\Sigma) \cup \mathsf{cl_i}(\Sigma),$$

where the entry clusters $\mathsf{cl_e}(\Sigma)$, and the internal clusters $\mathsf{cl_i}(\Sigma)$, are defined compositionally below.

Basic Net: $\Sigma = \mathsf{N}_{\alpha el}$. Then $\mathsf{cl_e}(\Sigma) \stackrel{\mathrm{df}}{=} \{{}^\circ\Sigma\}$ and $\mathsf{cl_i}(\Sigma) \stackrel{\mathrm{df}}{=} \varnothing$.

Parallel composition: $\Sigma = \Sigma_1 \| \Sigma_2$. Then

$$\begin{aligned}\mathsf{cl_e}(\Sigma) &\stackrel{\mathrm{df}}{=} \bigcup_{k=1}^{2} \{e_{\|}^{k} \lhd v_{\|}^{k} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_k)\}\\ \mathsf{cl_i}(\Sigma) &\stackrel{\mathrm{df}}{=} \bigcup_{k=1}^{2} \{v_{\|}^{k} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_i}(\Sigma_k)\}\ .\end{aligned}$$

Sequential composition: $\Sigma = \Sigma_1 \,\mathbf{;}\, \Sigma_2$. Then

$$\mathsf{cl_e}(\Sigma) \stackrel{\mathrm{df}}{=} \{e_{;} \lhd v_{;}^{1} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_1)\}$$

$$\mathsf{cl_i}(\Sigma) \stackrel{\mathrm{df}}{=} \bigcup_{k=1}^{2} \{v_{;}^{k} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_i}(\Sigma_k)\} \cup \{i_{;} \lhd (v_{;}^{1} \blacktriangleleft \Sigma_1^{\circ}, v_{;}^{2} \blacktriangleleft \mathsf{cl}) \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_2)\}\ .$$

Choice operator: $\Sigma = \Sigma_1 \Box \Sigma_2$. Then

$$\begin{aligned}\mathsf{cl_e}(\Sigma) \stackrel{\mathrm{df}}{=}\ & \{e_{\Box} \lhd (v_{\Box}^{1} \blacktriangleleft \mathsf{cl}, v_{\Box}^{2} \blacktriangleleft {}^{\circ}\Sigma_2 \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_1)\}\ \cup\\ & \{e_{\Box} \lhd (v_{\Box}^{1} \blacktriangleleft {}^{\circ}\Sigma_1, v_{\Box}^{2} \blacktriangleleft \mathsf{cl}) \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_2)\}\end{aligned}$$

$$\mathsf{cl_i}(\Sigma) \stackrel{\mathrm{df}}{=} \bigcup_{k=1}^{2} \{v_{\Box}^{k} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_i}(\Sigma_k)\}\ .$$

Iteration: $\Sigma = \langle\!\langle \Sigma_1 \circledast \Sigma_2 \circledast \Sigma_3 \rangle\!\rangle$. Then

$$\mathsf{cl_e}(\Sigma) \stackrel{\mathrm{df}}{=} \{e_{\circledast} \lhd v_{\circledast}^{1} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_1)\}$$

$$\begin{aligned}\mathsf{cl_i}(\Sigma) \stackrel{\mathrm{df}}{=}\ & \bigcup_{k=1}^{3} \{v_{\circledast}^{k} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_i}(\Sigma_k)\}\ \cup\\ & \{i_{\circledast} \lhd (v_{\circledast}^{1} \blacktriangleleft \Sigma_1^{\circ}, v_{\circledast}^{2} \blacktriangleleft \mathsf{cl}, v_{\circledast}^{2} \blacktriangleleft \Sigma_2^{\circ}, v_{\circledast}^{3} \blacktriangleleft {}^{\circ}\Sigma_3) \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_2)\}\ \cup\\ & \{i_{\circledast} \lhd (v_{\circledast}^{1} \blacktriangleleft \Sigma_1^{\circ}, v_{\circledast}^{2} \blacktriangleleft {}^{\circ}\Sigma_2, v_{\circledast}^{2} \blacktriangleleft \Sigma_2^{\circ}, v_{\circledast}^{3} \blacktriangleleft \mathsf{cl}) \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_3)\}\ .\end{aligned}$$

Scoping: $\Sigma = \Sigma_1\, \mathsf{sc}\, A$. Then

$$\begin{aligned}\mathsf{cl_e}(\Sigma) &\stackrel{\mathrm{df}}{=} \{e_{\mathsf{sc}\,A} \lhd v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_e}(\Sigma_1)\}\\ \mathsf{cl_i}(\Sigma) &\stackrel{\mathrm{df}}{=} \{v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in \mathsf{cl_i}(\Sigma_1)\}\ .\end{aligned}$$

Proposition 8. *Let Σ be a composite at-net. If $\mathsf{cl} \in \mathsf{cl_e}(\Sigma)$ then $\mathsf{cl} \subseteq {}^{\circ}\Sigma$, and if $\mathsf{cl} \in \mathsf{cl_i}(\Sigma)$ then $\mathsf{cl} \subseteq \ddot{\Sigma}$. Moreover, in the latter case, there is a unique interface region $Q \in \mathbb{IR}(\Sigma)$ such that $\mathsf{cl} \subseteq Q$.*

Proof. Follows from the definitions of net refinement and clusters, by a straightforward induction on the syntax of the expression from which Σ has been generated. The uniqueness property follows from proposition 6. □

Proposition 9. *Let Σ be an initial composite at-net, $\mathsf{cl} \in \mathsf{cl_i}(\Sigma)$ be one of its internal clusters, and $M_1 U_1 M_2 U_2 \dots M_n U_n M_{n+1}$ be a sequence of markings and steps such that $M_1 = M_\Sigma = {}^\circ\Sigma$ and $M_i[U_i\rangle M_{i+1}$, for $i = 1, \dots, n$. Moreover, let $M'_i = M_i \cap \mathsf{cl}$, for $i = 1, \dots, n+1$.*
Then there are indices $k_1 < k_2 < \dots < k_m$ such that $k_1 = 1$, $k_m = n+2$ and, for each $j < m$, one of the following holds:

$$\begin{array}{ll} \textit{Case 1:} & \varnothing = M'_{k_j} \subseteq M'_{k_j+1} \subseteq \dots \subseteq M'_{k_{j+1}-1} \; . \\ \textit{Case 2:} & Q = M'_{k_j} \supseteq M'_{k_j+1} \supseteq \dots \supseteq M'_{k_{j+1}-1} \; . \end{array}$$

Moreover the two cases strictly alternate, beginning with Case 1.

Proof. Follows from propositions 7 and 8. □

B.5 Pre-clusters of a transition

For every composite at-net Σ and a transition $t \in T_\Sigma$, the *pre-clusters* of t are defined compositionally below.

Basic Net: $\Sigma = \mathsf{N}_{\alpha el}$. Then, for $t = v_{\alpha el}$, ${}^\diamond t \overset{\text{df}}{=} \{{}^\circ\Sigma\}$.

Parallel composition: $\Sigma = \Sigma_1 \| \Sigma_2$. Then, for $t = v_\|^k \lhd u$ $(k = 1, 2)$:

$${}^\diamond t \overset{\text{df}}{=} \{e_\| \lhd v_\|^k \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_k)\} \cup \{v_\|^k \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_i}(\Sigma_k)\} \; .$$

Sequential composition: $\Sigma = \Sigma_1 \,;\, \Sigma_2$. Then, for $t = v_;^1 \lhd u$:

$${}^\diamond t \overset{\text{df}}{=} \{e_; \lhd v_;^1 \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_1)\} \cup \{v_;^1 \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_i}(\Sigma_1)\}$$

and for $t = v_;^2 \lhd u$:

$$\begin{aligned} {}^\diamond t \overset{\text{df}}{=} & \{i_; \lhd (v_;^1 \blacktriangleleft \Sigma_1^\circ, v_;^2 \blacktriangleleft \mathsf{cl}) \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_2)\} \;\cup \\ & \{v_;^2 \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_i}(\Sigma_2)\} \; . \end{aligned}$$

Choice: $\Sigma = \Sigma_1 \Box \Sigma_2$. Then, for $t = v_\Box^1 \lhd u$:

$$\begin{aligned} {}^\diamond t \overset{\text{df}}{=} & \{e_\Box \lhd (v_\Box^1 \blacktriangleleft \mathsf{cl}, v_\Box^2 \blacktriangleleft {}^\circ\Sigma_2) \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_1)\} \;\cup \\ & \{v_\Box^1 \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_i}(\Sigma_1)\} \end{aligned}$$

and for $t = v_\Box^2 \lhd u$:

$$\begin{aligned} {}^\diamond t \overset{\text{df}}{=} & \{e_\Box \lhd (v_\Box^1 \blacktriangleleft {}^\circ\Sigma_1, v_\Box^2 \blacktriangleleft \mathsf{cl}) \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_2)\} \;\cup \\ & \{v_\Box^2 \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\diamond u \cap \mathsf{cl_i}(\Sigma_2)\} \; . \end{aligned}$$

Iteration: $\Sigma = \langle\!\langle \Sigma_1 \circledast \Sigma_2 \circledast \Sigma_3 \rangle\!\rangle$. Then, for $t = v^1_\circledast \lhd u$:

$$^\Diamond t \stackrel{\text{df}}{=} \{e_\circledast \lhd v^1_\circledast \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\Diamond u \cap \mathsf{cl_e}(\Sigma_1)\} \cup \{v^1_\circledast \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\Diamond u \cap \mathsf{cl_i}(\Sigma_1)\}$$

for $t = v^2_\circledast \lhd u$:

$$^\Diamond t \stackrel{\text{df}}{=} \{i_\circledast \lhd (v^1_\circledast \blacktriangleleft \Sigma_1^\circ, v^2_\circledast \blacktriangleleft \mathsf{cl}, v^2_\circledast \blacktriangleleft \Sigma_2^\circ, v^3_\circledast \blacktriangleleft {}^\circ\Sigma_3) \mid \mathsf{cl} \in {}^\Diamond u \cap \mathsf{cl_e}(\Sigma_2)\} \;\cup$$
$$\{v^2_\circledast \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\Diamond u \cap \mathsf{cl_i}(\Sigma_2)\}$$

and for $t = v^3_\circledast \lhd u$:

$$^\Diamond t \stackrel{\text{df}}{=} \{i_\circledast \lhd (v^1_\circledast \blacktriangleleft \Sigma_1^\circ, v^2_\circledast \blacktriangleleft {}^\circ\Sigma_2, v^2_\circledast \blacktriangleleft \Sigma_2^\circ, v^3_\circledast \blacktriangleleft \mathsf{cl}) \mid \mathsf{cl} \in {}^\Diamond u \cap \mathsf{cl_e}(\Sigma_3)\} \;\cup$$
$$\{v^3_\circledast \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\Diamond u \cap \mathsf{cl_i}(\Sigma_3)\} \;.$$

Scoping: $\Sigma = \Sigma_1 \,\mathsf{sc}\, A$. Then, for $t = v_{\mathsf{sc}\,A} \lhd u$:

$$^\Diamond t \stackrel{\text{df}}{=} \{e_{\mathsf{sc}\,A} \lhd v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\Diamond u \cap \mathsf{cl_e}(\Sigma_1)\} \cup \{v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in {}^\Diamond u \cap \mathsf{cl_i}(\Sigma_1)\}$$

and for $t = v_{\mathsf{sc}\,A} \lhd \{u, w\}$:

$$^\Diamond t \stackrel{\text{df}}{=} \{e_{\mathsf{sc}\,A} \lhd v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in ({}^\Diamond u \cup {}^\Diamond w) \cap \mathsf{cl_e}(\Sigma_1)\} \;\cup$$
$$\{v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl} \mid \mathsf{cl} \in ({}^\Diamond u \cup {}^\Diamond w) \cap \mathsf{cl_i}(\Sigma_1)\} \;.$$

Proposition 10. *Let Σ be a composite at-net, $t \in T_\Sigma$ and $\mathsf{cl} \in {}^\Diamond t$. Then $\mathsf{cl} \subseteq {}^\bullet t$ and $\lambda_\Sigma(p, t) = \lambda_\Sigma(q, t)$, for all $p, q \in \mathsf{cl}$.*

Proof. See appendix G. □

Proposition 11. *Let Σ be a composite at-net, $t \in T_\Sigma$ and $p \in {}^\bullet t$. Then there is $\mathsf{cl} \in {}^\Diamond t$ such that $p \in \mathsf{cl}$.*

Proof. Follows by induction on the structure of the expression from which Σ has been derived, similarly as proposition 10. □

C Token-based arc-time Petri nets (at-boxes)

An *at-box* is a pair $\mathbf{\Theta} \stackrel{\text{df}}{=} (\Sigma, \mu)$ such that $\Sigma = \mathsf{box}(J)$, for some static or dynamic at-expression J given by the syntax (1,2) and

$$\mu : P_\Sigma \to \mathbb{N}^\perp$$

is a *token timing* mapping (a state) such that the following consistency conditions hold:

- For every $p \in P_\Sigma$, $\mu(p) = \perp$ iff $M_\Sigma(p) = 0$.
- For all $p, p' \in {}^\circ\Sigma$, if $\mu(p) \neq \perp \neq \mu(p')$ then $\mu(p) = \mu(p')$.

We say that $\mathbf{\Theta}$ is static/dynamic if so is J and denote $\mathbf{\Theta} \in \mathfrak{T}_J$. We then introduce some useful notations:

- $[\![\mathbf{\Theta}]\!] \stackrel{\mathrm{df}}{=} \Sigma$ and $\lfloor \mathbf{\Theta} \rfloor \stackrel{\mathrm{df}}{=} (\lfloor \Sigma \rfloor, \nu)$, where ν always returns $\perp$.
- The state $\mu^{\surd}$ is defined so that, for every $p \in P_\Sigma$,

$$\mu^{\surd}(p) \stackrel{\mathrm{df}}{=} \begin{cases} \mu(p) + 1 & \text{if } \mu(p) \neq \perp \\ \perp & \text{otherwise} \end{cases}$$

  and the at-box $\mathbf{\Theta}^{\surd}$ is then defined as $(\Sigma, \mu^{\surd})$.
- $\mathbf{\Theta}$ is *input-reachable* if it is reachable from the at-box $(\overline{\lfloor \Sigma \rfloor}, \nu)$, where ν returns 0 for all the entry places, and otherwise $\perp$. We will be interested only in those at-boxes which are input-reachable.

The above notions are well-defined. Indeed, it is clear that the two consistency conditions are satisfied in each case.

Proposition 12. *Let $\mathbf{\Theta}$ be an at-box in $\mathfrak{T}_J$.*

1. *$\lfloor \mathbf{\Theta} \rfloor$ is a static at-box in $\mathfrak{T}_{\lfloor J \rfloor}$.*
2. *If $\mathbf{\Theta}$ is static, then $\lfloor \mathbf{\Theta} \rfloor = \mathbf{\Theta}$.*

Proof. Follows from the properties of the standard box algebra. □

A set of transitions $U \subseteq T_\Sigma$ is *enabled* by $\mathbf{\Theta}$ if it is enabled by Σ and, for every $t \in U$ and every place $p \in {}^\bullet t$, we have that $\mu(p)$ tsat $\lambda_\Sigma(p,t)$. We denote this by $U \in \mathsf{enabled}(\mathbf{\Theta})$. This enabling is *urgent*, denoted $U \in \mathsf{urgent}(\mathbf{\Theta})$, if U is not enabled by $\mathbf{\Theta}^{\surd}$.

An enabled step may be *executed* and yield a *follower* at-box $\mathbf{\Xi} = (\Sigma', \nu)$ such that $\Sigma[U\rangle\Sigma'$ and, for every place $p \in P_\Sigma$,

$$\nu(p) \stackrel{\mathrm{df}}{=} \begin{cases} \perp & \text{if } p \in {}^\bullet U \\ 0 & \text{if } p \in U^\bullet \\ \mu(p) & \text{otherwise .} \end{cases}$$

We denote this by $\mathbf{\Theta}[U\rangle\mathbf{\Xi}$. Note that due to proposition 7 and the ex-directedness of Σ, we do need to consider the case when $p \in {}^\bullet U \cap U^\bullet$. A similar comment applies also to the formula for marking execution in the cat-boxes introduced in the next section.

A time move is enabled if there is no urgent enabled step; it then can be executed and yield a follower at-box: $\mathbf{\Theta}[\surd\rangle\mathbf{\Theta}^{\surd}$.

Proposition 13. *Let $\mathbf{\Theta}$ be an at-box and $\mathbf{\Theta}[U\rangle\mathbf{\Xi}$ or $\mathbf{\Theta}[\surd\rangle\mathbf{\Xi}$.*

1. *If $\mathbf{\Theta}$ is static, then $U = \varnothing$ and $\mathbf{\Theta} = \mathbf{\Xi}$.*
2. *If $\mathbf{\Theta}$ is dynamic then so is $\mathbf{\Xi}$.*

Proof. Follows from the properties of the standard box algebra and, additionally, we need to check that the two consistency conditions from the definition of at-boxes are satisfied. The latter is straightforward (ex-directedness of at-nets is important here). □

Proposition 14. *Let $\mathbf{\Theta}$ be an input-reachable at-box, $\mathbf{\Theta}[U\rangle\mathbf{\Xi}$, where U is a step consisting of transitions $t_1,\ldots,t_k$. Then there are at-boxes $\mathbf{\Theta}_0,\ldots,\mathbf{\Theta}_k$ such that $\mathbf{\Theta}_0=\mathbf{\Theta}$, $\mathbf{\Theta}_k=\mathbf{\Xi}$ and $\mathbf{\Theta}_{i-1}[t_i\rangle\mathbf{\Theta}_i$, for $i=1,\ldots,k$.*

Proof. Follows from the standard properties of safe Petri nets and proposition 7 which ensures that for each t_i no time token involved in the enabling of t_i is involved in the firing of the preceding transitions $t_1,\ldots,t_{i-1}$. □

Representing global behaviour of at-boxes As in the case of at-expressions, there are different representation capturing the overall behaviour of an at-box $\mathbf{\Theta}$. The first one we already introduced is that of reachability tree, $\mathsf{RT}_{\mathbf{\Theta}}$. We will also need the following.

– A *full reachability tree* of an at-box $\mathbf{\Theta}=(\Sigma,\mu)$, denoted by $\mathsf{fRT}_{\mathbf{\Theta}}$, has nodes labelled by token timings and arcs annotated by executed transition steps or time moves. More precisely, the root node is labelled by the initial token timing μ and, if a node is labelled by μ', then for every move $\mu'[x\rangle\mu''$ there is a unique descendant labelled by μ''; the arc leading to it is labelled by $\surd$ if $x=\surd$, and by U if $x=U$ is an executed transition step.
– A *full transition system* of an at-box $\mathbf{\Theta}$ is $\mathsf{fTS}_{\mathbf{\Theta}} \stackrel{\text{df}}{=} (V, \mathit{Arcs}, \mathit{init})$, where $V \stackrel{\text{df}}{=} [\mathbf{\Theta}\rangle$ is the set of states with $\mathit{init} \stackrel{\text{df}}{=} \mathbf{\Theta}$ being the initial state, and Arcs is the set of all labelled arcs of the form $(\mathbf{\Theta}',U,\mathbf{\Theta}'')$ and $(\mathbf{\Theta}',\surd,\mathbf{\Theta}'')$ such that $\mathbf{\Theta}',\mathbf{\Theta}''\in[\mathbf{\Theta}\rangle$ and, respectively, $\mathbf{\Theta}'[U\rangle\mathbf{\Theta}''$ and $\mathbf{\Theta}'[\surd\rangle\mathbf{\Theta}''$.
– A *transition system* of an at-box $\mathbf{\Theta}$, denoted by $\mathsf{ts}_{\mathbf{\Theta}}$, is obtained from $\mathsf{fTS}_{\mathbf{\Theta}}$ by replacing each arc $\mathbf{\Theta}'[U\rangle\mathbf{\Theta}''$ by $\mathbf{\Theta}'[\Gamma\rangle\mathbf{\Theta}''$, where *Gamma* is the multiset of communication labels of the transitions in U.

D Cluster-based arc-time Petri nets (cat-boxes)

We now introduce an auxiliary algebra of arc-timed boxes which will serve as a bridge between at-boxes and at-expressions.

A *cluster at-box* (or cat-box) is a pair $\mathfrak{A} \stackrel{\text{df}}{=} (\Sigma,\mathcal{M})$ such that $\Sigma=\mathsf{box}(J)$, for some static or dynamic at-expression given by the syntax (1,2) and

$$\mathcal{M}: CL_\Sigma \to \mathbb{D}$$

is a *cluster filling* (state) such that the following *consistency conditions* hold:

– For every cl in CL_Σ, $\mathcal{M}(\mathsf{cl})=\bot$ iff $M_\Sigma(\mathsf{cl})=\{0\}$.
– For all cl and cl' in $\{{}^\circ\Sigma\}\cup\mathsf{cl_e}$, if $\mathcal{M}(\mathsf{cl})\neq\bot\neq\mathcal{M}(\mathsf{cl}')$ then $\mathcal{M}(\mathsf{cl})=\mathcal{M}(\mathsf{cl}')$.

We say that $\mathfrak{A}$ is static/dynamic if so is J and denote $\mathfrak{A}\in\mathfrak{T}_J$. We then introduce some useful notations:

– $[\![\mathfrak{A}]\!] \stackrel{\text{df}}{=} \Sigma$ and $\lfloor\mathfrak{A}\rfloor \stackrel{\text{df}}{=} (\lfloor\Sigma\rfloor,\mathcal{N})$, where $\mathcal{N}$ always returns $\bot$.

– For every transition $t \in T_\Sigma$ and cluster $\mathsf{cl} \in {}^\diamond t$,

$$\lambda(\mathsf{cl}, t) \stackrel{\text{df}}{=} \lambda_\Sigma(p, t) \ ,$$

for any $p \in \mathsf{cl}$.

– The state $\mathcal{M}^\surd$ is defined so that, for every cluster cl in Σ,

$$\mathcal{M}^\surd(\mathsf{cl}) \stackrel{\text{df}}{=} \begin{cases} (\mathbb{E}+1)(\mathbb{L}+1) & \text{if } \mathcal{M}(\mathsf{cl}) = \mathbb{E}\mathbb{L} \\ \bot & \text{otherwise} \end{cases}$$

and the cat-box $\mathfrak{A}^\surd$ is then defined as $(\Sigma, \mathcal{M}^\surd)$.

The above notions are well-defined. This is immediate in all but one case, namely $\lambda(\mathsf{cl}, t)$ is well-defined by proposition 10.

Proposition 15. *Let $\mathfrak{A}$ be a cat-box in $\mathfrak{T}_J$.*

1. *$\lfloor \mathfrak{A} \rfloor$ is a static cat-box in $\mathfrak{T}_{\lfloor J \rfloor}$.*
2. *If $\mathfrak{A}$ is static, then $\lfloor \mathfrak{A} \rfloor = \mathfrak{A}$.*

Proof. Follows from the properties of the standard box algebra. □

A set of transitions $U \subseteq T_\Sigma$ is *enabled* by $\mathfrak{A}$ if it is enabled by Σ and, for every transition $t \in U$ and every cluster $\mathsf{cl} \in {}^\diamond t$, we have that $\mathcal{M}(\mathsf{cl})\ \mathsf{tsat}\ \lambda(\mathsf{cl}, t)$. We denote this by $U \in \mathsf{enabled}(\mathfrak{A})$. This enabling is *urgent*, denoted $U \in \mathsf{urgent}(\mathfrak{A})$, if U is not enabled by $\mathfrak{A}^\surd$.

An enabled step may be *executed* and yield a *follower* cat-box $\mathfrak{X} = (\Sigma', \mathcal{N})$ such that $\Sigma[U\rangle\Sigma'$ and, for every cluster cl in Σ,

$$\mathcal{N}(\mathsf{cl}) \stackrel{\text{df}}{=} \begin{cases} \bot & \text{if } M_{\Sigma'} \cap \mathsf{cl} = \varnothing \\ 0\mathbb{L} & \text{if } \mathsf{cl} \cap U^\bullet \neq 0 \text{ and } \mathcal{M}(\mathsf{cl}) = \mathbb{E}\mathbb{L} \\ 00 & \text{if } \mathsf{cl} \cap U^\bullet \neq 0 \text{ and } \mathcal{M}(\mathsf{cl}) = \bot \\ \mathcal{M}(\mathsf{cl}) & \text{otherwise .} \end{cases}$$

We denote this by $\mathfrak{A}[U\rangle\mathfrak{X}$.

A time move is enabled if there is no urgent enabled step; it then can be executed and yield a follower cat-box: $\mathfrak{A}[\surd\rangle\mathfrak{A}^\surd$.

Proposition 16. *Let $\mathfrak{A}$ be a cat-box and $\mathfrak{A}[U\rangle\mathfrak{X}$ or $\mathfrak{A}[\surd\rangle\mathfrak{X}$.*

1. *If $\mathfrak{A}$ is static, then $U = \varnothing$ and $\mathfrak{A} = \mathfrak{X}$.*
2. *If $\mathfrak{A}$ is dynamic then so is $\mathfrak{X}$.*

Proof. Follows from the properties of the standard box algebra and, additionally, we need to check that the two consistency conditions from the definition of at-boxes are satisfied. The latter is straightforward (ex-directedness of at-nets is again important here). □

Proposition 17. *Let $\mathfrak{A}[U\rangle\mathfrak{X}$, where $U = \{t_1, \ldots, t_k\}$. Then there are cat-boxes $\mathfrak{A}_0, \ldots, \mathfrak{A}_k$ such that $\mathfrak{A}_0 = \mathfrak{A}$, $\mathfrak{A}_k = \mathfrak{X}$ and $\mathfrak{A}_{i-1}[t_i\rangle\mathfrak{A}_i$, for $i = 1, \ldots, k$.*

Proof. Follows from the standard properties of safe Petri nets and proposition 9 which ensures that for each t_i no time token involved in the enabling of t_i is involved in the firing of the preceding transitions $t_1, \ldots, t_{i-1}$. □

Representing global behaviour of cat-boxes As for at-boxes, we have four different ways of capturing the overall behaviour of cat-boxes, namely $\mathsf{RT}_{\mathfrak{A}}$, $\mathsf{fRT}_{\mathfrak{A}}$ $\mathsf{TS}_{\mathfrak{A}}$ and $\mathsf{fTS}_{\mathfrak{A}}$. Their definitions are a straightforward adaptation of those for at-boxes.

D.1 An algebra of cat-boxes

We define an algebra of cat-boxes following the syntax (1,2). To start with, the basic at-box $\mathsf{N}^{\mathsf{at}}_{\alpha el} = (\mathsf{N}_{\alpha el}, \mathcal{M})$, where for every cluster $\mathsf{cl} \in CL_{\mathsf{N}_{\alpha el}}$ we have:

$$\mathcal{M}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \bot \tag{3}$$

is a basic building block of the algebra. In what now follows, we assume that $\mathfrak{A} = (\Sigma, \mathcal{M}) \in \mathfrak{T}_H$, $\mathfrak{X} = (\Psi, \mathcal{N}) \in \mathfrak{T}_J$ and $\mathfrak{V} = (\Phi, \mathcal{P}) \in \mathfrak{T}_K$ are cat-boxes.

Overbarring and underbarring: If H is a static at-expression and $\mathbb{EL} \in \mathbb{D}$, then $\overline{\mathfrak{A}}^{\mathbb{EL}} = (\overline{\Sigma}, \mathcal{R}) \in \mathfrak{T}_{\overline{H}^{\mathbb{EL}}}$ where, for every cluster $\mathsf{cl} \in CL_{\Sigma}$, we have:

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \mathbb{EL} & \text{if } \mathsf{cl} \in \mathsf{cl_e}(\Sigma) \ \text{ or } \ \mathsf{cl} = {}^{\circ}\Sigma \\ \bot & \text{otherwise .} \end{cases} \tag{4}$$

Similarly, $\underline{\mathfrak{A}}_{\mathbb{EL}} = (\underline{\Sigma}, \mathcal{N}) \in \mathfrak{T}_{\underline{H}_{\mathbb{EL}}}$ where, for every cluster $\mathsf{cl} \in CL_{\Sigma}$, we have:

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \mathbb{EL} & \text{if } \mathsf{cl} = \Sigma^{\circ} \\ \bot & \text{otherwise .} \end{cases} \tag{5}$$

Choice: $\mathfrak{A} \Box \mathfrak{X}$ is defined if $H \Box J$ is generated by the syntax (1,2), and then $\mathfrak{A} \Box \mathfrak{X} \stackrel{\mathrm{df}}{=} (\Sigma \Box \Psi, \mathcal{R}) \in \mathfrak{T}_{H \Box J}$ where, for every cluster $\mathsf{cl} \in CL_{\Sigma \Box \Psi}$, we have:

- when H is a dynamic at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \mathcal{M}({}^{\circ}\Sigma) & \text{if } \ \mathsf{cl} = {}^{\circ}(\Sigma \Box \Psi) \\ \mathcal{M}(\Sigma^{\circ}) & \text{if } \ \mathsf{cl} = (\Sigma \Box \Psi)^{\circ} \\ \mathcal{M}(\mathsf{cl}') & \text{if } \ \mathsf{cl} = e_{\Box} \lhd (v^1_{\Box} \blacktriangleleft \mathsf{cl}', v^2_{\Box} \blacktriangleleft {}^{\circ}\Psi) \\ \mathcal{M}({}^{\circ}\Sigma) & \text{if } \ \mathsf{cl} = e_{\Box} \lhd (v^1_{\Box} \blacktriangleleft {}^{\circ}\Sigma, v^2_{\Box} \blacktriangleleft \mathsf{cl}') \\ \mathcal{M}(\mathsf{cl}') & \text{if } \ \mathsf{cl} = v^1_{\Box} \blacktriangleleft \mathsf{cl}' \\ \bot & \text{if } \ \mathsf{cl} = v^2_{\Box} \blacktriangleleft \mathsf{cl}' \end{cases} \tag{6}$$

- when J is a dynamic at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \mathcal{N}({}^{\circ}\Psi) & \text{if } \ \mathsf{cl} = {}^{\circ}(\Sigma \Box \Psi) \\ \mathcal{N}(\Psi^{\circ}) & \text{if } \ \mathsf{cl} = (\Sigma \Box \Psi)^{\circ} \\ \mathcal{N}({}^{\circ}\Psi) & \text{if } \ \mathsf{cl} = e_{\Box} \lhd (v^1_{\Box} \blacktriangleleft \mathsf{cl}', v^2_{\Box} \blacktriangleleft {}^{\circ}\Psi) \\ \mathcal{N}(\mathsf{cl}') & \text{if } \ \mathsf{cl} = e_{\Box} \lhd (v^1_{\Box} \blacktriangleleft {}^{\circ}\Sigma, v^2_{\Box} \blacktriangleleft \mathsf{cl}') \\ \bot & \text{if } \ \mathsf{cl} = v^1_{\Box} \blacktriangleleft \mathsf{cl}' \\ \mathcal{N}(\mathsf{cl}') & \text{if } \ \mathsf{cl} = v^2_{\Box} \blacktriangleleft \mathsf{cl}' \end{cases} \tag{7}$$

- when both H and J are static at-expressions,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \bot \ . \tag{8}$$

Sequence: $\mathfrak{A}\,;\mathfrak{X}$ is defined if $H\,;J$ is generated by the syntax (1,2), and then $\mathfrak{A}\,;\mathfrak{X} \stackrel{\text{df}}{=} (\Sigma\,;\Psi,\mathcal{R}) \in \mathfrak{T}_{H\,;\,J}$ where, for every cluster $\mathsf{cl} \in CL_{\Sigma\,;\,\Psi}$, we have:

– when H is a dynamic at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\text{df}}{=} \begin{cases} \mathcal{M}(^\circ\Sigma) & \text{if } \mathsf{cl} = {}^\circ(\Sigma\,;\Psi) \\ \bot & \text{if } \mathsf{cl} = (\Sigma\,;\Psi)^\circ \\ \mathcal{M}(\mathsf{cl}') & \text{if } \mathsf{cl} = e_;\lhd(v_;^1 \blacktriangleleft \mathsf{cl}') \\ \mathcal{M}(\mathsf{cl}') & \text{if } \mathsf{cl} = v_;^1 \blacktriangleleft \mathsf{cl}' \\ \bot & \text{if } \mathsf{cl} = v_;^2 \blacktriangleleft \mathsf{cl}' \\ \mathcal{M}(\Sigma^\circ) & \text{if } \mathsf{cl} = i_;\lhd(v_;^1 \blacktriangleleft \Sigma^\circ, v_;^2 \blacktriangleleft \mathsf{cl}') \end{cases} \tag{9}$$

– when J is a dynamic at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\text{df}}{=} \begin{cases} \bot & \text{if } \mathsf{cl} = {}^\circ(\Sigma\,;\Psi) \\ \mathcal{N}(\Psi^\circ) & \text{if } \mathsf{cl} = (\Sigma\,;\Psi)^\circ \\ \bot & \text{if } \mathsf{cl} = e_;\lhd(v_;^1 \blacktriangleleft \mathsf{cl}') \\ \bot & \text{if } \mathsf{cl} = v_;^1 \blacktriangleleft \mathsf{cl}' \\ \mathcal{N}(\mathsf{cl}') & \text{if } \mathsf{cl} = v_;^2 \blacktriangleleft \mathsf{cl}' \\ \mathcal{N}(\mathsf{cl}') & \text{if } \mathsf{cl} = i_;\lhd(v_;^1 \blacktriangleleft \Sigma^\circ, v_;^2 \blacktriangleleft \mathsf{cl}') \end{cases} \tag{10}$$

– when both H and J are static at-expressions,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\text{df}}{=} \bot\,. \tag{11}$$

Parallel Composition: $\mathfrak{A}\|\mathfrak{X}$ is defined if $H\|J$ is generated by the syntax (1,2), and then $\mathfrak{A}\|\mathfrak{X} \stackrel{\text{df}}{=} (\Sigma\|\Psi,\mathcal{R}) \in \mathfrak{T}_{H\|J}$ where, for every cluster $\mathsf{cl} \in CL_{\Sigma\|\Psi}$, we have:

$$\mathcal{R}(\mathsf{cl}) \stackrel{\text{df}}{=} \begin{cases} \mathcal{M}(^\circ\Sigma) \oplus \mathcal{N}(^\circ\Psi) & \text{if } \mathsf{cl} = {}^\circ(\Sigma\|\Psi) \\ \mathcal{M}(\Sigma^\circ) \oplus \mathcal{N}(\Psi^\circ) & \text{if } \mathsf{cl} = (\Sigma\|\Psi)^\circ \\ \mathcal{M}(\mathsf{cl}') & \text{if } \mathsf{cl} = e_\|^1\lhd v_\|^1 \blacktriangleleft \mathsf{cl}' \\ \mathcal{N}(\mathsf{cl}') & \text{if } \mathsf{cl} = e_\|^2\lhd v_\|^2 \blacktriangleleft \mathsf{cl}' \\ \mathcal{M}(\mathsf{cl}') & \text{if } \mathsf{cl} = v_\|^1 \blacktriangleleft \mathsf{cl}' \\ \mathcal{N}(\mathsf{cl}') & \text{if } \mathsf{cl} = v_\|^2 \blacktriangleleft \mathsf{cl}'\,. \end{cases} \tag{12}$$

Note that when both H and J are static at-expressions, then

$$\mathcal{R}(\mathsf{cl}) \stackrel{\text{df}}{=} \bot\,. \tag{13}$$

for every cluster $\mathsf{cl} \in CL_{\Sigma\|\Psi}$.

Iteration: $\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle$ is defined if $\langle\!\langle H \circledast J \circledast K \rangle\!\rangle$ is generated by the syntax (1,2), and then

$$\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle \stackrel{\text{df}}{=} (\langle\!\langle \Sigma \circledast \Psi \circledast \Phi \rangle\!\rangle, \mathcal{R}) \in \mathfrak{T}_{\langle\!\langle H \circledast J \circledast K \rangle\!\rangle}$$

where, for every cluster $\mathsf{cl} \in CL_{\langle\!\langle \Sigma \circledast \Psi \circledast \Phi \rangle\!\rangle}$, we have:

– when H is a dynamic at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \mathcal{M}({}^\circ\Sigma) & \text{if } \mathsf{cl} = {}^\circ\langle\!\langle \Sigma \circledast \Psi \circledast \Phi \rangle\!\rangle \\ \bot & \text{if } \mathsf{cl} = \langle\!\langle \Sigma \circledast \Psi \circledast \Phi \rangle\!\rangle^\circ \\ \mathcal{M}({}^\circ\Sigma) & \text{if } \mathsf{cl} = e_\circledast \lhd (v^1_\circledast \blacktriangleleft \mathsf{cl}') \\ \mathcal{M}(\mathsf{cl}') & \text{if } \mathsf{cl} = v^1_\circledast \blacktriangleleft \mathsf{cl}' \\ \bot & \text{if } \mathsf{cl} = v^2_\circledast \blacktriangleleft \mathsf{cl}' \\ \bot & \text{if } \mathsf{cl} = v^3_\circledast \blacktriangleleft \mathsf{cl}' \\ \mathcal{M}(\Sigma^\circ) & \text{if } \mathsf{cl} = i_\circledast \lhd (v^1_\circledast \blacktriangleleft \Sigma^\circ, v^2_\circledast \blacktriangleleft \mathsf{cl}', \\ & \qquad\qquad v^2_\circledast \blacktriangleleft \Psi^\circ, v^3_\circledast \blacktriangleleft {}^\circ\Phi) \\ \mathcal{M}(\Sigma^\circ) & \text{if } \mathsf{cl} = i_\circledast \lhd (v^1_\circledast \blacktriangleleft \Sigma^\circ, v^2_\circledast \blacktriangleleft {}^\circ\Psi, \\ & \qquad\qquad v^2_\circledast \blacktriangleleft \Psi^\circ, v^3_\circledast \blacktriangleleft \mathsf{cl}') \end{cases} \tag{14}$$

– when J is a dynamic at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \bot & \text{if } \mathsf{cl} = {}^\circ\langle\!\langle \Sigma \circledast \Psi \circledast \Phi \rangle\!\rangle \\ \bot & \text{if } \mathsf{cl} = \langle\!\langle \Sigma \circledast \Psi \circledast \Phi \rangle\!\rangle)^\circ \\ \bot & \text{if } \mathsf{cl} = e_\circledast \lhd (v^1_\circledast \blacktriangleleft \mathsf{cl}') \\ \bot & \text{if } \mathsf{cl} = v^1_\circledast \blacktriangleleft \mathsf{cl}' \\ \mathcal{N}(\mathsf{cl}') & \text{if } \mathsf{cl} = v^2_\circledast \blacktriangleleft \mathsf{cl}' \\ \bot & \text{if } \mathsf{cl} = v^3_\circledast \blacktriangleleft \mathsf{cl}' \\ \mathcal{N}(\mathsf{cl}') \oplus \mathcal{N}(\Psi^\circ) & \text{if } \mathsf{cl} = i_\circledast \lhd (v^1_\circledast \blacktriangleleft \Sigma^\circ, v^2_\circledast \blacktriangleleft \mathsf{cl}', \\ & \qquad\qquad v^2_\circledast \blacktriangleleft \Psi^\circ, v^3_\circledast \blacktriangleleft {}^\circ\Phi) \\ \mathcal{N}({}^\circ\Psi) \oplus \mathcal{N}(\Psi^\circ) & \text{if } \mathsf{cl} = i_\circledast \lhd (v^1_\circledast \blacktriangleleft \Sigma^\circ, v^2_\circledast \blacktriangleleft {}^\circ\Psi, \\ & \qquad\qquad v^2_\circledast \blacktriangleleft \Psi^\circ, v^3_\circledast \blacktriangleleft \mathsf{cl}') \end{cases} \tag{15}$$

– when K is a dynamic at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \bot & \text{if } \mathsf{cl} = {}^\circ\langle\!\langle \Sigma \circledast \Psi \circledast \Phi \rangle\!\rangle \\ \mathcal{P}(\Phi^\circ) & \text{if } \mathsf{cl} = \langle\!\langle \Sigma \circledast \Psi \circledast \Phi \rangle\!\rangle^\circ \\ \bot & \text{if } \mathsf{cl} = e_\circledast \lhd (v^1_\circledast \blacktriangleleft \mathsf{cl}') \\ \bot & \text{if } \mathsf{cl} = v^1_\circledast \blacktriangleleft \mathsf{cl}' \\ \bot & \text{if } \mathsf{cl} = v^2_\circledast \blacktriangleleft \mathsf{cl}' \\ \mathcal{P}(\mathsf{cl}') & \text{if } \mathsf{cl} = v^3_\circledast \blacktriangleleft \mathsf{cl}' \\ \mathcal{P}({}^\circ\Phi) & \text{if } \mathsf{cl} = i_\circledast \lhd (v^1_\circledast \blacktriangleleft \Sigma^\circ, v^2_\circledast \blacktriangleleft \mathsf{cl}', \\ & \qquad\qquad v^2_\circledast \blacktriangleleft \Psi^\circ, v^3_\circledast \blacktriangleleft {}^\circ\Phi) \\ \mathcal{P}(\mathsf{cl}') & \text{if } \mathsf{cl} = i_\circledast \lhd (v^1_\circledast \blacktriangleleft \Sigma^\circ, v^2_\circledast \blacktriangleleft {}^\circ\Psi, \\ & \qquad\qquad v^2_\circledast \blacktriangleleft \Psi^\circ, v^3_\circledast \blacktriangleleft \mathsf{cl}') \end{cases} \tag{16}$$

– when H, J and K are static at-expressions,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \bot\ . \tag{17}$$

Scoping: $\mathfrak{A}\,\mathsf{sc}\,A$ is defined if $H\,\mathsf{sc}\,A$ is generated by the syntax (1,2), and then $\mathfrak{A}\,\mathsf{sc}\,A \stackrel{\mathrm{df}}{=} (\Sigma\,\mathsf{sc}\,A, \mathcal{R}) \in \mathfrak{T}_{H\,\mathsf{sc}\,A}$ where, for every cluster $\mathsf{cl} \in CL_{\Sigma\,\mathsf{sc}\,A}$, we have:

– when H is a dynamic at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \mathcal{M}(^\circ\Sigma) & \text{if } \mathsf{cl} = {}^\circ(\Sigma\,\mathsf{sc}\,A) \\ \mathcal{M}(\Sigma^\circ) & \text{if } \mathsf{cl} = (\Sigma\,\mathsf{sc}\,A)^\circ \\ \mathcal{M}(\mathsf{cl}') & \text{if } \mathsf{cl} = e_{\mathsf{sc}\,A} \lhd (v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl}') \\ \mathcal{M}(\mathsf{cl}') & \text{if } \mathsf{cl} = v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl}' \end{cases} \tag{18}$$

– when H is a static at-expression,

$$\mathcal{R}(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \bot\,. \tag{19}$$

Note that for each of the above operations, one can easily check that the result is indeed a valid cat-box corresponding to the at-expression given in the definition.

Static properties of cat-boxes An important result from the point of view of developing a correspondence between cat-boxes and at-expressions is given next (see also table 1).

Proposition 18. *Let $\mathfrak{A}$, $\mathfrak{X}$ and $\mathfrak{V}$ be static cat-boxes and $\mathbb{EL}, \mathbb{E}'\mathbb{L}' \in \mathbb{D}$. Then the following hold.*

1. *For choice composition:*

$$\overline{\mathfrak{A}\Box\mathfrak{X}}^{\mathbb{EL}} = \overline{\mathfrak{A}}^{\mathbb{EL}}\Box\mathfrak{X} = \mathfrak{A}\Box\overline{\mathfrak{X}}^{\mathbb{EL}}$$
$$\underline{\mathfrak{A}\Box\mathfrak{X}}_{\mathbb{EL}} = \underline{\mathfrak{A}}_{\mathbb{EL}}\Box\mathfrak{X} = \mathfrak{A}\Box\underline{\mathfrak{X}}_{\mathbb{EL}}\,.$$

2. *For iteration:*

$$\overline{\langle\!\langle\mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V}\rangle\!\rangle}^{\mathbb{EL}} = \langle\!\langle\overline{\mathfrak{A}}^{\mathbb{EL}} \circledast \mathfrak{X} \circledast \mathfrak{V}\rangle\!\rangle$$
$$\underline{\langle\!\langle\mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V}\rangle\!\rangle}_{\mathbb{EL}} = \langle\!\langle\mathfrak{A} \circledast \mathfrak{X} \circledast \underline{\mathfrak{V}}_{\mathbb{EL}}\rangle\!\rangle$$
$$\langle\!\langle\underline{\mathfrak{A}}_{\mathbb{EL}} \circledast \mathfrak{X} \circledast \mathfrak{V}\rangle\!\rangle = \langle\!\langle\mathfrak{A} \circledast \overline{\mathfrak{X}}^{\mathbb{EL}} \circledast \mathfrak{V}\rangle\!\rangle = \langle\!\langle\mathfrak{A} \circledast \underline{\mathfrak{X}}_{\mathbb{EL}} \circledast \mathfrak{V}\rangle\!\rangle = \langle\!\langle\mathfrak{A} \circledast \mathfrak{X} \circledast \overline{\mathfrak{V}}^{\mathbb{EL}}\rangle\!\rangle\,.$$

3. *For sequence composition:*

$$\overline{\mathfrak{A}\,;\mathfrak{X}}^{\mathbb{EL}} = \overline{\mathfrak{A}}^{\mathbb{EL}}\,;\mathfrak{X}$$
$$\underline{\mathfrak{A}}_{\mathbb{EL}}\,;\mathfrak{X} = \mathfrak{A}\,;\overline{\mathfrak{X}}^{\mathbb{EL}}$$
$$\mathfrak{A}\,;\underline{\mathfrak{X}}_{\mathbb{EL}} = \underline{\mathfrak{A}\,;\mathfrak{X}}_{\mathbb{EL}}\,.$$

4. *For parallel composition:*

$$\overline{\mathfrak{A}\|\mathfrak{X}}^{\mathbb{EL}} = \overline{\mathfrak{A}}^{\mathbb{EL}}\|\overline{\mathfrak{X}}^{\mathbb{EL}}$$
$$\underline{\mathfrak{A}}_{\mathbb{EL}}\|\underline{\mathfrak{X}}_{\mathbb{E}'\mathbb{L}'} = \underline{\mathfrak{A}\|\mathfrak{X}}_{\min\{\mathbb{E},\mathbb{E}'\}\max\{\mathbb{L},\mathbb{L}'\}}\,.$$

5. *For scoping:*

$$\overline{\mathfrak{A}\,\mathsf{sc}\,A}^{\mathbb{EL}} = \overline{\mathfrak{A}}^{\mathbb{EL}}\,\mathsf{sc}\,A$$
$$\underline{\mathfrak{A}\,\mathsf{sc}\,A}_{\mathbb{EL}} = \underline{\mathfrak{A}}_{\mathbb{EL}}\,\mathsf{sc}\,A\,.$$

Proof. See appendix G. □

Structural equivalence We now want to capture situations where different applications of a same operator box lead to the same cat-box. We start by defining three auxiliary relations which are the smallest equivalence relations on pairs of cat-boxes satisfying the following (below $\mathfrak{A}$, $\mathfrak{X}$ and $\mathfrak{V}$ are static cat-boxes and $\mathbb{EL} \in \mathbb{D}$):

- $(\overline{\mathfrak{A}}^{\mathbb{EL}}, \mathfrak{X}) \equiv_{\Box} (\mathfrak{A}, \overline{\mathfrak{X}}^{\mathbb{EL}})$ and $(\underline{\mathfrak{A}}_{\mathbb{EL}}, \mathfrak{X}) \equiv_{\Box} (\mathfrak{A}, \underline{\mathfrak{X}}_{\mathbb{EL}})$.
- $(\underline{\mathfrak{A}}_{\mathbb{EL}}, \mathfrak{X}) \equiv_{;} (\mathfrak{A}, \overline{\mathfrak{X}}^{\mathbb{EL}})$.
- $(\underline{\mathfrak{A}}_{\mathbb{EL}}, \mathfrak{X}, \mathfrak{V}) \equiv_{\circledast} (\mathfrak{A}, \overline{\mathfrak{X}}^{\mathbb{EL}}, \mathfrak{V}) \equiv_{\circledast} (\mathfrak{A}, \underline{\mathfrak{X}}_{\mathbb{EL}}, \mathfrak{V}) \equiv_{\circledast} (\mathfrak{A}, \mathfrak{X}, \overline{\mathfrak{V}}^{\mathbb{EL}})$.

Moreover, $\equiv_{\|}$ is the identity on the pairs of cat-boxes.

Proposition 19. *Let $\mathfrak{A}, \mathfrak{A}', \mathfrak{X}$ and $\mathfrak{X}'$ be cat-boxes.*

1. $\mathfrak{A} \Box \mathfrak{X} = \mathfrak{A}' \Box \mathfrak{X}'$ iff $(\mathfrak{A}, \mathfrak{X}) \equiv_{\Box} (\mathfrak{A}', \mathfrak{X}')$.
2. $\mathfrak{A} \,;\, \mathfrak{X} = \mathfrak{A}' \,;\, \mathfrak{X}'$ iff $(\mathfrak{A}, \mathfrak{X}) \equiv_{;} (\mathfrak{A}', \mathfrak{X}')$.
3. $\mathfrak{A} \| \mathfrak{X} = \mathfrak{A}' \| \mathfrak{X}'$ iff $(\mathfrak{A}, \mathfrak{X}) \equiv_{\|} (\mathfrak{A}', \mathfrak{X}')$.
4. $\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle = \langle\!\langle \mathfrak{A}' \circledast \mathfrak{X}' \circledast \mathfrak{V}' \rangle\!\rangle$ iff $(\mathfrak{A}, \mathfrak{X}, \mathfrak{V}) \equiv_{\circledast} (\mathfrak{A}', \mathfrak{X}', \mathfrak{V}')$.

Proof. See appendix G. □

Structural execution of transition steps We now provide a characterisation of steps executed by cat-boxes which reflects the compositional way in which they have been defined, providing a direct link to the execution rules of the corresponding at-expressions.

Proposition 20. *Let $\Omega_{op} \in \{\Omega_{\Box}, \Omega_{\circledast}, \Omega_{;}, \Omega_{\|}\}$ be any n-unary ($n \geq 2$) operator box and $\widetilde{\mathfrak{A}} = (\mathfrak{A}_1, \ldots, \mathfrak{A}_n)$ be a tuple of static and dynamic cat-boxes in its domain of application.*

1. *If $\mathfrak{A}_i[U_i\rangle\mathfrak{X}_i$ (for $i \leq n$), then $\widetilde{\mathfrak{X}} = (\mathfrak{X}_1, \ldots, \mathfrak{X}_n)$ is in the domain of application of Ω_{op} and $\Omega_{op}(\widetilde{\mathfrak{A}})[U\rangle\Omega_{op}(\widetilde{\mathfrak{X}})$, where*
$$U = (v^1_{op} \blacktriangleleft U_1) \cup \ldots \cup (v^n_{op} \blacktriangleleft U_n)\ . \tag{20}$$
2. *If $\Omega_{op}(\widetilde{\mathfrak{A}})[U\rangle\mathfrak{K}$, then there are tuples $\widetilde{\mathfrak{X}}, \widetilde{\mathfrak{V}}$ of cat-boxes in the application domain of Ω_{op} as well as steps $U_1, \ldots, U_n$ (some of them possibly empty) such that (20) holds, $\widetilde{\mathfrak{A}} \equiv_{\Omega_{op}} \widetilde{\mathfrak{X}}$, $\mathfrak{X}_i[U_i\rangle\mathfrak{V}_i$ (for $i \leq n$) and $\mathfrak{K} = \Omega_{op}(\widetilde{\mathfrak{V}})$.*

Note: As a consequence, $\mathsf{enabled}(\Omega_{op}(\widetilde{\mathfrak{A}}))$ *comprises exactly all sets*
$$(v^1_{op} \blacktriangleleft U_1) \cup \ldots \cup (v^n_{op} \blacktriangleleft U_n)$$
of transitions such that there is $\widetilde{\mathfrak{X}} = (\mathfrak{X}_1, \ldots, \mathfrak{X}_n)$ satisfying $\widetilde{\mathfrak{X}} \equiv_{\Omega_{op}} \widetilde{\mathfrak{A}}$ and $U_i \in \mathsf{enabled}(\mathfrak{X}_i)$ (for $i \leq n$).

Proof. Follows from similar results holding in the standard box algebra, proposition 19, and the fact that the age of tokens and the time annotations are consistently inherited through the composition operation specified by Ω_{op}. □

Proposition 21. *Let $\mathfrak{A}$ be a dynamic cat-box and $A \subseteq \mathcal{A}$.*

1. *If $\mathfrak{A}[\{t_1, u_1, \ldots, t_k, u_k, w_1, \ldots, w_m\}\rangle\mathfrak{X}$ where $\lambda_{\lfloor\mathfrak{A}\rfloor}(t_i) = \widehat{\lambda_{\lfloor\mathfrak{A}\rfloor}(u_i)} \in A$ for all $i \leq k$ and $\lambda_{\lfloor\mathfrak{A}\rfloor}(w_j) \notin A$ for all $j \leq m$, then $\Omega_{\mathsf{sc}\,A}(\mathfrak{A})[U\rangle\Omega_{\mathsf{sc}\,A}(\mathfrak{X})$, where*

$$U = \{v_{\mathsf{sc}\,A} \lhd \{t_1, u_1\}, \ldots, v_{\mathsf{sc}\,A} \lhd \{t_k, u_k\}, v_{\mathsf{sc}\,A} \lhd w_1, \ldots, v_{\mathsf{sc}\,A} \lhd w_m\}\ . \quad (21)$$

2. *If $\Omega_{\mathsf{sc}\,A}(\mathfrak{A})[U\rangle\mathfrak{K}$ then there are transitions $t_1, u_1, \ldots, t_k, u_k, w_1, \ldots, w_m$ and a cat-box $\mathfrak{X}$ as in part (1) which satisfy $\mathfrak{K} = \Omega_{\mathsf{sc}\,A}(\mathfrak{X})$ and (21).*

Note: As a consequence, $\mathsf{enabled}(\Omega_{\mathsf{sc}\,A}(\mathfrak{A}))$ *comprises exactly all*

$$U = \{v_{\mathsf{sc}\,A} \lhd \{t_1, u_1\}, \ldots, v_{\mathsf{sc}\,A} \lhd \{t_k, u_k\}, v_{\mathsf{sc}\,A} \lhd w_1, \ldots, v_{\mathsf{sc}\,A} \lhd w_m\}$$

such that $\lambda_{\lfloor\mathfrak{A}\rfloor}(t_i) = \widehat{\lambda_{\lfloor\mathfrak{A}\rfloor}(u_i)} \in A$ for all $i \leq k$ and $\lambda_{\lfloor\mathfrak{A}\rfloor}(w_j) \notin A$ for all $j \leq m$.

Proof. Follows from a similar result holding in the standard box algebra, and the fact that the age of tokens and the time annotations are consistently inherited through the composition operation specified by $\Omega_{\mathsf{sc}\,A}$. ☐

Structural characterisation of urgent transitions We now provide a compositional characterisation of urgent transitions of cat-boxes.

Proposition 22. *Let $\Omega_{op} \in \{\Omega_{\Box}, \Omega_{\circledast}, \Omega_{;}, \Omega_{\|}\}$ be any n-unary ($n \geq 2$) operator box and $\widetilde{\mathfrak{A}} = (\mathfrak{A}_1, \ldots, \mathfrak{A}_n)$ be a tuple of static and dynamic cat-boxes in its domain of application.*

1. *If $t \in \mathsf{urgent}(\mathfrak{A}_i)$, for some $i \leq n$, then $v^i_{op} \lhd t \in \mathsf{urgent}(\Omega_{op}(\widetilde{\mathfrak{A}}))$.*
2. *If $v^i_{op} \lhd t \in \mathsf{urgent}(\Omega_{op}(\widetilde{\mathfrak{A}}))$, for some $i \leq n$, then there is a tuple $\widetilde{\mathfrak{X}} = (\mathfrak{X}_1, \ldots, \mathfrak{X}_n)$ of cat-boxes in the application domain of Ω_{op} such that $\widetilde{\mathfrak{A}} \equiv_{\Omega_{op}} \widetilde{\mathfrak{X}}$ and $t \in \mathsf{urgent}(\mathfrak{X}_i)$.*

Proof. Follows from the note in the formulation of proposition 20, and the fact that the age of tokens and the time annotations are consistently inherited through the composition operation specified by Ω_{op}. ☐

Proposition 23. *Let $\mathfrak{A}$ be a dynamic cat-box, $A \subseteq \mathcal{A}$ and $v_{\mathsf{sc}\,A} \lhd U \in T_{\Omega_{\mathsf{sc}\,A}(\mathfrak{A})}$. Then*

$$v_{\mathsf{sc}\,A} \lhd U \in \mathsf{urgent}(\Omega_{\mathsf{sc}\,A}(\llbracket\mathfrak{A}\rrbracket)) \iff U \cap \mathsf{urgent}(\mathfrak{A}) \neq \varnothing\ .$$

Proof. Follows from the note in the formulation of proposition 21, and the fact that the age of tokens and the time annotations are consistently inherited through the composition operation specified by $\Omega_{\mathsf{sc}\,A}$. ☐

From at-expressions to cat-boxes We now provide a compositional translation from at-expressions to cat-boxes. Note that it is applicable to static and dynamic at-expressions, unlike the previous translation from static at-expressions to at-boxes.

The mapping cBox from at-expressions to cat-boxes is defined so that:

$$\begin{aligned}
\mathsf{cBox}(\alpha el) &\stackrel{\text{df}}{=} \mathsf{N}^{\mathsf{cat}}_{\alpha el} \\
\mathsf{cBox}(\overline{H}^{\mathbb{EL}}) &\stackrel{\text{df}}{=} \overline{\mathsf{cBox}(H)}^{\mathbb{EL}} \\
\mathsf{cBox}(\underline{H}_{\mathbb{EL}}) &\stackrel{\text{df}}{=} \underline{\mathsf{cBox}(H)}_{\mathbb{EL}} \\
\mathsf{cBox}(H \,\mathsf{sc}\, A) &\stackrel{\text{df}}{=} \mathsf{cBox}(H) \,\mathsf{sc}\, A \\
\mathsf{cBox}(H \Box J) &\stackrel{\text{df}}{=} \mathsf{cBox}(H) \Box \mathsf{cBox}(J) \\
\mathsf{cBox}(H \| J) &\stackrel{\text{df}}{=} \mathsf{cBox}(H) \| \mathsf{cBox}(J) \\
\mathsf{cBox}(H \,;\, J) &\stackrel{\text{df}}{=} \mathsf{cBox}(H) \,;\, \mathsf{cBox}(J) \\
\mathsf{cBox}(\langle\!\langle H \circledast J \circledast I \rangle\!\rangle) &\stackrel{\text{df}}{=} \langle\!\langle \mathsf{cBox}(H) \circledast \mathsf{cBox}(J) \circledast \mathsf{cBox}(I) \rangle\!\rangle \,,
\end{aligned}$$

where $\mathsf{N}^{\mathsf{cat}}_{\alpha el}$ is $\mathsf{N}_{\alpha el}$ with the token filling mapping returning only $\perp$. The semantical mapping always returns a cat-box, and the property of corresponding to a static or dynamic box has been captured by the syntax (1,2).

Proposition 24. *Let H be an at-expression.*

1. $\mathsf{cBox}(H)$ *is a static or dynamic cat-box.*
2. $\mathsf{cBox}(H)$ *is a static cat-box* iff H *is a static at-expression.*

Proof. Follows by induction on the structure of the at-expressions, using similar results holding in the standard box algebra. ☐

Relationship between at-expressions and cat-boxes The consistency between the denotational and the operational semantics of at-expressions will be formulated in terms of the full transition systems they generate.

We now have a fundamental result which demonstrates that the operational and denotational semantics of an at-expression capture the same behaviour.

Theorem 2. *For every at-expression H,*

$$\mathsf{iso}_H \stackrel{\text{df}}{=} \left\{ ([J]_\equiv, \mathsf{cBox}(J)) \;\middle|\; [J]_\equiv \textit{ is a node of } \mathsf{fTS}_H \right\}$$

is an isomorphism between the transition systems fTS_H and $\mathsf{fTS}_{\mathsf{cBox}(H)}$.

Proof. We proceed by induction on the structure of H. The result clearly holds when $\lfloor H \rfloor = \alpha el$. In the inductive step we do not need to consider H which is completely overbarred or underbarred (since then a rewriting, based on the rules in table 1, can be applied to push the bar inside the expression). After that we consider various cases for executing (transition or time) steps from H as well as $\mathsf{cBox}(H)$, and derive the appropriate steps in the counterpart node using the operational semantics rules, propositions 20, 21, 22, 23 and 24 as well as $\mathsf{cBox}(H^{\surd}) = \mathsf{cBox}(H)^{\surd}$. □

From the above result, a number of immediate corollaries can be derived, as stated next.

Theorem 3. *For every at-expression H and the corresponding cat-box $\mathsf{Box}(H)$, we have that:*

1. TS_H *and* $\mathsf{TS}_{\mathsf{Box}(H)}$ *are isomorphic.*
2. fRT_H *and* $\mathsf{fRT}_{\mathsf{Box}(H)}$ *are isomorphic.*
3. RT_H *and* $\mathsf{RT}_{\mathsf{Box}(H)}$ *are isomorphic.*

Proof. Follows from theorem 2 and the fact that, for both expressions and boxes, moving from a transition-based graph representing global behaviour to a label-based graph amounts to replacing in the original arcs all the U's by their multisets of communication labels (duplicate arcs are then deleted). □

E Relationship between at-boxes and cat-boxes

We are now going to relate the global behaviour of at-boxes and cat-boxes. This time, however, the main correspondence result will be expressed in terms of reachability trees rather than transition systems.

Let $\mathbf{\Theta} = (\Sigma, \mu)$ be an input-reachable at-box. Then $\yen(\mathbf{\Theta}) \stackrel{\mathrm{df}}{=} (\Sigma, \mathcal{M})$ where

$$\yen(\mu) : CL_\Sigma \to \mathbb{D}^\perp$$

is a cluster filling mapping such that, for every $\mathsf{cl} \in CL_\Sigma$:

$$\yen(\mu)(\mathsf{cl}) \stackrel{\mathrm{df}}{=} \begin{cases} \perp & \text{if } M_\Sigma(\mathsf{cl}) = \{\perp\} \\ \mathbb{EL} & \text{otherwise}, \end{cases}$$

with $\mathbb{E} = \min(\mu(\mathsf{cl} \cap M_\Sigma))$ and $\mathbb{L} = \max(\mu(\mathsf{cl} \cap M_\Sigma))$. It is easy to see that $\yen(\mathbf{\Theta})$ is a cat-box since the two conditions from the definition of a cat-box are satisfied due to the two corresponding conditions in the definition of an at-box.

Proposition 25. *Let $\mathbf{\Theta} = (\Sigma, \mu)$ be an input-reachable at-box, $\mathfrak{A} = \yen(\mathbf{\Theta}) = (\Sigma, \mathcal{M})$, and $t \in T_\Sigma$.*

1. $t \in \mathsf{enabled}(\mathbf{\Theta})$ iff $t \in \mathsf{enabled}(\mathfrak{A})$.
2. $t \in \mathsf{urgent}(\mathbf{\Theta})$ iff $t \in \mathsf{urgent}(\mathfrak{A})$.
3. $\surd$ *is enabled in* $\mathbf{\Theta}$ iff $\surd$ *is enabled in* $\mathfrak{A}$.
4. *If* $\mathbf{\Theta}[\{t\}\rangle\mathbf{\Xi}$ *then* $\mathfrak{A}[\{t\}\rangle\yen(\mathbf{\Xi})$.
5. *If* $\mathbf{\Theta}[\surd\rangle\mathbf{\Xi}$ *then* $\mathfrak{A}[\surd\rangle\yen(\mathbf{\Xi})$.

Proof. (1,2) ($\Longrightarrow$) Suppose that $t \in \mathsf{enabled}(\boldsymbol{\Theta})$ and $\mathsf{cl} \in {}^{\diamond}t$. Then, by proposition 10, $\mathsf{cl} \subseteq {}^{\bullet}t$ and $\lambda_{\mathfrak{A}}(\mathsf{cl}, t) = \lambda_{\Sigma}(p, t)$, for all $p \in \mathsf{cl}$. Thus, since $\mu(p)$ $\mathtt{tsat}$ $\lambda_{\Sigma}(p, t)$, for all $p \in \mathsf{cl}$, we have $\mathcal{M}(\mathsf{cl})$ $\mathtt{tsat}$ $\lambda_{\mathfrak{A}}(\mathsf{cl}, t)$. Moreover, if $t \in \mathsf{urgent}(\boldsymbol{\Theta})$ then $t \in \mathsf{urgent}(\mathfrak{A})$ since, for any any set of integers $K = \{k_1, \ldots, k_l\}$, we have

$$\begin{aligned} \min\{1 + k_1, \ldots, 1 + k_l\} &= 1 + \min K \\ \max\{1 + k_1, \ldots, 1 + k_l\} &= 1 + \max K \ . \end{aligned} \tag{22}$$

($\Longleftarrow$) Suppose that $t \in \mathsf{enabled}(\mathfrak{A})$ and $p \in {}^{\bullet}t$. Then, by proposition 11, there is $\mathsf{cl} \subseteq {}^{\bullet}t$ such that $p \in \mathsf{cl}$. After that we proceed by essentially reversing the argument for the ($\Longrightarrow$) implication.

(3) Follows from part (2).

(4) Let $\Sigma[\{t\}\rangle\Psi$ and $\boldsymbol{\Xi} = (\Psi, \nu)$. By part (1), there is a cat-box $\mathfrak{X} = (\Psi, \mathcal{N})$ such that $\mathfrak{A}[\{t\}\rangle\mathfrak{X}$. All we need to show is that $\mathcal{N} = ¥(\nu)$. To this end we take $\mathsf{cl} \in CL_{\Sigma}$. If $({}^{\bullet}t \cup t^{\bullet}) \cap \mathsf{cl} = \varnothing$ then

$$\mathcal{N}(\mathsf{cl}) = \mathcal{M}(\mathsf{cl}) = ¥(\mu(\mathsf{cl})) = ¥(\nu)(\mathsf{cl})$$

clearly holds. So, we assume that $({}^{\bullet}t \cup t^{\bullet}) \cap \mathsf{cl} \neq \varnothing$ and then consider three cases.

Case 1: $\mathsf{cl} \subseteq {}^{\circ}\Sigma$. Due to the ex-directedness of Σ, we have that ${}^{\bullet}t \cap \mathsf{cl} \neq \varnothing$ and $t^{\bullet} \cap \mathsf{cl} = \varnothing$. Hence we have the following:

- If $M_{\Psi} = \varnothing$ then $\mathcal{N}(\mathsf{cl}) = \bot = ¥(\nu)(\mathsf{cl})$.
- If $M_{\Psi} \neq \varnothing$ then $\mathcal{N}(\mathsf{cl}) = \mathcal{M}(\mathsf{cl})$, by definition of a step in cat-boxes. On the other hand, the second condition in the definition of an at-box guarantees that $¥(\nu)(\mathsf{cl}) = ¥(\mu)(\mathsf{cl})$. Hence $\mathcal{N}(\mathsf{cl}) = ¥(\nu)(\mathsf{cl})$.

Case 2: $\mathsf{cl} \subseteq \Sigma^{\circ}$. Due to the ex-directedness of Σ, we have that $t^{\bullet} \cap \mathsf{cl} \neq \varnothing$ and ${}^{\bullet}t \cap \mathsf{cl} = \varnothing$. Hence we have the following:

- If $\mathcal{M}(\mathsf{cl}) = \mathbb{EL}$ then $\mathcal{N}(\mathsf{cl}) = 0\mathbb{L}$. On the other hand, $\nu(\mathsf{cl}) = \mu(\mathsf{cl}) \cup \{0\}$ and so $¥(\nu)(\mathsf{cl}) = 0\mathbb{L}$.
- If $\mathcal{M}(\mathsf{cl}) = \bot$ then $\mathcal{N}(\mathsf{cl}) = 00$. On the other hand, $\nu(\mathsf{cl}) = \{0\}$ and so $¥(\nu)(\mathsf{cl}) = 00$.

Case 2: $\mathsf{cl} \subseteq \ddot{\Sigma}$. By proceeding similarly as above, we may verify the property when $M_{\Sigma} \cap \mathsf{cl} = \varnothing$ or $M_{\Psi} \cap \mathsf{cl} = \varnothing$ or $t^{\bullet} \cap \mathsf{cl} \neq \varnothing$ (which, by proposition 7 means that ${}^{\bullet}t \cap \mathsf{cl} \neq \varnothing$). The only situation which needs consideration is when:

$$M_{\Sigma} \cap \mathsf{cl} \neq \varnothing \neq M_{\Psi} \cap \mathsf{cl} \ \text{ and } \ {}^{\bullet}t \cap \mathsf{cl} \neq \varnothing = t^{\bullet} \cap \mathsf{cl} \ .$$

We then have $\mathcal{N}(\mathsf{cl}) = \mathcal{M}(\mathsf{cl})$, and so it suffices to show that $\nu(\mathsf{cl} \cap M_{\Psi}) = \mu(\mathsf{cl} \cap M_{\Sigma})$.

From proposition 9 it follows that we had Case 2 situation when t was executed. Moreover, if we look at the tokens residing in the places of cl we observe that they age uniformly and, crucially, if two tokens were produced by firing of the same transition filling the cluster, and they are still present in $\boldsymbol{\Theta}$ then their age given by μ is exactly the same.

From proposition 9 it follows that we must have had Case 2 situation when executing t. Therefore, we have that $\nu(\mathsf{cl} \cap M_\Psi) \subseteq \mu(\mathsf{cl} \cap M_\Sigma)$. Suppose now that $p \in (\mathsf{cl} \cap M_\Sigma) \setminus (\mathsf{cl} \cap M_\Psi)$ and that u was the transition which for the last time filled p with a timed token. Furthermore, without loss of generality, assume that cl (or, more precisely, its predecessor) has been formed by an application of the sequence operator on nets, $\Phi\,;\Phi'$. We therefore had a number of transitions $t_1, \ldots, t_m$ which were predecessors of the transitions emptying the cluster cl since the last time it has been filled. We note that there was at least one place q in ${}^\circ\Phi' \setminus {}^\bullet\{t_1, \ldots, t_m\}$ because $M_\Psi \cap \mathsf{cl} \neq \varnothing$. Let $r \in \Phi^\circ$ be any output place of a transition which was a predecessor of u. In the interface region $\Phi\,;\Phi'$ there existed then a place resulting from a combination of r and q. Its successor is then present in $\mathsf{cl} \cap M_\Psi$ and it has been filled for the last time by transition u at the same time as p. It therefore follows that $\mu(p) \in \nu(\mathsf{cl} \cap M_\Psi)$, and so $\mu(\mathsf{cl} \cap M_\Sigma) \subseteq \nu(\mathsf{cl} \cap M_\Psi)$

(5) By part (3), $\surd$ is enabled in $¥(\boldsymbol{\Theta})$. Moreover, we have $¥(\boldsymbol{\Theta})[\surd\rangle¥(\boldsymbol{\Xi})$ by property (22). □

Theorem 4. *Let $\boldsymbol{\Theta}$ be an input-reachable at-box. Then the following hold.*

1. $\mathsf{fTS}_{\boldsymbol{\Theta}}$ *is strongly bisimilar (see [15]) to* $\mathsf{fTS}_{¥(\boldsymbol{\Theta})}$.
2. $\mathsf{TS}_{\boldsymbol{\Theta}}$ *is strongly bisimilar to* $\mathsf{TS}_{¥(\boldsymbol{\Theta})}$.
3. $\mathsf{fRT}_{\boldsymbol{\Theta}}$ *is isomorphic to* $\mathsf{fRT}_{¥(\boldsymbol{\Theta})}$.
4. $\mathsf{RT}_{\boldsymbol{\Theta}}$ *isisomorphic to* $\mathsf{RT}_{¥(\boldsymbol{\Theta})}$.

Proof. (1) Follows from propositions 14, 17 and 25, using the mapping ¥ to relate the nodes of the two transition systems.

(2) This is an immediate consequence of part (1).

(3) Follows from part (1) and the fact that both transition systems are deterministic (no annotation can label two different arrows outgoing from a node of the trees; this follows from the properties of transition systems of Petri nets, and the properties of the evolutions in the box algebra[4]).

(4) This is an immediate consequence of part (3). □

F Relationship between at-expressions and at-boxes

We now can finally show and extend the key result formulated in the main body of the paper.

Theorem 5. *Let $G = \overline{E}^{00}$ be an initial dynamic at-expression and $\boldsymbol{\Theta} = \overline{\mathsf{Box}(E)}^{00}$ be the corresponding at-box. Then the following hold.*

1. fTS_G *is strongly bisimilar to* $\mathsf{fTS}_{\boldsymbol{\Theta}}$.
2. TS_G *is strongly bisimilar to* $\mathsf{TS}_{\boldsymbol{\Theta}}$.
3. fRT_G *is isomorphic to* $\mathsf{fRT}_{\boldsymbol{\Theta}}$.

[4] In particular, that if $G \xrightarrow{\Gamma} H$ and $G \xrightarrow{\Gamma} J$ then $H \equiv J$, which is easily re-stated in the at-expressions framework as well.

4. RT_G *isisomorphic to* RT_{Θ}.

Proof. Follows from theorems 2 and 4. □

Note that theorem 1 is then nothing but part (4) of the above result.

G Selected proofs

Proof of proposition 10

The proof proceeds by induction on the structure of the expression from which Σ has been derived. Below we assume that $t \in T_\Sigma$, $\mathsf{cl} \in {}^\diamond t$ and $p, q \in \mathsf{cl}$.

Base net: $\Sigma = \mathsf{N}_{\alpha el}$. Then $t = v_{\alpha el}$ and the property clearly holds.

Parallel composition: $\Sigma = \Sigma_1 \| \Sigma_2$.
Case 1: $t = v_\|^1 \lhd u$ where $u \in T_{\Sigma_1}$. Then, by the definition of ${}^\diamond t$, we have two possibilities:

- $\mathsf{cl} = e_\| \lhd v_\|^1 \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_1)$.
  By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^\bullet t \Leftrightarrow \mathsf{cl}' \subseteq {}^\bullet u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^\bullet u$; and (iii) $\lambda_{\Sigma_1}(p', u) = \lambda_{\Sigma_1}(q', u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = e_\| \lhd v_\|^1 \lhd p'$ and $q = e_\| \lhd v_\|^1 \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_\Sigma(p, t) = \lambda_{\Sigma_1}(p', u)$ and $\lambda_\Sigma(q, t) = \lambda_{\Sigma_1}(q', u)$.
  Now, $\mathsf{cl} \subseteq {}^\bullet t$ follows from (i) and (ii). Moreover, $\lambda_\Sigma(p, u) = \lambda_\Sigma(q, u)$ follows from (iii) and (iv).
- $\mathsf{cl} = v_\|^1 \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^\diamond u \cap \mathsf{cl_i}(\Sigma_1)$.
  By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^\bullet t \Leftrightarrow \mathsf{cl}' \subseteq {}^\bullet u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^\bullet u$; and (iii) $\lambda_{\Sigma_1}(p', u) = \lambda_{\Sigma_1}(q', u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = v_\|^1 \lhd p'$ and $q = v_\|^1 \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_\Sigma(p, t) = \lambda_{\Sigma_1}(p', u)$ and $\lambda_\Sigma(q, t) = \lambda_{\Sigma_1}(q', u)$.
  Now, $\mathsf{cl} \subseteq {}^\bullet t$ follows from (i) and (ii). Moreover, $\lambda_\Sigma(p, u) = \lambda_\Sigma(q, u)$ follows from (iii) and (iv).

Case 2: $t = v_\|^2 \lhd u$ where $u \in T_{\Sigma_2}$. Then we proceed similarly as in Case 1.

Sequential composition: $\Sigma = \Sigma_1 \,;\, \Sigma_2$.
Case 1: $t = v_;^1 \lhd u$ where $u \in T_{\Sigma_1}$. Then, by the definition of ${}^\diamond t$, we have two possibilities:

- $\mathsf{cl} = e_; \lhd v_;^1 \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_1)$.
  By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^\bullet t \Leftrightarrow \mathsf{cl}' \subseteq {}^\bullet u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^\bullet u$; and (iii) $\lambda_{\Sigma_1}(p', u) = \lambda_{\Sigma_1}(q', u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = e_; \lhd v_;^1 \lhd p'$ and $q = e_; \lhd v_;^1 \lhd q'$ where

$p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_\Sigma(p,t) = \lambda_{\Sigma_1}(p',u)$ and $\lambda_\Sigma(q,t) = \lambda_{\Sigma_1}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^\bullet t$ follows from (i) and (ii). Moreover, $\lambda_\Sigma(p,u) = \lambda_\Sigma(q,u)$ follows from (iii) and (iv).

– $\mathsf{cl} = v^1_; \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^\diamond u \cap \mathsf{cl_i}(\Sigma_1)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^\bullet t \Leftrightarrow \mathsf{cl}' \subseteq {}^\bullet u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^\bullet u$; and (iii) $\lambda_{\Sigma_1}(p',u) = \lambda_{\Sigma_1}(q',u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = v^1_; \lhd p'$ and $q = v^1_; \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_\Sigma(p,t) = \lambda_{\Sigma_1}(p',u)$ and $\lambda_\Sigma(q,t) = \lambda_{\Sigma_1}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^\bullet t$ follows from (i) and (ii). Moreover, $\lambda_\Sigma(p,u) = \lambda_\Sigma(q,u)$ follows from (iii) and (iv).

Case 2: $t = v^2_; \lhd u$ where $u \in T_{\Sigma_2}$. Then, by the definition of ${}^\diamond t$, we have two possibilities:

– $\mathsf{cl} = i_; \lhd (v^1_; \blacktriangleleft \Sigma_1^\circ, v^2_; \blacktriangleleft \mathsf{cl}')$, where $\mathsf{cl}' \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_2)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^\bullet t \Leftrightarrow \mathsf{cl}' \subseteq {}^\bullet u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^\bullet u$; and (iii) $\lambda_{\Sigma_2}(p',u) = \lambda_{\Sigma_2}(q',u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = i_; \lhd (v^1_; \lhd w, v^2_; \lhd p')$ and $q = i_; \lhd (v^1_; \lhd w', v^2_; \lhd q')$ where $w, w' \in \Sigma_1^\circ$ and $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_\Sigma(p,t) = \lambda_{\Sigma_2}(p',u)$ and $\lambda_\Sigma(q,t) = \lambda_{\Sigma_2}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^\bullet t$ follows from (i) and (ii). Moreover, $\lambda_\Sigma(p,u) = \lambda_\Sigma(q,u)$ follows from (iii) and (iv).

– $\mathsf{cl} = v^2_; \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^\diamond u \cap \mathsf{cl_i}(\Sigma_2)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^\bullet t \Leftrightarrow \mathsf{cl}' \subseteq {}^\bullet u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^\bullet u$; and (iii) $\lambda_{\Sigma_2}(p',u) = \lambda_{\Sigma_2}(q',u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = v^2_; \lhd p'$ and $q = v^2_; \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_\Sigma(p,t) = \lambda_{\Sigma_2}(p',u)$ and $\lambda_\Sigma(q,t) = \lambda_{\Sigma_2}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^\bullet t$ follows from (i) and (ii). Moreover, $\lambda_\Sigma(p,u) = \lambda_\Sigma(q,u)$ follows from (iii) and (iv).

Choice: $\Sigma = \Sigma_1 \Box \Sigma_2$.
Case 1: $t = v^1_\Box \lhd u$ where $u \in T_{\Sigma_1}$. Then, by the definition of ${}^\diamond t$, we have two possibilities:

– $\mathsf{cl} = e_\Box \lhd (v^1_\Box \blacktriangleleft \mathsf{cl}', v^2_\Box \blacktriangleleft {}^\circ\Sigma_2)$, where $\mathsf{cl}' \in {}^\diamond u \cap \mathsf{cl_e}(\Sigma_1)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^\bullet t \Leftrightarrow \mathsf{cl}' \subseteq {}^\bullet u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^\bullet u$; and (iii) $\lambda_{\Sigma_1}(p',u) = \lambda_{\Sigma_1}(q',u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = e_\Box \lhd (v^1_\Box \lhd p', v^2_\Box \lhd w)$ and $q = e_\Box \lhd (v^1_\Box \lhd q', v^2_\Box \lhd w')$ where $w, w' \in {}^\circ\Sigma_2$ and $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_\Sigma(p,t) = \lambda_{\Sigma_1}(p',u)$ and $\lambda_\Sigma(q,t) = \lambda_{\Sigma_1}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^\bullet t$ follows from (i) and (ii). Moreover, $\lambda_\Sigma(p,u) = \lambda_\Sigma(q,u)$ follows from (iii) and (iv).

– $\mathsf{cl} = v^1_{\square} \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_i}(\Sigma_1)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_1}(p', u) = \lambda_{\Sigma_1}(q', u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = v^1_{\square} \lhd p'$ and $q = v^1_{\square} \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p, t) = \lambda_{\Sigma_1}(p', u)$ and $\lambda_{\Sigma}(q, t) = \lambda_{\Sigma_1}(q', u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p, u) = \lambda_{\Sigma}(q, u)$ follows from (iii) and (iv).

Case 2: $t = v^2_{\square} \lhd u$ where $u \in T_{\Sigma_2}$. Then we proceed similarly as in Case 1.

Iteration: $\Sigma = \langle\langle \Sigma_1 \circledast \Sigma_2 \circledast \Sigma_3 \rangle\rangle$.
Case 1: $t = v^1_{\circledast} \lhd u$ where $u \in T_{\Sigma_1}$. Then, by the definition of ${}^{\diamond}t$, we have two possibilities:

– $\mathsf{cl} = e_{\circledast} \lhd v^1_{\circledast} \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_e}(\Sigma_1)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_1}(p', u) = \lambda_{\Sigma_1}(q', u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = e_{\circledast} \lhd v^1_{\circledast} \lhd p'$ and $q = e_{\circledast} \lhd v^1_{\circledast} \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p, t) = \lambda_{\Sigma_1}(p', u)$ and $\lambda_{\Sigma}(q, t) = \lambda_{\Sigma_1}(q', u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p, u) = \lambda_{\Sigma}(q, u)$ follows from (iii) and (iv).
– $\mathsf{cl} = v^1_{\circledast} \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_i}(\Sigma_1)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_1}(p', u) = \lambda_{\Sigma_1}(q', u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = v^1_{\circledast} \lhd p'$ and $q = v^1_{\circledast} \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p, t) = \lambda_{\Sigma_1}(p', u)$ and $\lambda_{\Sigma}(q, t) = \lambda_{\Sigma_1}(q', u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p, u) = \lambda_{\Sigma}(q, u)$ follows from (iii) and (iv).

Case 2: $t = v^2_{\circledast} \lhd u$ where $u \in T_{\Sigma_2}$. Then, by the definition of ${}^{\diamond}t$, we have two possibilities:

– $\mathsf{cl} = i_{\circledast} \lhd (v^1_{\circledast} \blacktriangleleft \Sigma_1^{\circ}, v^2_{\circledast} \blacktriangleleft \mathsf{cl}', v^2_{\circledast} \blacktriangleleft \Sigma_2^{\circ}, v^3_{\circledast} \blacktriangleleft {}^{\circ}\Sigma_3)$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_e}(\Sigma_2)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_2}(p', u) = \lambda_{\Sigma_2}(q', u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = i_{\circledast} \lhd (v^1_{\circledast} \lhd w, v^2_{\circledast} \lhd p', v^2_{\circledast} \lhd y, v^3_{\circledast} \lhd z)$ and $q = i_{\circledast} \lhd (v^1_{\circledast} \lhd w', v^2_{\circledast} \lhd q', v^2_{\circledast} \lhd y', v^3_{\circledast} \lhd z')$ where $w, w' \in \Sigma_1^{\circ}$, $y, y' \in \Sigma_2^{\circ}$, $z, z' \in {}^{\circ}\Sigma_3$ and $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p, t) = \lambda_{\Sigma_2}(p', u)$ and $\lambda_{\Sigma}(q, t) = \lambda_{\Sigma_2}(q', u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p, u) = \lambda_{\Sigma}(q, u)$ follows from (iii) and (iv).
– $\mathsf{cl} = v^2_{\circledast} \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_i}(\Sigma_2)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_2}(p', u) = \lambda_{\Sigma_2}(q', u)$,

for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = v^2_{\circledast} \lhd p'$ and $q = v^2_{\circledast} \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p,t) = \lambda_{\Sigma_2}(p',u)$ and $\lambda_{\Sigma}(q,t) = \lambda_{\Sigma_2}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p,u) = \lambda_{\Sigma}(q,u)$ follows from (iii) and (iv).

Case 3: $t = v^3_{\circledast} \lhd u$ where $u \in T_{\Sigma_3}$. Then, by the definition of ${}^{\diamond}t$, we have two possibilities:

– $\mathsf{cl} = i_{\circledast} \lhd (v^1_{\circledast} \blacktriangleleft \Sigma_1^{\circ}, v^2_{\circledast} \blacktriangleleft {}^{\circ}\Sigma_2, v^2_{\circledast} \blacktriangleleft \Sigma_2^{\circ}, v^3_{\circledast} \blacktriangleleft \mathsf{cl}')$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_e}(\Sigma_3)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_3}(p',u) = \lambda_{\Sigma_3}(q',u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = i_{\circledast} \lhd (v^1_{\circledast} \lhd w, v^2_{\circledast} \lhd y, v^2_{\circledast} \lhd z, v^3_{\circledast} \lhd p')$ and $q = i_{\circledast} \lhd (v^1_{\circledast} \lhd w', v^2_{\circledast} \lhd y', v^2_{\circledast} \lhd z', v^3_{\circledast} \lhd q')$ where $w, w' \in \Sigma_1^{\circ}$, $y, y' \in {}^{\circ}\Sigma_2$, $z, z' \in \Sigma_2^{\circ}$ and $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p,t) = \lambda_{\Sigma_3}(p',u)$ and $\lambda_{\Sigma}(q,t) = \lambda_{\Sigma_3}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p,u) = \lambda_{\Sigma}(q,u)$ follows from (iii) and (iv).
– $\mathsf{cl} = v^3_{\circledast} \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_i}(\Sigma_3)$
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_3}(p',u) = \lambda_{\Sigma_3}(q',u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = v^3_{\circledast} \lhd p'$ and $q = v^3_{\circledast} \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p,t) = \lambda_{\Sigma_3}(p',u)$ and $\lambda_{\Sigma}(q,t) = \lambda_{\Sigma_3}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p,u) = \lambda_{\Sigma}(q,u)$ follows from (iii) and (iv).

Scoping: $\Sigma = \Sigma_1 \,\mathsf{sc}\, A$.
Case 1: $t = v_{\mathsf{sc}\,A} \lhd u$ where $u \in T_{\Sigma_1}$. Then, by the definition of ${}^{\diamond}t$, we have two possibilities:

– $\mathsf{cl} = e_{\mathsf{sc}\,A} \lhd v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_e}(\Sigma_1)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_1}(p',u) = \lambda_{\Sigma_1}(q',u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = e_{\mathsf{sc}\,A} \lhd v_{\mathsf{sc}\,A} \lhd p'$ and $q = e_{\mathsf{sc}\,A} \lhd v_{\mathsf{sc}\,A} \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p,t) = \lambda_{\Sigma_1}(p',u)$ and $\lambda_{\Sigma}(q,t) = \lambda_{\Sigma_1}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p,u) = \lambda_{\Sigma}(q,u)$ follows from (iii) and (iv).
– $\mathsf{cl} = v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl}'$, where $\mathsf{cl}' \in {}^{\diamond}u \cap \mathsf{cl_i}(\Sigma_1)$.
By the definition of net refinement, we have: (i) $\mathsf{cl} \subseteq {}^{\bullet}t \Leftrightarrow \mathsf{cl}' \subseteq {}^{\bullet}u$. And, by the induction hypothesis: (ii) $\mathsf{cl}' \subseteq {}^{\bullet}u$; and (iii) $\lambda_{\Sigma_1}(p',u) = \lambda_{\Sigma_1}(q',u)$, for all $p', q' \in \mathsf{cl}'$. Moreover, we have that $p = v_{\mathsf{sc}\,A} \lhd p'$ and $q = v_{\mathsf{sc}\,A} \lhd q'$ where $p', q' \in \mathsf{cl}'$, and by the definition of net refinement: (iv) $\lambda_{\Sigma}(p,t) = \lambda_{\Sigma_1}(p',u)$ and $\lambda_{\Sigma}(q,t) = \lambda_{\Sigma_1}(q',u)$.
Now, $\mathsf{cl} \subseteq {}^{\bullet}t$ follows from (i) and (ii). Moreover, $\lambda_{\Sigma}(p,u) = \lambda_{\Sigma}(q,u)$ follows from (iii) and (iv).

Case 2: $t = v_{\mathsf{sc}\,A} \lhd \{u, w\}$. Then we proceed similarly as in Case 1. □

Proof of proposition 18

It follows from the standard box algebra results that the underlying at-nets are in each case equal. Therefore, all we need to do is check whether the cluster filling mapping are also identical.

Case 1: $\overline{\mathfrak{A}\square\mathfrak{X}}^{\mathbb{EL}} = \overline{\mathfrak{A}}^{\mathbb{EL}}\square\mathfrak{X} = \mathfrak{A}\square\overline{\mathfrak{X}}^{\mathbb{EL}}$. Then, after denoting $\overline{\mathfrak{A}\square\mathfrak{X}}^{\mathbb{EL}} = A$, $\overline{\mathfrak{A}}^{\mathbb{EL}}\square\mathfrak{X} = B$ and $\mathfrak{A}\square\overline{\mathfrak{X}}^{\mathbb{EL}} = C$, we have a number of sub-cases:

- For $\mathsf{cl} = {}^\circ[\![\mathfrak{A}\square\mathfrak{X}]\!]$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}({}^\circ[\![\mathfrak{A}]\!]) \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}({}^\circ[\![\mathfrak{X}]\!]) \overset{(4)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = [\![\mathfrak{A}\square\mathfrak{X}]\!]^\circ$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \perp$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}([\![\mathfrak{A}]\!]^\circ) \overset{(4)}{=} \perp$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}([\![\mathfrak{X}]\!]^\circ) \overset{(4)}{=} \perp$.
- For $\mathsf{cl} = e_\square \lhd (v_\square^1 \blacktriangleleft \mathsf{cl}', v_\square^2 \blacktriangleleft {}^\circ[\![\mathfrak{X}]\!])$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}({}^\circ[\![\mathfrak{X}]\!]) \overset{(4)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = e_\square \lhd (v_\square^1 \blacktriangleleft {}^\circ[\![\mathfrak{A}]\!], v_\square^2 \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}({}^\circ[\![\mathfrak{A}]\!]) \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = v_\square^1 \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \perp$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \perp$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \perp$.
- For $\mathsf{cl} = v_\square^2 \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \perp$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \perp$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \perp$.

Case 2: $\underline{\mathfrak{A}\Box\mathfrak{X}}_{\mathbb{EL}} = \underline{\mathfrak{A}}_{\mathbb{EL}}\Box\mathfrak{X} = \mathfrak{A}\Box\underline{\mathfrak{X}}_{\mathbb{EL}}$. Then, after denoting $\underline{\mathfrak{A}\Box\mathfrak{X}}_{\mathbb{EL}} = A$, $\underline{\mathfrak{A}}_{\mathbb{EL}}\Box\mathfrak{X} = B$ and $\mathfrak{A}\Box\underline{\mathfrak{X}}_{\mathbb{EL}} = C$, we have a number of sub-cases:

- For $\mathsf{cl} = {}^\circ[\![\mathfrak{A}\Box\mathfrak{X}]\!]$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}({}^\circ[\![\mathfrak{A}]\!]) \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}({}^\circ[\![\mathfrak{X}]\!]) \overset{(5)}{=} \bot$.
- For $\mathsf{cl} = [\![\mathfrak{A}\Box\mathfrak{X}]\!]^\circ$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}([\![\mathfrak{A}]\!]^\circ) \overset{(5)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}([\![\mathfrak{X}]\!]^\circ) \overset{(5)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = e_\Box \lhd (v_\Box^1 \blacktriangleleft \mathsf{cl}', v_\Box^2 \blacktriangleleft {}^\circ[\![\mathfrak{X}]\!])$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}({}^\circ[\![\mathfrak{X}]\!]) \overset{(5)}{=} \bot$.
- For $\mathsf{cl} = e_\Box \lhd (v_\Box^1 \blacktriangleleft {}^\circ[\![\mathfrak{A}]\!], v_\Box^2 \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}({}^\circ[\![\mathfrak{A}]\!]) \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$.
- For $\mathsf{cl} = v_\Box^1 \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \bot$.
- For $\mathsf{cl} = v_\Box^2 \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(6)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(7)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$.

Case 3: $\overline{\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle}^{\mathbb{EL}} = \langle\!\langle \overline{\mathfrak{A}}^{\mathbb{EL}} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle$. Then, after denoting

$$\overline{\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle}^{\mathbb{EL}} = A \quad \text{and} \quad \langle\!\langle \overline{\mathfrak{A}}^{\mathbb{EL}} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle = B \,,$$

we have a number of sub-cases:

- For $\mathsf{cl} = {}^{\circ}\llbracket \langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle \rrbracket$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(14)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}({}^{\circ}\llbracket \mathfrak{A} \rrbracket) \overset{(4)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = \llbracket \langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle \rrbracket^{\circ}$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(14)}{=} \bot$.
- For $\mathsf{cl} = e^1_{\circledast} \lhd (v^1_{\circledast} \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(14)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}({}^{\circ}\llbracket \mathfrak{A} \rrbracket) \overset{(4)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = v^1_{\circledast} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(14)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \bot$.
- For $\mathsf{cl} = v^2_{\circledast} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(14)}{=} \bot$.
- For $\mathsf{cl} = v^3_{\circledast} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(14)}{=} \bot$.
- For $\mathsf{cl} = i_{\circledast} \lhd (v^1_{\circledast} \blacktriangleleft \llbracket \mathfrak{A} \rrbracket^{\circ}, v^2_{\circledast} \blacktriangleleft \mathsf{cl}', v^2_{\circledast} \blacktriangleleft \llbracket \mathfrak{X} \rrbracket^{\circ}, v^3_{\circledast} \blacktriangleleft {}^{\circ}\llbracket \mathfrak{V} \rrbracket)$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(14)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\llbracket \mathfrak{A} \rrbracket^{\circ}) \overset{(4)}{=} \bot$.
- For $\mathsf{cl} = i_{\circledast} \lhd (v^1_{\circledast} \blacktriangleleft \llbracket \mathfrak{A} \rrbracket^{\circ}, v^2_{\circledast} \blacktriangleleft {}^{\circ}\llbracket \mathfrak{X} \rrbracket, v^2_{\circledast} \blacktriangleleft \llbracket \mathfrak{X} \rrbracket^{\circ}, v^3_{\circledast} \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(14)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\llbracket \mathfrak{A} \rrbracket^{\circ}) \overset{(4)}{=} \bot$.

Case 4: $\underline{\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle}_{\mathbb{EL}} = \langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \underline{\mathfrak{V}}_{\mathbb{EL}} \rangle\!\rangle$. Then, after denoting $\underline{\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle}_{\mathbb{EL}} = A$ and $\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \underline{\mathfrak{V}}_{\mathbb{EL}} \rangle\!\rangle = B$, we have a number of sub-cases:

- For $\mathsf{cl} = {}^{\circ}\llbracket \langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle \rrbracket$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(16)}{=} \bot$.
- For $\mathsf{cl} = \llbracket \langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle \rrbracket^{\circ}$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \mathbb{EL}$

and
$\mathcal{M}_B(\mathsf{cl}) \overset{(16)}{=} \mathcal{M}_{\underline{\mathfrak{V}}_{\mathbb{EL}}}(\llbracket \mathfrak{V} \rrbracket^\circ) \overset{(5)}{=} \mathbb{EL}$.

– For $\mathsf{cl} = e^1_\circledast \lhd (v^1_\circledast \blacktriangleleft \mathsf{cl}')$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(16)}{=} \bot$.
– For $\mathsf{cl} = v^1_\circledast \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(16)}{=} \bot$.
– For $\mathsf{cl} = v^2_\circledast \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(16)}{=} \bot$.
– For $\mathsf{cl} = v^3_\circledast \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(16)}{=} \mathcal{M}_{\underline{\mathfrak{V}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$.
– For $\mathsf{cl} = i_\circledast \lhd (v^1_\circledast \blacktriangleleft \llbracket \mathfrak{A} \rrbracket^\circ, v^2_\circledast \blacktriangleleft \mathsf{cl}', v^2_\circledast \blacktriangleleft \llbracket \mathfrak{X} \rrbracket^\circ, v^3_\circledast \blacktriangleleft {}^\circ\llbracket \mathfrak{V} \rrbracket)$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(16)}{=} \mathcal{M}_{\underline{\mathfrak{V}}_{\mathbb{EL}}}({}^\circ\llbracket \mathfrak{V} \rrbracket) \overset{(5)}{=} \bot$.
– For $\mathsf{cl} = i_\circledast \lhd (v^1_\circledast \blacktriangleleft \llbracket \mathfrak{A} \rrbracket^\circ, v^2_\circledast \blacktriangleleft {}^\circ\llbracket \mathfrak{X} \rrbracket, v^2_\circledast \blacktriangleleft \llbracket \mathfrak{X} \rrbracket^\circ, v^3_\circledast \blacktriangleleft \mathsf{cl}')$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(16)}{=} \mathcal{M}_{\underline{\mathfrak{V}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$.

Case 5: $\langle\!\langle \underline{\mathfrak{A}}_{\mathbb{EL}} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle = \langle\!\langle \mathfrak{A} \circledast \overline{\mathfrak{X}}^{\mathbb{EL}} \circledast \mathfrak{V} \rangle\!\rangle = \langle\!\langle \mathfrak{A} \circledast \underline{\mathfrak{X}}_{\mathbb{EL}} \circledast \mathfrak{V} \rangle\!\rangle = \langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \overline{\mathfrak{V}}^{\mathbb{EL}} \rangle\!\rangle$. Then, after denoting $\langle\!\langle \underline{\mathfrak{A}}_{\mathbb{EL}} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle = A$, $\langle\!\langle \mathfrak{A} \circledast \overline{\mathfrak{X}}^{\mathbb{EL}} \circledast \mathfrak{V} \rangle\!\rangle = B$, $\langle\!\langle \mathfrak{A} \circledast \underline{\mathfrak{X}}_{\mathbb{EL}} \circledast \mathfrak{V} \rangle\!\rangle = C$ and $\langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \overline{\mathfrak{V}}^{\mathbb{EL}} \rangle\!\rangle = D$, we have a number of sub-cases:

– For $\mathsf{cl} = {}^\circ\llbracket \langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle \rrbracket$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(14)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}({}^\circ\llbracket \mathfrak{A} \rrbracket) \overset{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(15)}{=} \bot$
and
$\mathcal{M}_C(\mathsf{cl}) \overset{(15)}{=} \bot$
and
$\mathcal{M}_D(\mathsf{cl}) \overset{(16)}{=} \bot$.
– For $\mathsf{cl} = \llbracket \langle\!\langle \mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V} \rangle\!\rangle \rrbracket^\circ$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(14)}{=} \bot$
and

$\mathcal{M}_B(\mathsf{cl}) \stackrel{(15)}{=} \bot$
and
$\mathcal{M}_C(\mathsf{cl}) \stackrel{(15)}{=} \bot$
and
$\mathcal{M}_D(\mathsf{cl}) \stackrel{(16)}{=} \mathcal{M}_{\overline{\mathfrak{Y}}^{\mathbb{EL}}}(\llbracket\mathfrak{Y}\rrbracket^\circ) \stackrel{(4)}{=} \bot.$

- For $\mathsf{cl} = e^1_{\circledast} \lhd (v^1_{\circledast} \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(14)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(^\circ\llbracket\mathfrak{A}\rrbracket) \stackrel{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(15)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \stackrel{(15)}{=} \bot$
  and
  $\mathcal{M}_D(\mathsf{cl}) \stackrel{(16)}{=} \bot.$
- For $\mathsf{cl} = v^1_{\circledast} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(14)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(15)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \stackrel{(15)}{=} \bot$
  and
  $\mathcal{M}_D(\mathsf{cl}) \stackrel{(16)}{=} \bot.$
- For $\mathsf{cl} = v^2_{\circledast} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(14)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(15)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \stackrel{(15)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \bot$
  and
  $\mathcal{M}_D(\mathsf{cl}) \stackrel{(16)}{=} \bot.$
- For $\mathsf{cl} = v^3_{\circledast} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(14)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(15)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \stackrel{(15)}{=} \bot$
  and
  $\mathcal{M}_D(\mathsf{cl}) \stackrel{(16)}{=} \mathcal{M}_{\overline{\mathfrak{Y}}^{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \bot.$
- For $\mathsf{cl} = i_{\circledast} \lhd (v^1_{\circledast} \blacktriangleleft \llbracket\mathfrak{A}\rrbracket^\circ, v^2_{\circledast} \blacktriangleleft \mathsf{cl}', v^2_{\circledast} \blacktriangleleft \llbracket\mathfrak{X}\rrbracket^\circ, v^3_{\circledast} \blacktriangleleft {}^\circ\llbracket\mathfrak{Y}\rrbracket)$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(14)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\llbracket\mathfrak{A}\rrbracket^\circ) \stackrel{(5)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(15)}{=} \max\{\mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\mathsf{cl}'), \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\llbracket\mathfrak{X}\rrbracket^\circ)\} \stackrel{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_C(\mathsf{cl}) \stackrel{(15)}{=} \max\{\mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\mathsf{cl}'), \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\llbracket\mathfrak{X}\rrbracket^\circ)\} \stackrel{(5)}{=} \mathbb{EL}$

and
$\mathcal{M}_D(\mathsf{cl}) \overset{(16)}{=} \mathcal{M}_{\overline{\mathfrak{Y}}^{\mathbb{EL}}}(^\circ[\![\mathfrak{Y}]\!]) \overset{(4)}{=} \mathbb{EL}$.

– For $\mathsf{cl} = i_{\circledast} \lhd (v^1_{\circledast} \blacktriangleleft [\![\mathfrak{A}]\!]^\circ, v^2_{\circledast} \blacktriangleleft {}^\circ[\![\mathfrak{X}]\!], v^2_{\circledast} \blacktriangleleft [\![\mathfrak{X}]\!]^\circ, v^3_{\circledast} \blacktriangleleft \mathsf{cl}')$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(14)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}([\![\mathfrak{A}]\!]^\circ) \overset{(5)}{=} \mathbb{EL}$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(15)}{=} \max\{\mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(^\circ[\![\mathfrak{X}]\!]), \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}([\![\mathfrak{X}]\!]^\circ)\} \overset{(4)}{=} \mathbb{EL}$
and
$\mathcal{M}_C(\mathsf{cl}) \overset{(15)}{=} \max\{\mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(^\circ[\![\mathfrak{X}]\!]), \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}([\![\mathfrak{X}]\!]^\circ)\} \overset{(5)}{=} \mathbb{EL}$
and
$\mathcal{M}_D(\mathsf{cl}) \overset{(16)}{=} \mathcal{M}_{\overline{\mathfrak{Y}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \mathbb{EL}$.

Case 6: $\overline{\mathfrak{A} ; \mathfrak{X}}^{\mathbb{EL}} = \overline{\mathfrak{A}}^{\mathbb{EL}} ; \mathfrak{X}$. Then, after denoting $\overline{\mathfrak{A} ; \mathfrak{X}}^{\mathbb{EL}} = A$ and $\overline{\mathfrak{A} ; \mathfrak{X}}^{\mathbb{EL}} = B$, we have a number of sub-cases:

– For $\mathsf{cl} = {}^\circ[\![\mathfrak{A} ; \mathfrak{X}]\!]$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(9)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(^\circ[\![\mathfrak{A}]\!]) \overset{(4)}{=} \mathbb{EL}$.

– For $\mathsf{cl} = [\![\mathfrak{A} ; \mathfrak{X}]\!]^\circ$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(9)}{=} \bot$.

– For $\mathsf{cl} = e_; \lhd (v^1_; \blacktriangleleft \mathsf{cl}')$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(9)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \mathbb{EL}$.

– For $\mathsf{cl} = v^1_; \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(9)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \bot$.

– For $\mathsf{cl} = v^2_; \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(9)}{=} \bot$.

– For $\mathsf{cl} = i_; \lhd (v^1_; \blacktriangleleft [\![\mathfrak{A}]\!]^\circ, v^2_; \blacktriangleleft \mathsf{cl}')$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \overset{(4)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \overset{(9)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}([\![\mathfrak{A}]\!]^\circ) \overset{(4)}{=} \bot$.

Case 7: $\underline{\mathfrak{A}}_{\mathbb{EL}} ; \mathfrak{X} = \mathfrak{A} ; \overline{\mathfrak{X}}^{\mathbb{EL}}$. Then, after denoting $\underline{\mathfrak{A}}_{\mathbb{EL}} ; \mathfrak{X} = A$ and $\mathfrak{A} ; \overline{\mathfrak{X}}^{\mathbb{EL}} = B$, we have a number of sub-cases:

- For $\mathsf{cl} = {}^{\circ}[\![\mathfrak{A}\,;\mathfrak{X}]\!]$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(9)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}({}^{\circ}[\![\mathfrak{A}]\!]) \stackrel{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(10)}{=} \bot$.
- For $\mathsf{cl} = [\![\mathfrak{A}\,;\mathfrak{X}]\!]^{\circ}$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(9)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(10)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}([\![\mathfrak{X}]\!]^{\circ}) \stackrel{(4)}{=} \bot$.
- For $\mathsf{cl} = e_{;}\lhd(v^1_{;} \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(9)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(10)}{=} \bot$.
- For $\mathsf{cl} = v^1_{;} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(9)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(10)}{=} \bot$.
- For $\mathsf{cl} = v^2_{;} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(9)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(10)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \bot$.
- For $\mathsf{cl} = i_{;}\lhd(v^1_{;} \blacktriangleleft [\![\mathfrak{A}]\!]^{\circ}, v^2_{;} \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(9)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}([\![\mathfrak{A}]\!]^{\circ}) \stackrel{(5)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(10)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \mathbb{EL}$.

Case 8: $\mathfrak{A}\,;\underline{\mathfrak{X}}_{\mathbb{EL}} = \underline{\mathfrak{A}\,;\mathfrak{X}}_{\mathbb{EL}}$. Then, after denoting $\mathfrak{A}\,;\underline{\mathfrak{X}}_{\mathbb{EL}} = A$ and $\underline{\mathfrak{A}\,;\mathfrak{X}}_{\mathbb{EL}} = B$, we have a number of sub-cases:

- For $\mathsf{cl} = {}^{\circ}[\![\mathfrak{A}\,;\mathfrak{X}]\!]$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(10)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(5)}{=} \bot$.
- For $\mathsf{cl} = [\![\mathfrak{A}\,;\mathfrak{X}]\!]^{\circ}$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(10)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}([\![\mathfrak{X}]\!]^{\circ}) \stackrel{(5)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(5)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = e_{;}\lhd(v^1_{;} \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(10)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(5)}{=} \bot$.
- For $\mathsf{cl} = v^1_{;} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \stackrel{(10)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \stackrel{(5)}{=} \bot$.

– For $\mathsf{cl} = v^2_; \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \stackrel{(10)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \stackrel{(5)}{=} \bot$.
– For $\mathsf{cl} = i_; \lhd(v^1_; \blacktriangleleft \llbracket\mathfrak{A}\rrbracket^\circ, v^2_; \blacktriangleleft \mathsf{cl}')$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \stackrel{(10)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(5)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \stackrel{(5)}{=} \bot$.

Case 9: $\overline{\mathfrak{A}\|\mathfrak{X}}^{\mathbb{EL}} = \overline{\mathfrak{A}}^{\mathbb{EL}}\|\overline{\mathfrak{X}}^{\mathbb{EL}}$. Then, after denoting $\overline{\mathfrak{A}\|\mathfrak{X}}^{\mathbb{EL}} = A$ and $\overline{\mathfrak{A}}^{\mathbb{EL}}\|\overline{\mathfrak{X}}^{\mathbb{EL}} = B$, we have a number of sub-cases:

– For $\mathsf{cl} = {}^\circ\llbracket\mathfrak{A}\|\mathfrak{X}\rrbracket$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \stackrel{(4)}{=} \mathbb{EL}$
and
$\mathcal{M}_B(\mathsf{cl}) \stackrel{(12)}{=} min\{\mathbb{E}, \mathbb{E}\}max\{\mathbb{L}, \mathbb{L}\} = \mathbb{EL}$.
– For $\mathsf{cl} = \llbracket\mathfrak{A}\|\mathfrak{X}\rrbracket^\circ$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \stackrel{(4)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \stackrel{(12)}{=} \bot$.
– For $\mathsf{cl} = e^1_\| \lhd v^1_\| \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \stackrel{(4)}{=} \mathbb{EL}$
and
$\mathcal{M}_B(\mathsf{cl}) \stackrel{(12)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \mathbb{EL}$.
– For $\mathsf{cl} = e^2_\| \lhd v^2_\| \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \stackrel{(4)}{=} \mathbb{EL}$
and
$\mathcal{M}_B(\mathsf{cl}) \stackrel{(12)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \mathbb{EL}$.
– For $\mathsf{cl} = v^1_\| \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \stackrel{(4)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \stackrel{(12)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \bot$.
– For $\mathsf{cl} = v^2_\| \blacktriangleleft \mathsf{cl}'$ we have the following:
$\mathcal{M}_A(\mathsf{cl}) \stackrel{(4)}{=} \bot$
and
$\mathcal{M}_B(\mathsf{cl}) \stackrel{(12)}{=} \mathcal{M}_{\overline{\mathfrak{X}}^{\mathbb{EL}}}(\mathsf{cl}') \stackrel{(4)}{=} \bot$.

Case 10: $\underline{\mathfrak{A}}_{\mathbb{EL}}\|\underline{\mathfrak{X}}_{\mathbb{E}'\mathbb{L}'} = \underline{\mathfrak{A}\|\mathfrak{X}}_{\min\{\mathbb{E},\mathbb{E}'\}\max\{\mathbb{L},\mathbb{L}'\}}$. Then, after denoting

$$\underline{\mathfrak{A}}_{\mathbb{EL}}\|\underline{\mathfrak{X}}_{\mathbb{E}'\mathbb{L}'} = A \quad \text{and} \quad \underline{\mathfrak{A}\|\mathfrak{X}}_{\min\{\mathbb{E},\mathbb{E}'\}\max\{\mathbb{L},\mathbb{L}'\}} = B\ ,$$

we have a number of sub-cases:

- For $\mathsf{cl} = {}^{\circ}[\![\mathfrak{A}\|\mathfrak{X}]\!]$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(12)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(5)}{=} \bot$.
- For $\mathsf{cl} = [\![\mathfrak{A}\|\mathfrak{X}]\!]^{\circ}$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(12)}{=} \min\{\mathbb{E},\mathbb{E}'\}\max\{\mathbb{L},\mathbb{L}'\}$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(5)}{=} \min\{\mathbb{E},\mathbb{E}'\}\max\{\mathbb{L},\mathbb{L}'\}$.
- For $\mathsf{cl} = e^1_{\|} \lhd v^1_{\|} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(12)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(5)}{=} \bot$.
- For $\mathsf{cl} = e^2_{\|} \lhd v^2_{\|} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(12)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(5)}{=} \bot$.
- For $\mathsf{cl} = v^1_{\|} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(12)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(5)}{=} \bot$.
- For $\mathsf{cl} = v^2_{\|} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_A(\mathsf{cl}) \overset{(12)}{=} \mathcal{M}_{\underline{\mathfrak{X}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \bot$
  and
  $\mathcal{M}_B(\mathsf{cl}) \overset{(5)}{=} \bot$.

Case 11: $\overline{\mathfrak{A}\,\mathsf{sc}\,A}^{\mathbb{EL}} = \overline{\mathfrak{A}}^{\mathbb{EL}}\,\mathsf{sc}\,A$. Then, after denoting $\overline{\mathfrak{A}\,\mathsf{sc}\,A}^{\mathbb{EL}} = B$ and $\overline{\mathfrak{A}}^{\mathbb{EL}}\,\mathsf{sc}\,A = C$, we have a number of sub-cases:

- For $\mathsf{cl} = {}^{\circ}[\![\mathfrak{A}\,\mathsf{sc}\,A]\!]$ we have the following:
  $\mathcal{M}_B(\mathsf{cl}) \overset{(18)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}({}^{\circ}[\![\mathfrak{A}]\!]) \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = [\![\mathfrak{A}\,\mathsf{sc}\,A]\!]^{\circ}$ we have the following:
  $\mathcal{M}_B(\mathsf{cl}) \overset{(18)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}([\![\mathfrak{A}]\!]^{\circ}) \overset{(4)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(4)}{=} \bot$.
- For $\mathsf{cl} = e_{\mathsf{sc}\,A} \lhd (v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_B(\mathsf{cl}) \overset{(18)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(4)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_B(\mathsf{cl}) \overset{(18)}{=} \mathcal{M}_{\overline{\mathfrak{A}}^{\mathbb{EL}}}(\mathsf{cl}') \overset{(4)}{=} \bot$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(4)}{=} \bot$.

Case 12: $\underline{\mathfrak{A}\,\mathsf{sc}\,A}_{\mathbb{EL}} = \underline{\mathfrak{A}}_{\mathbb{EL}}\,\mathsf{sc}\,A$. Then, after denoting

$$\underline{\mathfrak{A}\,\mathsf{sc}\,A}_{\mathbb{EL}} = B \quad\text{and}\quad \underline{\mathfrak{A}}_{\mathbb{EL}}\,\mathsf{sc}\,A = C\ ,$$

we have a number of sub-cases:

- For $\mathsf{cl} = {}^{\circ}[\![\mathfrak{A}\,\mathsf{sc}\,A]\!]$ we have the following:
  $\mathcal{M}_B(\mathsf{cl}) \overset{(18)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}({}^{\circ}[\![\mathfrak{A}]\!]) \overset{(5)}{=} \perp$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(5)}{=} \perp$.
- For $\mathsf{cl} = [\![\mathfrak{A}\,\mathsf{sc}\,A]\!]^{\circ}$ we have the following:
  $\mathcal{M}_B(\mathsf{cl}) \overset{(18)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}([\![\mathfrak{A}]\!]^{\circ}) \overset{(5)}{=} \mathbb{EL}$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(5)}{=} \mathbb{EL}$.
- For $\mathsf{cl} = e_{\mathsf{sc}\,A} \lhd (v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl}')$ we have the following:
  $\mathcal{M}_B(\mathsf{cl}) \overset{(18)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \perp$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(5)}{=} \perp$.
- For $\mathsf{cl} = v_{\mathsf{sc}\,A} \blacktriangleleft \mathsf{cl}'$ we have the following:
  $\mathcal{M}_B(\mathsf{cl}) \overset{(18)}{=} \mathcal{M}_{\underline{\mathfrak{A}}_{\mathbb{EL}}}(\mathsf{cl}') \overset{(5)}{=} \perp$
  and
  $\mathcal{M}_C(\mathsf{cl}) \overset{(5)}{=} \perp$. □

Proof of proposition 19

If $(\mathfrak{A}, \mathfrak{X}) = (\mathfrak{A}', \mathfrak{X}')$ or then the proof is trivial. We therefore assume that $(\mathfrak{A}, \mathfrak{X}) \neq (\mathfrak{A}', \mathfrak{X}')$ and then consider four cases.

Case 1: $\mathfrak{A}\square\mathfrak{X} = \mathfrak{A}'\square\mathfrak{X}'$ iff $(\mathfrak{A}, \mathfrak{X}) \equiv_{\square} (\mathfrak{A}', \mathfrak{X}')$.
($\Longleftarrow$) Without loss of generality

$$\mathfrak{A} = \overline{\mathfrak{A}'}^{\mathbb{EL}} \text{ and } \mathfrak{X}' = \overline{\mathfrak{X}}^{\mathbb{EL}}\ .$$

Then $\overline{\mathfrak{A}'}^{\mathbb{EL}}\square\mathfrak{X} = \mathfrak{A}'\square\overline{\mathfrak{X}}^{\mathbb{EL}}$ follows from proposition 18(1)
($\Longrightarrow$) We first observe that $\mathfrak{A}\square\mathfrak{X} = \mathfrak{A}'\square\mathfrak{X}'$ implies

$$[\![\mathfrak{A}]\!]\square[\![\mathfrak{X}]\!] = [\![\mathfrak{A}']\!]\square[\![\mathfrak{X}']\!]\ .$$

Hence, from the results of the standard box algebra it follows that, without loss of generality, $[\![\mathfrak{A}]\!] = \overline{[\![\mathfrak{A}']\!]}$ and $[\![\mathfrak{X}']\!] = \overline{[\![\mathfrak{X}]\!]}$. Consequently, $\mathfrak{A}'$ and $\mathfrak{X}$ must be of the form:

$$\mathfrak{A} = \overline{\mathfrak{A}'}^{\mathbb{EL}} \quad\text{and}\quad \mathfrak{X}' = \overline{\mathfrak{X}}^{\mathbb{E}'\mathbb{L}'}\ ,$$

for some $\mathbb{EL}, \mathbb{E}'\mathbb{L}' \in \mathbb{D}$. All we need to show now is that $\mathbb{EL} = \mathbb{E}'\mathbb{L}'$. From our hypothesis and the proof of proposition 18, we know that:

$$\mathcal{M}_{\overline{\mathfrak{A}'}^{\mathbb{EL}}\square\mathfrak{X}}(\mathsf{cl}) = \mathcal{M}_{\mathfrak{A}'\square\overline{\mathfrak{X}}^{\mathbb{E}'\mathbb{L}'}}(\mathsf{cl}) \iff \mathbb{EL} = \mathbb{E}'\mathbb{L}'\ ,$$

for the cluster $\mathsf{cl} = e_{\square} \lhd (v^1_{\square} \blacktriangleleft {}^{\circ}[\![\mathfrak{A}]\!], v^2_{\square} \blacktriangleleft {}^{\circ}[\![\mathfrak{X}]\!])$. Hence $\mathbb{EL} = \mathbb{E}'\mathbb{L}'$.

Case 2: $\mathfrak{A}\,;\mathfrak{X} = \mathfrak{A}'\,;\mathfrak{X}'$ iff $(\mathfrak{A},\mathfrak{X}) \equiv_{;} (\mathfrak{A}',\mathfrak{X}')$.
$(\Longleftarrow)$ Without loss of generality

$$\mathfrak{A} = \underline{\mathfrak{A}'}_{\mathbb{EL}} \text{ and } \mathfrak{X}' = \overline{\mathfrak{X}}^{\mathbb{EL}} .$$

Then $\underline{\mathfrak{A}'}_{\mathbb{EL}}\,;\mathfrak{X} = \mathfrak{A}'\,;\overline{\mathfrak{X}}^{\mathbb{EL}}$ follows from proposition 18(3).
$(\Longrightarrow)$ We first observe that $\mathfrak{A}\,;\mathfrak{X} = \mathfrak{A}'\,;\mathfrak{X}'$ implies

$$\llbracket\mathfrak{A}\rrbracket\,;\llbracket\mathfrak{X}\rrbracket = \llbracket\mathfrak{A}'\rrbracket\,;\llbracket\mathfrak{X}'\rrbracket .$$

Hence, from the results of the standard box algebra it follows that, without loss of generality, $\llbracket\mathfrak{A}\rrbracket = \underline{\llbracket\mathfrak{A}'\rrbracket}$ and $\llbracket\mathfrak{X}'\rrbracket = \overline{\llbracket\mathfrak{X}\rrbracket}$. Consequently, $\mathfrak{A}$ and $\mathfrak{X}'$ must be of the form:

$$\mathfrak{A} = \underline{\mathfrak{A}'}_{\mathbb{EL}} \text{ and } \mathfrak{X}' = \overline{\mathfrak{X}}^{\mathbb{E}'\mathbb{L}'} ,$$

for some $\mathbb{EL}, \mathbb{E}'\mathbb{L}' \in \mathbb{D}$. All we need to show now is that $\mathbb{EL} = \mathbb{E}'\mathbb{L}'$. From our hypothesis and the proof of proposition 18 we know that:

$$\mathcal{M}_{\underline{\mathfrak{A}'}_{\mathbb{EL}}\,;\mathfrak{X}}(\mathsf{cl}) = \mathcal{M}_{\mathfrak{A}'\,;\overline{\mathfrak{X}}^{\mathbb{E}'\mathbb{L}'}}(\mathsf{cl}) \Longleftrightarrow \mathbb{EL} = \mathbb{E}'\mathbb{L}'$$

for any cluster $\mathsf{cl} = i_{;}\lhd(v^1_{;} \blacktriangleleft \llbracket\mathfrak{A}\rrbracket^{\circ}, v^2_{;} \blacktriangleleft \mathsf{cl}')$. Hence $\mathbb{EL} = \mathbb{E}'\mathbb{L}'$.

Case 3: $\mathfrak{A}\|\mathfrak{X} = \mathfrak{A}'\|\mathfrak{X}'$ iff $(\mathfrak{A},\mathfrak{X}) \equiv_{\|} (\mathfrak{A}',\mathfrak{X}')$.
$(\Longleftarrow)$ Then $\mathfrak{A} = \mathfrak{A}'$ and $\mathfrak{X} = \mathfrak{X}'$, and so $\mathfrak{A}\|\mathfrak{X} = \mathfrak{A}'\|\mathfrak{X}'$.
$(\Longrightarrow)$ We first observe that $\mathfrak{A}\|\mathfrak{X} = \mathfrak{A}'\|\mathfrak{X}'$ implies

$$\llbracket\mathfrak{A}\rrbracket\|\llbracket\mathfrak{X}\rrbracket = \llbracket\mathfrak{A}'\rrbracket\|\llbracket\mathfrak{X}'\rrbracket .$$

Hence, from the results of the standard box algebra it follows that $\llbracket\mathfrak{A}\rrbracket = \llbracket\mathfrak{A}'\rrbracket$ and $\llbracket\mathfrak{X}\rrbracket = \llbracket\mathfrak{X}'\rrbracket$. It is then easy to see that $\mathfrak{A} = \mathfrak{A}'$ and $\mathfrak{X} = \mathfrak{X}'$.

Case 4: $\langle\!\langle\mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V}\rangle\!\rangle = \langle\!\langle\mathfrak{A}' \circledast \mathfrak{X}' \circledast \mathfrak{V}'\rangle\!\rangle$ iff $(\mathfrak{A},\mathfrak{X},\mathfrak{V}) \equiv_{\circledast} (\mathfrak{A}',\mathfrak{X}',\mathfrak{V}')$.
Without loss of generality

$$\mathfrak{A} = \underline{\mathfrak{A}'}_{\mathbb{EL}} \text{ and } \mathfrak{X}' = \overline{\mathfrak{X}}^{\mathbb{EL}} \text{ and } \mathfrak{V} = \mathfrak{V}' .$$

Then $\langle\!\langle\mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V}\rangle\!\rangle = \langle\!\langle\mathfrak{A}' \circledast \mathfrak{X}' \circledast \mathfrak{V}'\rangle\!\rangle$ follows from proposition 18(2).
$(\Longrightarrow)$ We first observe that $\langle\!\langle\mathfrak{A} \circledast \mathfrak{X} \circledast \mathfrak{V}\rangle\!\rangle = \langle\!\langle\mathfrak{A}' \circledast \mathfrak{X}' \circledast \mathfrak{V}'\rangle\!\rangle$ implies

$$\langle\!\langle\llbracket\mathfrak{A}\rrbracket \circledast \llbracket\mathfrak{X}\rrbracket \circledast \llbracket\mathfrak{V}\rrbracket\rangle\!\rangle = \langle\!\langle\llbracket\mathfrak{A}'\rrbracket \circledast \llbracket\mathfrak{X}'\rrbracket \circledast \llbracket\mathfrak{V}'\rrbracket\rangle\!\rangle .$$

Hence, from the results of the standard box algebra it follows that, without loss of generality, $\llbracket\mathfrak{A}\rrbracket = \underline{\llbracket\mathfrak{A}'\rrbracket}$, $\llbracket\mathfrak{X}'\rrbracket = \overline{\llbracket\mathfrak{X}\rrbracket}$ and, moreover, $\llbracket\mathfrak{V}\rrbracket = \llbracket\mathfrak{V}'\rrbracket$ is a static at-net. Consequently, $\mathfrak{A}$ and $\mathfrak{X}'$ must be of the form:

$$\mathfrak{A} = \underline{\mathfrak{A}'}_{\mathbb{EL}} \text{ and } \mathfrak{X}' = \overline{\mathfrak{X}}^{\mathbb{E}'\mathbb{L}'} ,$$

for some $\mathbb{EL}, \mathbb{E}'\mathbb{L}' \in \mathbb{D}$, and $\mathfrak{V} = \mathfrak{V}'$. All we need to show now is that $\mathbb{EL} = \mathbb{E}'\mathbb{L}'$. From our hypothesis and the proof of proposition 18 we know that:

$$\mathcal{M}_{\langle\!\langle\underline{\mathfrak{A}'}_{\mathbb{EL}}\circledast\mathfrak{X}\circledast\mathfrak{V}\rangle\!\rangle}(\mathsf{cl}) = \mathcal{M}_{\langle\!\langle\mathfrak{A}'\circledast\overline{\mathfrak{X}}^{\mathbb{E}'\mathbb{L}'}\circledast\mathfrak{V}\rangle\!\rangle}(\mathsf{cl}) \Longleftrightarrow \mathbb{EL} = \mathbb{E}'\mathbb{L}'$$

for any cluster $\mathsf{cl} = i_{\circledast}\lhd(v^1_{\circledast} \blacktriangleleft \llbracket\mathfrak{A}\rrbracket^{\circ}, v^2_{\circledast} \blacktriangleleft \mathsf{cl}', v^2_{\circledast} \blacktriangleleft \llbracket\mathfrak{X}\rrbracket^{\circ}, v^3_{\circledast} \blacktriangleleft {}^{\circ}\llbracket\mathfrak{V}\rrbracket)$. Hence $\mathbb{EL} = \mathbb{E}'\mathbb{L}'$. $\square$