Recursion and Petri Nets

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

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

Abstract. This paper shows how to define Petri nets through recursive equations. It specifically addresses this problem within the context of the *box algebra*, a model of concurrent computation which combines Petri nets and standard process algebras. The paper presents a detailed investigation of the solvability of recursive equations on nets in what is arguably the most general setting. For it allows an infinite number of possibly unguarded equations, each equation possibly involving infinitely many recursion variables. The main result is that by using a suitable partially ordered domain of nets, it is always possible to solve a system of equations by constructing the limit of a chain of successive approximations.
Keywords: Petri nets, box algebra, refinement, recursion, approximation, limit construction, guardedness.

1 Introduction

The box algebra (see [4, 7, 21]) is a formal model of non-sequential computation which combines concepts derived from two areas of concurrency theory, viz. Petri nets [2, 24, 27] and process algebras [1, 19, 20, 23]. It is based on a set of process expressions, called *box expressions*. Each box expression has associated two (consistent) kinds of semantics: a Petri net called a *box* (with its standard Petri net transition firing rule), and an operational semantics defined using the usual SOS derivation rules [26]. Boxes are essentially (safe) labelled Petri nets enriched with the possibility of algebraic manipulation (in the same way as expressions). A particular instance of the box algebra is PBC (Petri Box Calculus) [3, 4] whose development was a direct inspiration for the introduction of the box algebra. The technique of associating a Petri net with a process algebra expressions has also been studied for various other process algebras [10, 13, 17, 18, 25, 28].

The box algebra is parameterised by a set of operators, OpBox, which can be used to construct valid box expressions. For each operator $\mathsf{op} \in \mathsf{OpBox}$, there is an associated operator in the domain of boxes, Ω_{op}. This allows one to compositionally define, for every composite box expression $E = \mathsf{op}(E_1, E_2, \ldots)$, a corresponding net, $\mathsf{box}(E) = \Omega_{\mathsf{op}}(\mathsf{box}(E_1), \mathsf{box}(E_2), \ldots)$; and, for every constant K, a net $\mathsf{box}(K)$. Common operators are sequence (denoted by the semicolon), choice (denoted by $\square$), and parallel composition (denoted by $\|$). Indeed, all three operators are used in PBC. However, the box algebra allows one to define much richer (even infinite) sets of operators, as we shall explain later on.

The box algebra allows one to formulate recursive definitions of the form $X \stackrel{\mathrm{df}}{=} \mathsf{op}(K_1, K_2, \ldots)$, where X is a variable, and each K_i is a variable or a constant. Thus, for example, the variable X defined by $X \stackrel{\mathrm{df}}{=} \alpha; X$ describes a system which can indefinitely perform an α-move and never terminate. In the domain of boxes, the equation $X \stackrel{\mathrm{df}}{=} \mathsf{op}(K_1, K_2, \ldots)$ is transformed into a corresponding equation on nets, $X \stackrel{\mathrm{df}}{=} \Omega_{\mathsf{op}}(L_1, L_2, \ldots)$, where $L_i = K_i$ if K_i is a variable, and $K_i = \mathsf{box}(K_i)$ if K_i is a constant. It was shown in [9, 21] that such equations can be solved using an approach based on successive approximations, starting from particularly simple nets. This was possible since the operator Ω_{op} induced a continuous mapping in a partially ordered domain of boxes. What is more, the resulting solutions possessed a (highly) desirable property of generating the same behaviour as that of the construct $X \stackrel{\mathrm{df}}{=} \mathsf{op}(K_1, K_2, \ldots)$ executed under the SOS operational semantics rules. However, the results of [9, 21] were only applicable to finite operators; i.e., it was assumed that $\mathsf{op}(K_1, K_2, \ldots) = \mathsf{op}(K_1, K_2, \ldots, K_n)$, for some natural number n. In this paper we remove this restriction, and assume that op can in general take any number of arguments, in particular infinitely many. The interest in such general operators is motivated by a wish to model operators like $\square_{i \in I} E_i$, i.e., a fully generalised choice operator of Milner's CCS [23]. The latter, in turn, can be used to give a formal semantics to a process algebra with value-passing. In addition to treating operators with infinitely many arguments, the present paper extends the results of [9,21] also in the sense that we now admit more general types of nets. The interest here is that iterative constructors such as Ω_* in figure 2 can be accommodated in full generality.

The paper is organised as follows. In the next section, we introduce various classes of Petri nets used throughout the paper. Section 3 defines net refinement, the basic tool used to structure nets in the box algebra. Sections 4 and 5 constitute the central part of the paper; we show that a recursive equation has always solutions which can be approximated by starting from particularly simple nets. Section 6 investigates the simplifying consequences of assuming that recursive equations satisfy various kinds of finiteness properties. Section 7 presents examples illustrating and explaining the theory. Section 8 extends the results developed for a single recursive equation to a fully general case of an arbitrary system of equations.

This paper subsumes and extends our previous work in this area, contained in [3,6,9,15,16,21]. A companion paper [8] addresses, again in full generality, the question of behavioural consistency between Petri nets and process algebras. Although in our examples we will use finite operators and plain boxes, in order to be able to represent them graphically, the overall theory will be carefully devised to remain valid for infinite nets as well.

Throughout the paper we use the standard mathematical notation. In particular, we use $\mathsf{mult}(Z)$ to denote the set of all finite multisets over a set Z. Other multiset-related notation includes the sum (+), difference (-, only used when the second argument is not greater than the first one) and multiplication by a natural number ($\cdot$).

2 The domain of Petri nets

The approach to compositionality adopted by the box algebra is based on transition refinement, i.e., each operator on nets will be based on a special net Ω_{op} whose transitions $t_1, t_2, \ldots$ are refined by the corresponding nets $\Sigma_1, \Sigma_2, \ldots$ in the process of forming a new net $\Sigma = \Omega_{\mathsf{op}}(\Sigma_1, \Sigma_2, \ldots)$. To carry this out, we need to be able to distinguish those (labelled) nets that are easily composable with one another. The chosen class of Petri nets, called *boxes*, have interfaces expressed by labellings of places and transitions. There are two main classes of boxes which we will be interested in, viz. plain boxes and operator boxes. Plain boxes (the $\Sigma_1, \Sigma_2, \ldots$ above) form the class of elements of our Petri net domain upon which various operators are defined. Operator boxes (the Ω_{op} above) are patterns (or functions) defining the ways of constructing new plain boxes out of given ones.

Actions and relabellings: We assume a set Lab of *actions* to be given. When executing a transition, or a set of concurrently enabled transitions, we shall be able to consider the action (or multiset of actions), Γ, with which it (or they) are labelled. A *relabelling* ρ is a relation

$$\rho \subseteq \mathsf{mult}(\mathsf{Lab}) \times \mathsf{Lab} \tag{1}$$

such that $(\emptyset, \alpha) \in \rho$ if and only if $\rho = \{(\emptyset, \alpha)\}$. The intuition behind a pair (Γ, α) belonging to ρ is that it specifies a synchronisation of the transitions giving rise to Γ, followed by an interface change, yielding a transition labelled by α.

Three specific kinds of relabelling are of particular interest. A *constant* relabelling, $\rho_\alpha = \{(\emptyset, \alpha)\}$, where α is an action in Lab, can be identified with α itself, so that we may consider the set of actions Lab to be embedded in the set of all relabellings[1] (and, by $(\emptyset, \alpha) \in \rho \Leftrightarrow \rho = \{(\emptyset, \alpha)\}$, retrieve actions from relabellings). The *restriction* $\rho_A = \{(\{\alpha\}, \alpha) \mid \alpha \in A\}$ only keeps the actions belonging to some set $A \subseteq \mathsf{Lab}$. And the *identity* relabelling, $\rho_{id} = \{(\{\alpha\}, \alpha) \mid \alpha \in \mathsf{Lab}\}$ captures the 'keep things as they are' interface (non)change. Note that $\rho_{id} = \rho_{\mathsf{Lab}}$.

A non-constant relabelling will be called *transformational*, and so both ρ_A and ρ_{id} are transformational relabellings.

[1] It is preferable to proceed in this way, rather than to use $\mathsf{Lab} \cup \{\rho \mid \rho \text{ is a relabelling}\}$ as the set of labels. For Lab could itself be a set of relabellings making it difficult to distinguish between plain and operator boxes.

Relabellings have been defined in this generality for they are meant to encompass both normal labels Lab (then they are constant) and all communication interface operators such as restriction and synchronisation (then they are transformational). We will specify the transformational relabelling each time a concrete communication interface operator is defined.

Labelled nets: We define a *(marked) labelled net* to be a tuple $\Sigma = (S, T, W, \lambda, M)$ such that: S and T are disjoint sets of respectively *places* and *transitions*; W is a *weight function* from the set $(S \times T) \cup (T \times S)$ to the set of natural numbers $\mathbf{N} = \{0, 1, 2, \ldots\}$; λ is a *labelling* function for places and transitions such that $\lambda(s) \in \{\mathsf{e}, \mathsf{i}, \mathsf{x}\}$, for every place $s \in S$, and $\lambda(t)$ is a relabelling ρ of the form (1), for every transition $t \in T$; and M is a *marking*, i.e., a mapping assigning a natural number to each place $s \in S$.

We adopt the standard rules about representing nets as directed graphs, viz. places are represented as circles, transitions as rectangles, the flow relation generated by W is indicated by arcs annotated with the corresponding weights, and markings are shown by placing tokens within circles. As usual, the zero weight arcs will be omitted and the unit weight arcs (or unitary arcs) will be left unannotated. To avoid ambiguity, we will sometimes decorate the various components of Σ with the index Σ; thus, T_Σ denotes the set of transitions of Σ, etc. A net is *finite* if both S and T are finite sets. Figure 1 shows the graph of a finite labelled net $\Sigma_0 = (S_0, T_0, W_0, \lambda_0, M_0)$ defined thus:

$$
\begin{aligned}
S_0 &= \{s_0, s_1, s_2, s_3\} \\
T_0 &= \{t_0, t_1, t_2\} \\
W_0 &= ((TS \cup ST) \times \{1\}) \ \cup \ (((S \times T) \setminus ST \ \cup \ (T \times S) \setminus TS) \times \{0\}) \\
\lambda_0 &= \{(s_0, \mathsf{e}), (s_1, \mathsf{i}), (s_2, \mathsf{x}), (s_3, \mathsf{e}), (t_0, a), (t_1, b), (t_2, a)\} \\
M_0 &= \{(s_0, 1), (s_1, 0), (s_2, 0), (s_3, 1)\}.
\end{aligned}
$$

where $TS = \{(t_0, s_1), (t_1, s_2), (t_2, s_3)\}$ and $ST = \{(s_0, t_0), (s_1, t_1), (s_3, t_2)\}$.

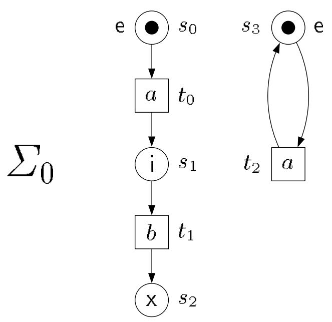


Fig. 1. A labelled net.

If the labelling of a place s in a labelled net Σ is e then s is an *entry* place, if i then s is an *internal* place, and if x then s is an *exit* place. By convention, ${}^\circ\Sigma$, Σ° and $\ddot{\Sigma}$ denote respectively the entry, exit and internal places of Σ. For every place (transition) x, we use ${}^\bullet x$ to denote is pre-set, i.e., the set of all transitions (places) y such that there is an arc from y to x, $W(y, x) > 0$. The post-set $x^\bullet$ is defined in a similar way. The pre- and post-set notation extends in the usual way to sets R of places and transitions, e.g., ${}^\bullet R = \bigcup\{{}^\bullet r \mid r \in R\}$. In what follows, all nets are assumed to be T-*restricted*, i.e., the pre- and post-sets of each transition are non-empty. No assumption of that kind is made for places. For the labelled net of figure 1 we have ${}^\circ\Sigma_0 = \{s_0, s_3\}$, $\Sigma_0^\circ = \{s_2\}$, $\ddot{\Sigma}_0 = \{s_1\}$, ${}^\bullet s_0 = \emptyset$, $s_0^\bullet = \{t_0\}$ and $\{s_0, s_1\}^\bullet = \{t_0, t_1\} = {}^\bullet\{s_1, s_2\}$.

A labelled net Σ is *ex-restricted* if there is at least one entry and at least one exit place, ${}^\circ\Sigma \neq \emptyset \neq \Sigma^\circ$; and *simple* if W always returns 0 or 1.

Step sequence semantics: We adopt finite step sequence semantics for a labelled net $\Sigma = (S, T, W, \lambda, M)$. A finite multiset of transitions U, called a *step*, is *enabled* by Σ if for every place $s \in S$, $M(s) \geq \sum_{t \in U} W(s,t) \cdot U(t)$. An enabled step U can be *executed*, leading to a follower marking M' defined, for every place $s \in S$, by

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

We will denote this by $M\ [U\rangle\ M'$. A *finite step sequence* of Σ is a finite (possibly empty) sequence $\sigma = U_1 \ldots U_k$ of steps for which there are markings $M_0, \ldots, M_k$ such that $M_\Sigma = M_0$ and, for every $1 \leq i \leq k$, $M_{i-1}\ [U_i\rangle\ M_i$. This shall be denoted by $M_\Sigma\ [\sigma\rangle\ M_{\Sigma_k}$ and the marking M_{Σ_k} will be called *reachable* from M_Σ.

Consider again the marked labelled net Σ_0 in figure 1. In the initial marking, transitions t_0 and t_2 are enabled concurrently and so $\{t_0, t_2\}$ is an enabled step. After this step has been executed, transitions t_1 and t_2 are enabled concurrently and, hence, $\{t_0, t_2\}\{t_1, t_2\}$ is a step sequence of the net from the marking shown in figure 1. A step does not need to be maximal. Thus, for instance, $\{t_0\}$ is also a step of Σ_0, and both $\{t_0\}\{t_1\}\{t_2\}\{t_2\}$ and $\{t_0, t_2\}\{t_1\}\{t_2\}$ are step sequences.

Plain and operator boxes: A *box* is an ex-restricted and T-restricted labelled net Σ.[2] A box is *plain* if its transitions are all labelled by constant relabellings.[3] Plain boxes with empty markings will be called *rec-boxes* (or recursion boxes), and their set will be denoted by Box^{rec}. A plain box without any transitions nor internal places will be called an ex-*box*. An *operator box* is a simple box Ω all of whose relabellings are transformational. Figure 2 shows one plain box and five operator boxes of the PBC algebra[4] [6, 7]. The plain box N_α models a single action expression, α. The operator boxes $\Omega_{\mathsf{rs}\,A}$, $\Omega_;$, $\Omega_{\Box}$, $\Omega_{\|}$ and Ω_* model respectively restriction, sequence, choice, concurrent composition, and iteration.

Let $\mathbf{\Sigma}$ be a function from the transitions of Ω to rec-boxes. We will refer to $\mathbf{\Sigma}$ as an Ω-*tuple* and denote $\mathbf{\Sigma}(v)$ by Σ_v, for every $v \in T_\Omega$. If the set of transitions of Ω is finite, we will assume that $T_\Omega = \{v_1, \ldots, v_n\}$ is implicitly ordered, and denote $\mathbf{\Sigma} = (\Sigma_{v_1}, \ldots, \Sigma_{v_n})$. For $T \subseteq T_\Omega$, we will use $\mathbf{\Sigma}_T$ to denote the restriction of $\mathbf{\Sigma}$ to the transitions in T.

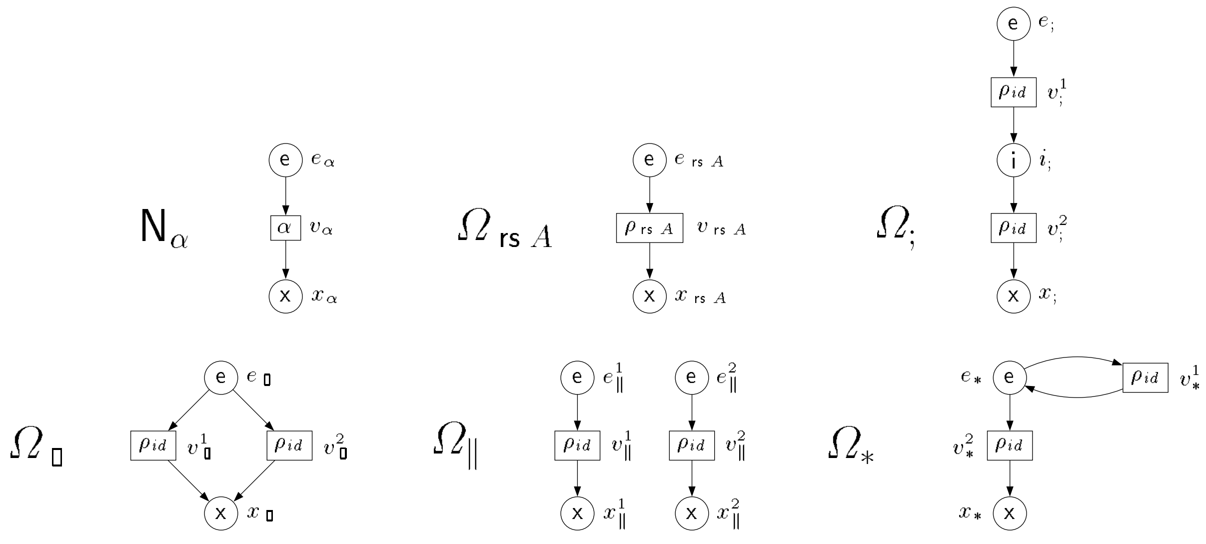


Fig. 2. A plain box N_α and five operator boxes for PBC, where A is a set of actions and $\rho_{\mathsf{rs}\,A} = \rho_A$.

[2] We will explain in sections 3 and 7.3 why the ex-restrictedness property is important.
[3] Or actions, according to our convention of identifying constant relabellings with actions.
[4] There are also other PBC constants and operators, not shown in figure 2.

3 Net refinement

The basis for providing plain boxes with an algebraic structure is a general *simultaneous refinement and relabelling meta-operator* (*net refinement*, for short), which embodies a mechanism combining transition refinement and interface change defined by relabellings. It is defined for an operator box Ω which serves as a pattern for gluing together an Ω-tuple of rec-boxes Σ (one plain box for every transition in Ω) along their entry and exit interfaces. The relabellings annotating the transitions of Ω specify the interface changes for the boxes in Σ. Recursive systems are defined on unmarked boxes (i.e., those with empty markings), so that we limit the definition of refinement to this case.

Place and transition names of operator and plain boxes: As far as net refinement as such is concerned, the names (identities) of newly constructed transitions and places are irrelevant, provided that we always choose them fresh. However, in our approach to solving recursive definitions on boxes, the *names* of places and transitions are crucial since we use them to define the inclusion order on the domain of labelled nets.

A key to the construction of recursive nets is the use of labelled trees as place and transition names.[5] First of all, we shall assume that there are two disjoint infinite sets of place and transition names, $\mathsf{P_{root}}$ and $\mathsf{T_{root}}$. Each name $\eta \in \mathsf{P_{root}} \cup \mathsf{T_{root}}$ (or a pair (t, α), where t is a name in $\mathsf{T_{root}}$ and α is a label in Lab) can be viewed as a special tree with a single node (both a root and a leaf) labelled with η (or (t, α)), called a single root tree. Moreover, we shall allow more complicated trees as transition and place names, and use a linear notation to express such trees. To this end, the expression $x \lhd \mathcal{S}$, where x is a single root tree labelled with x (x is a name η or a pair (t, α)) and $\mathcal{S}$ is a multiset of trees, is a new tree where the trees of the multiset are appended (with their multiplicity) to the root. If $\mathcal{S} = \{p\}$ is a singleton multiset then $x \lhd \mathcal{S}$ will simply be denoted by $x \lhd p$, and if $\mathcal{S}$ is empty then $x \lhd \mathcal{S} = x$.

We shall further assume that in every operator box, all places and transitions are simply names (i.e., single root trees) from respectively $\mathsf{P_{root}}$ and $\mathsf{T_{root}}$. For the plain boxes, the trees used as names may be more complex. Each transition tree is a finite tree labelled with elements of $\mathsf{T_{root}}$ (at the leaves) and $\mathsf{T_{root}} \times \mathsf{Lab}$ (elsewhere), and each place tree is a possibly infinite (in depth and width) tree labelled with names from $\mathsf{P_{root}}$ and $\mathsf{T_{root}}$, which has the form:

$$t_1 \lhd t_2 \lhd \ldots \lhd t_n \lhd s \lhd \mathcal{S},$$

where $t_1, \ldots, t_n \in \mathsf{T_{root}}$ $(n \geq 0)$ are transition names and $s \in \mathsf{P_{root}}$ is a place name (so that no confusion will be possible between transition-trees and place-trees: the latter always have a label from $\mathsf{P_{root}}$ and the former never). We comprise all these trees (including the basic ones consisting only of a root as special cases) in our sets of allowed transition and place names, denoted respectively by $\mathsf{P_{tree}}$ and $\mathsf{T_{tree}}$. The definition of net refinement will be done in such a way that, provided all the names occurring in Ω are single root trees, and all the names occurring in Σ are in the set $\mathsf{P_{tree}} \cup \mathsf{T_{tree}}$ of allowed names, then all the names in $\Omega(\Sigma)$ belong there, too.

The definition of $\Omega(\Sigma)$ Under the assumptions made about the operator box Ω and Ω-tuple of rec-boxes Σ, the result of a simultaneous substitution of boxes Σ for the transitions in Ω is a labelled net $\Omega(\Sigma)$ with the empty marking whose components are defined thus.

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

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

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

$$p \;=\; s \lhd (\{v \lhd x_v\}_{v \in {}^\bullet s} + \{w \lhd e_w\}_{w \in s^\bullet}) \;\in\; \mathsf{SP}^s_{\mathsf{new}} \tag{2}$$

[5] We define such trees up to isomorphism. Moreover, one can always think of these trees as simply an additional injective labelling for places and transitions of a labelled net (whose nodes are then defined up to isomorphism).

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

Let y be a transition in Ω. Then for $p = v \triangleleft i$ in $\mathsf{ST}^v_{\mathsf{new}}$, we define $\mathsf{trees}^y(p) = \{i\}$ if $v = y$, and $\mathsf{trees}^y(p) = \emptyset$ if $v \neq y$; and, for $p \in \mathsf{SP}^s_{\mathsf{new}}$ as in (2), we define

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

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

$$T_{\Omega(\mathbf{\Sigma})} = \bigcup_{v \in T_\Omega} \mathsf{T}^v_{\mathsf{new}} = \bigcup_{v \in T_\Omega} \{(v,\alpha) \triangleleft R \mid R \in \mathsf{mult}(T_{\Sigma_v}) \wedge (\lambda_{\Sigma_v}(R), \alpha) \in \lambda_\Omega(v)\}.$$

Notice that the multiset R in $(v,\alpha)\triangleleft R$ will never be empty since no pair in $\lambda_\Omega(v)$ has the empty multiset as its left argument (since all relabellings in an operator box are transformational).

Similarly as for places, we will denote by $\mathsf{trees}(u)$ the multiset of transitions R upon which a newly constructed transition $u = (v,\alpha)\triangleleft R \in \mathsf{T}^v_{\mathsf{new}}$ is based.

Labelling: The label of a node z in $\Omega(\mathbf{\Sigma})$ is defined in the following way:

$$\lambda_{\Omega(\mathbf{\Sigma})}(z) = \begin{cases} \mathsf{i} & \text{if } z \in \mathsf{ST}^v_{\mathsf{new}} \\ \lambda_\Omega(s) & \text{if } z \in \mathsf{SP}^s_{\mathsf{new}} \\ \alpha & \text{if } z = (v,\alpha)\triangleleft R \in \mathsf{T}^v_{\mathsf{new}}. \end{cases}$$

Weights: For a place p and transition u in $\mathsf{T}^v_{\mathsf{new}}$, the weight function is given by:

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

Figure 3 shows a complete example of net refinement, including the tree names of newly created places and transitions. In the linear notation we have, for example, $\tau_1 = e\triangleleft\{v_1\triangleleft e_1^2, v_1\triangleleft x_1, v_2\triangleleft e_2\}$, $\tau_2 = v_1\triangleleft i_1^2$ and $\xi_2 = (v_1,g)\triangleleft\{t_a,t_b\}$. Moreover, $\mathsf{trees}^{v_1}(\tau_1) = \{e_1^2, x_1\}$, $\mathsf{trees}^{v_2}(\tau_1) = \emptyset$, $\mathsf{trees}^{v_2}(\tau_2) = \{i_1^2\}$ and $\mathsf{trees}(\xi_2) = \{t_a,t_b\}$

Remarks: One can easily show that the result of net refinement, $\Omega(\mathbf{\Sigma})$, is always a plain box. It is, in general *not* the case that if we start from boxes satisfying some desirable (behavioural) properties, then the result will retain them as well. Indeed, we show in [5] that some extra – intuitively well justified – conditions need to be introduced for this to hold, even for some basic behavioural properties. In this paper we will essentially be concerned with the net theoretic (structural) properties of net refinement. However, in all examples the conditions we mentioned are satisfied, and the reader is urged to verify that the behaviour of the constructed nets is indeed the expected one.

The mechanism we have described is compositional since the definition is based on, and respects, the structure of the components. More precisely, one can easily see that $\Omega(\mathbf{\Sigma})$ is a rec-box. Moreover, the T-restrictedness of $\Omega(\mathbf{\Sigma})$ is basically due to the T-restrictedness of Ω and the ex-restrictedness of all the boxes in $\mathbf{\Sigma}$.

Net refinement has a well formed graph-theoretic definition and, as a result, it is insensitive to taking isomorphic instances of nets. That is, isomorphism is preserved through net refinement. We also remark that the above definition may also be applied to boxes with non-empty markings (see, e.g., [3,7]), but for the purposes of this paper such an extension is irrelevant.

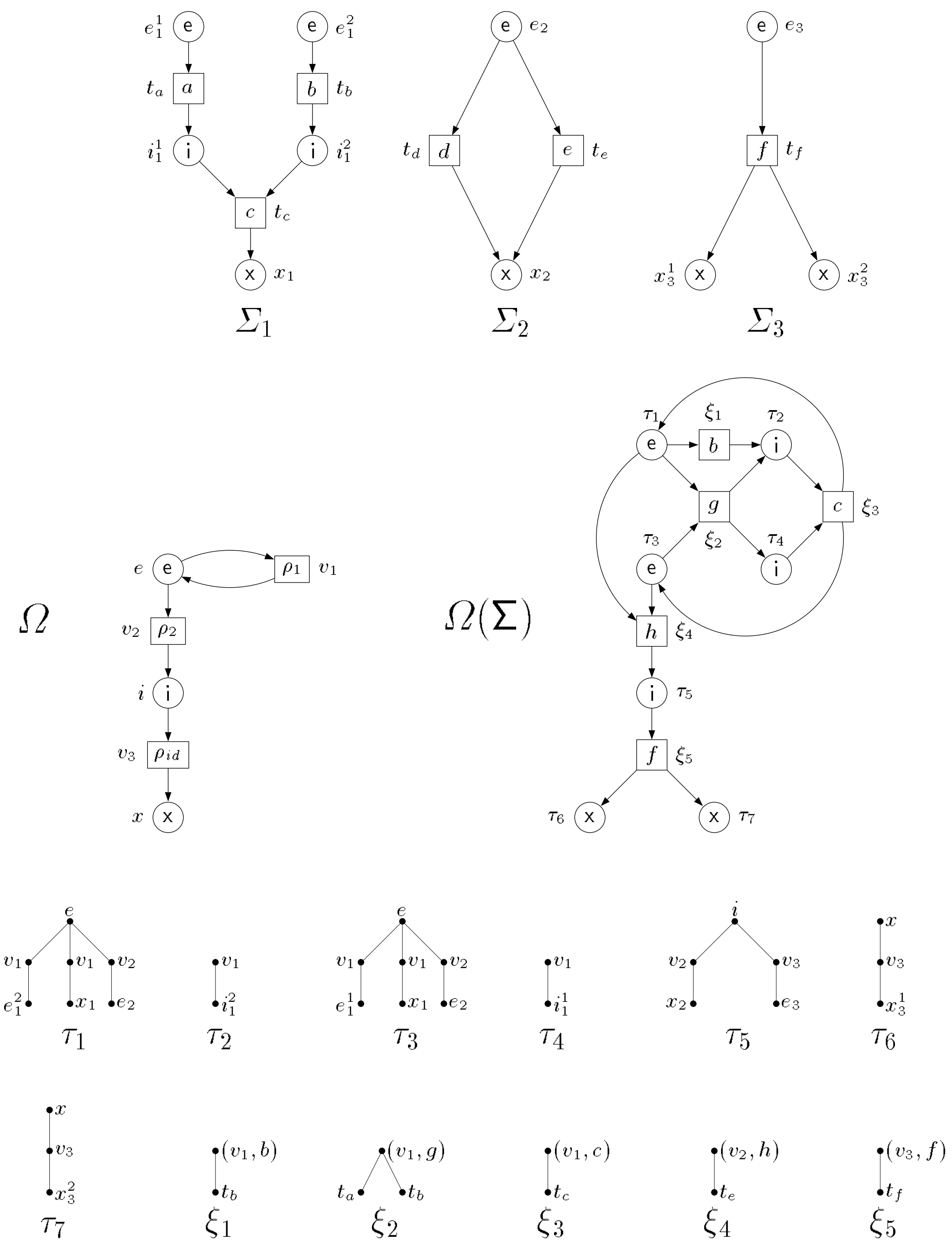


Fig. 3. Net refinement $\Omega(\Sigma)$ where $\rho_1 = \rho_{id} \backslash \{(\{a\}, a)\} \cup \{(\{a, b\}, g)\}$ and $\rho_2 = \rho_{id} \backslash \{(\{e\}, e), (\{d\}, d)\} \cup \{(\{e\}, h)\}$.

4 Inclusion order on labelled nets

We equip the set of rec-boxes with an *ordering relation*, $\sqsubseteq$. For two rec-boxes Σ and Θ, we denote $\Sigma \sqsubseteq \Theta$ if $S_\Sigma \subseteq S_\Theta$, $T_\Sigma \subseteq T_\Theta$, $W_\Sigma = W_\Theta|_{(S_\Sigma \times T_\Sigma)\cup(T_\Sigma \times S_\Sigma)}$ and $\lambda_\Sigma = \lambda_\Theta|_{S_\Sigma \cup T_\Sigma}$. The relation $\sqsubseteq$ is a partial order, and if $\Sigma \sqsubseteq \Theta$ then ${}^\circ\Sigma \subseteq {}^\circ\Theta$ and $\Sigma^\circ \subseteq \Theta^\circ$. A $\sqsubseteq$-*chain* is an infinite sequence

$$\chi = \Sigma^0 \ \Sigma^1 \ \Sigma^2 \ \ldots$$

of rec-boxes such that $\Sigma^k \sqsubseteq \Sigma^{k+1}$ for every $k \geq 0$. Each such chain has a unique least upper bound or *limit* (a labelled net with the empty marking), denoted $\bigsqcup\chi$; the Σ^k's will be called the *approximations* of $\bigsqcup\chi$. The places and transitions of $\bigsqcup\chi$ are respectively the union of the places and transitions of all labelled nets in the chain. For any two elements x and y of $\bigsqcup\chi$ (one of them a place, the other a transition), the weight $W(x, y)$ is defined as the weight between x and y in the first Σ^k to which both of them belong (note that, by the definition of $\sqsubseteq$, this weight does not change in subsequent approximations). Finally, for a place or transition x in $\bigsqcup\chi$, the label $\lambda(x)$ is defined as the label of x in the first and hence in all Σ^k containing x.

A mapping ψ from labelled nets to labelled nets is $\sqsubseteq$-*chain-preserving* if, for every $\sqsubseteq$-chain $\chi = \Sigma^0 \ \Sigma^1 \ \ldots$ of labelled nets, the sequence $\psi(\chi) = \psi(\Sigma^0) \ \psi(\Sigma^1) \ \ldots$ is also a $\sqsubseteq$-chain.

Let $\mathcal{C}$ be a property of labelled nets. We will say that $\mathcal{C}$ is *limit-preserved* if for every $\sqsubseteq$-chain χ of labelled nets satisfying $\mathcal{C}$, it is the case that $\bigsqcup\chi$ satisfies $\mathcal{C}$ as well.

Proposition 1 The following properties of labelled nets are limit-preserved.

(1) There is at least one entry place and at least one exit place.
(2) Every transition has a non-empty pre-set and non-empty post-set.
(3) All transition labels are constant relabellings.

Proof: Follows directly from the definitions of a chain and its limit. □

5 Solving a recursive equation

Until section 8, we shall only consider a *single* recursive equation for a net variable X:

$$X \stackrel{\mathrm{df}}{=} \Omega(\Delta) \tag{3}$$

where Ω is an operator box, and $\Delta : T_\Omega \to \mathsf{Box}^{rec} \cup \{X\}$ is an Ω-tuple such that each Δ_v is either a rec-box or X. A rec-box Σ will be a *solution* of (3) whenever $\Sigma = \Omega(\Delta[\Sigma])$, where $\Delta[\Sigma]$ is Δ with each X replaced by Σ (and the '=' means actual equality). Without loss of generality, we shall also assume that for every rec-box Δ_v in Δ, it is the case that $T_{\Delta_v} \subseteq \mathsf{T}_{\mathsf{root}} \backslash T_\Omega$ and $S_{\Delta_v} \subseteq \mathsf{P}_{\mathsf{root}} \backslash S_\Omega$.[6]

For example, in PBC, the solution of the recursive net equation $X \stackrel{\mathrm{df}}{=} \Omega_{\|}(\mathsf{N}_\alpha, X)$ yields a box model for the (process) variable X defined by $X \stackrel{\mathrm{df}}{=} \alpha \| X$, and the solution of $X \stackrel{\mathrm{df}}{=} \Omega_{;}(\mathsf{N}_\alpha, X)$ does the same for $X \stackrel{\mathrm{df}}{=} \alpha; X$. Since we only allow a single net operator on the right hand side, more complex defining equations need to be decomposed into equivalent simple recursive systems which is always possible.

In general, the recursive equation (3) may have several different, even non-isomorphic, solutions; but each solution is uniquely identified by the sets of entry and exit places. A formal statement of this fact is preceded by an auxiliary lemma which shows that all solutions of (3) have the same set of transitions, and that the inclusion order $\sqsubseteq$ on the set of possible solutions is equivalent to the ordering resulting from the ordinary set inclusion on their entry and exit places.

[6] Recall that, by the definition of an operator box, $T_\Omega \subseteq \mathsf{T}_{\mathsf{root}}$ and $S_\Omega \subseteq \mathsf{P}_{\mathsf{root}}$.

Lemma 2 Let Σ and Θ be solutions of (3).

(1) The sets of transitions of Σ and Θ and their labellings are the same.
(2) The following are equivalent:
  (a) ${}^\circ\Sigma \subseteq {}^\circ\Theta$ and $\Sigma^\circ \subseteq \Theta^\circ$.
  (b) $\Sigma \sqsubseteq \Theta$.

Proof: Let us denote $\Sigma = (S_0, T_0, W_0, \lambda_0)$ and $\Theta = (S_1, T_1, W_1, \lambda_1)$.

(1) Suppose $T_0 \not\subseteq T_1$ and $t = (v, \alpha) \lhd R \in T_0$ is a smallest (with respect to the number of nodes) transition tree such that $t \notin T_1$ (we can assume that such a t exists since all transition trees are finite). Knowing that $\Sigma = \Omega(\Delta[\Sigma])$, we proceed as follows.
If $\Delta_v \in \mathsf{Box}^{rec}$ then R is a multiset of transitions of Δ_v and, since $\Delta[\Sigma]_v = \Delta_v = \Delta[\Theta]_v$ and $\Theta = \Omega(\Delta[\Theta])$, we have $t \in T_1$ which produces a contradiction.
If $\Delta_v = X$ then all transitions in R are also in T_0. Now, if R is also a multiset of transitions in T_1 then, by $\Theta = \Omega(\Delta[\Theta]))$ and $\Delta[\Theta]_v = \Theta$, $t \in T_1$ contradicting our original assumption. Hence there is u in R such that $u \notin T_1$. Moreover, u is a proper sub-tree of t, which produces a contradiction with the minimality of t.
Hence $T_0 \subseteq T_1$. Similarly, one can show that $T_1 \subseteq T_0$, and so $T_0 = T_1$.

That the labellings of the transitions in Σ and Θ are the same follows from the definition of net refinement and $\Sigma = \Omega(\Delta[\Sigma])$ and $\Theta = \Omega(\Delta[\Theta])$, which together imply that each transition is of the form $(v, \alpha) \lhd R$ and has the label α.

(2) The (a) $\Leftarrow$ (b) implication follows directly from the definition of $\sqsubseteq$. Below we prove that (a) $\Rightarrow$ (b).

We first show that all places of Σ are also in Θ, with the same labels. Since every place p of Σ is a tree of the form $t_1 \lhd \ldots t_k \lhd s \lhd \mathcal{S}$, where $t_1, \ldots, t_k \in \mathsf{T}_{\mathsf{root}}$ and $s \in \mathsf{P}_{\mathsf{root}}$, we can proceed by induction on k. For $k = 0$, we have $p = s \lhd \mathcal{S}$, where $s \in \mathsf{P}_{\mathsf{root}}$. Thus, by the definition of net refinement and $\Sigma = \Omega(\Delta[\Sigma])$, $p \in \mathsf{SP}^s_{\mathsf{new}}$ where $\mathsf{SP}^s_{\mathsf{new}}$ is as in the definition of net refinement. Since, from (a), we have ${}^\circ(\Delta[\Sigma]_v) \subseteq {}^\circ(\Delta[\Theta]_v)$ and $(\Delta[\Sigma]_v)^\circ \subseteq (\Delta[\Theta]_v)^\circ$ for every $v \in T_\Omega$, p is also a place in $\Omega(\Delta[\Theta]) = \Theta$, with the same label $\lambda_\Omega(s)$. The induction step for $k > 0$ is straightforward. If $p = t_1 \lhd \ldots t_k \lhd s \lhd \mathcal{S}$, then $p \in \mathsf{ST}^{t_1}_{\mathsf{new}}$ and the place $t_2 \lhd \ldots t_k \lhd s \lhd \mathcal{S}$ belongs either to some rec-box in Δ or Σ; hence, by $\Theta = \Omega(\Delta[\Theta])$ and the definition of net refinement and the induction hypothesis, we obtain that p also belongs to Θ, with the same i-label.

Hence $S_0 \subseteq S_1$ and $\lambda_0|_{S_0} = \lambda_1|_{S_1}$. Moreover, by (1), we have $T_0 = T_1$ and $\lambda_0|_{T_0} = \lambda_1|_{T_1}$. The fact that $W_0 = W_1|_{(S_0 \times T_0) \cup (T_0 \times S_0)}$ immediately results from the fact that in a refinement the arc weights are determined by the structure of name trees, and by the same kind of argument on the size of the transition names as in the proof of (1). □

Theorem 3. If Σ and Θ are solutions of (3) such that ${}^\circ\Sigma = {}^\circ\Theta$ and $\Sigma^\circ = \Theta^\circ$ then $\Sigma = \Theta$.
Proof: Follows from lemma 2(2) and $\sqsubseteq$ being a partial order. □

5.1 Using fixpoints to solve recursive equations

It easily follows from the definition of net refinement that the right-hand side of the recursive equation (3) induces a mapping $\mathbf{F} : \mathsf{Box}^{rec} \to \mathsf{Box}^{rec}$ defined, for every rec-box Σ, by

$$\mathbf{F}(\Sigma) \;=\; \Omega(\Delta[\Sigma]). \tag{4}$$

Hence solving equation (3) amounts to finding fixpoint(s) of $\mathbf{F}$, and we have already defined the inclusion order on labelled nets, $\sqsubseteq$, which will be suitable for the latter purpose. As we have seen, it is possible to form $\sqsubseteq$-chains and limits in the usual way. Moreover, every $\sqsubseteq$-chain of rec-boxes χ has a limit $\bigsqcup \chi$ and, by proposition 1, this limit is also a rec-box. The usual treatment of fixpoint theory would now proceed by establishing the monotonicity (and possibly continuity) of the mapping $\mathbf{F}$ defined by (4).

We say that $\mathbf{F}$ is $\sqsubseteq$-*monotonic* if $\Sigma \sqsubseteq \Theta$ implies $\mathbf{F}(\Sigma) \sqsubseteq \mathbf{F}(\Theta)$. Moreover, a $\sqsubseteq$-monotonic $\mathbf{F}$ is $\sqsubseteq$-*continuous* if for every $\sqsubseteq$-chain χ of rec-boxes, $\mathbf{F}(\bigsqcup \chi) \;=\; \bigsqcup\; \mathbf{F}(\chi)\;\ldots$. Notice that $\bigsqcup\; \mathbf{F}(\chi)$ is well defined as a monotonic $\mathbf{F}$ is also $\sqsubseteq$-chain-preserving and so $\mathbf{F}(\chi)$ is a $\sqsubseteq$-chain.

Proposition 4 $\mathbf{F}$ is $\sqsubseteq$-monotonic.
Proof: Follows directly from the definitions of net refinement and $\sqsubseteq$. □

Using proposition 4, a similar result does not hold for $\sqsubseteq$-continuity without making an additional assumption. We will say that (3) is X-*finitary* if, for every place s in Ω,

$$\left|\{v \in {}^\bullet s \cup s^\bullet \mid \Delta_v = X\}\right| \;<\; \infty.$$

Intuitively, this means that each place in the operator box has 'access' to only finitely many instances of X.

Proposition 5 $\mathbf{F}$ is $\sqsubseteq$-continuous if and only if (3) is X-finitary.
Proof: ($\Longrightarrow$) Suppose that (3) is not X-finitary. Then there is $s \in S_\Omega$ such that, without loss of generality, the set $V = \{v \in {}^\bullet s \mid \Delta_v = X\}$ is infinite. In particular, this means that V contains a countably infinite set of transitions $U = \{v_0, v_1, \ldots\}$. We now define a $\sqsubseteq$-chain $\chi = \Sigma^0\ \Sigma^1\ \ldots$ of rec-boxes such that, for every $k \geq 0$,

$$\Sigma^k \;=\; \Big(\{e, x_0, \ldots, x_k\}, \emptyset, \emptyset, \{e\} \times \{\mathsf{e}\} \cup \{x_0, \ldots, x_k\} \times \{\mathsf{x}\}, \emptyset\Big)$$

where $e, x_0, x_1, \ldots$ are distinct places (i.e., χ is a chain of unmarked ex-boxes. For every transition $v \in s^\bullet$, let e_v be a place in ${}^\circ(\Delta_v)$ if $\Delta_v \in \mathsf{Box}^{rec}$, and $e_v = e$ if $\Delta_v = X$. Moreover, for every transition $z \in Z = {}^\bullet s \setminus U$, let x_z be a place in $(\Delta_z)^\circ$ if $\Delta_z \in \mathsf{Box}^{rec}$, and $x_z = x_0$ if $\Delta_z = X$. Then the following place

$$p \;=\; s \lhd \Big(\{v \lhd e_v\}_{v \in s^\bullet} + \{w \lhd x_z\}_{z \in Z} + \{v_i \lhd x_i\}_{i \in \mathbf{N}}\Big)$$

belongs to the rec-box $\mathbf{F}(\bigsqcup \chi)$, but it does not belong to any $\mathbf{F}(\Sigma^i)$ and thus neither to $\bigsqcup \mathbf{F}(\chi)$ proving that $\mathbf{F}$ is not $\sqsubseteq$-continuous.

($\Longleftarrow$) Suppose that $\chi = \Sigma^0\ \Sigma^1\ \ldots$ is a $\sqsubseteq$-chain of rec-boxes. We need to show that $\Theta = \mathbf{F}(\Sigma)$ where $\Sigma = \bigsqcup \chi$ and $\Theta = \bigsqcup \mathbf{F}(\chi)$. Since $\Sigma^i \sqsubseteq \Sigma$ and $\mathbf{F}$ is $\sqsubseteq$-monotonic, $\mathbf{F}(\Sigma^i) \sqsubseteq \mathbf{F}(\Sigma)$ for every $i \geq 0$, and thus $\Theta \sqsubseteq \mathbf{F}(\Sigma)$. As a consequence, to prove $\Theta = \mathbf{F}(\Sigma)$ it suffices to show that $T_{\mathbf{F}(\Sigma)} \subseteq T_\Theta$ and $S_{\mathbf{F}(\Sigma)} \subseteq S_\Theta$.

The set of transitions of $\mathbf{F}(\Sigma)$ is included[7] in Θ. Indeed, if $(v, \alpha) \lhd R \in T_{\mathbf{F}(\Sigma)}$ and $\Delta_v = X$ then, since there are only finitely many transitions in R, there is $i \geq 0$ such that all transitions in R belong to Σ^i. Hence $(v, \alpha) \lhd R \in T_{\mathbf{F}(\Sigma^i)} \subseteq T_\Theta$.

To demonstrate that $S_{\mathbf{F}(\Sigma)} \subseteq S_\Theta$ we observe that, by (3) being X-finitary, to construct $p \in S_{\mathbf{F}(\Sigma)}$ we only need finitely many (possibly none) places which belong to Σ and therefore there is $i \geq 0$ such that all these places belong to Σ^i and hence $p \in S_{\mathbf{F}(\Sigma^i)} \subseteq S_\Theta$. □

Although the $\sqsubseteq$-continuity of $\mathbf{F}$ does not hold in the general case, there is a closely related and for our purposes sufficient, property of $\mathbf{F}$ obtained by strengthening the inclusion order. For two rec-boxes such that $\Sigma \sqsubseteq \Theta$, we shall denote $\Sigma \mathrel{\tilde{\sqsubseteq}} \Theta$ if ${}^\circ\Sigma = {}^\circ\Theta$ and $\Sigma^\circ = \Theta^\circ$. We then define $\tilde{\sqsubseteq}$-chains, $\tilde{\sqsubseteq}$-chain preservation, $\tilde{\sqsubseteq}$-monotonicity and $\tilde{\sqsubseteq}$-continuity in the same way as the corresponding notions for $\sqsubseteq$. Notice that since $\tilde{\sqsubseteq}$-chains are also $\sqsubseteq$-chains, proposition 1 and other properties enjoyed by the latter hold for $\tilde{\sqsubseteq}$-chains as well.

Notice that right from the beginning we could have only considered the stronger inclusion order, $\tilde{\sqsubseteq}$, which would have simplified the presentation. However, in section 6, we will need $\sqsubseteq$ rather than $\tilde{\sqsubseteq}$ to discuss solutions to (3) obtained as limits of finite approximations. Also, observe that neither $(\mathsf{Box}^{rec}, \sqsubseteq)$ nor $(\mathsf{Box}^{rec}, \tilde{\sqsubseteq})$ generate lattices, since they may contain minimal elements (ex-boxes) which are incomparable. Hence fixpoint theory needs to be applied with care.

[7] Notice that this property is true in all generality, i.e., it does not rely on the recursive equation being X-finitary.

Theorem 6. $\mathbf{F}$ is $\tilde{\sqsubseteq}$-monotonic and $\tilde{\sqsubseteq}$-continuous.
Proof: Suppose that $\Sigma \tilde{\sqsubseteq} \Theta$. Then, by the $\sqsubseteq$-monotonicity of $\mathbf{F}$, $\mathbf{F}(\Sigma) \sqsubseteq \mathbf{F}(\Theta)$. As to the equality of the entry and exit places, we observe that from ${}^\circ\Sigma = {}^\circ\Theta$ and $\Sigma^\circ = \Theta^\circ$ it follows that for every $s \in S_\Omega$, $\mathsf{SP}^s_{\mathsf{new}}$ in $\mathbf{F}(\Sigma)$ is equal to $\mathsf{SP}^s_{\mathsf{new}}$ in $\mathbf{F}(\Theta)$, where each $\mathsf{SP}^s_{\mathsf{new}}$ is the auxiliary set of places used in the definition of net refinement. Hence $\mathbf{F}$ is $\tilde{\sqsubseteq}$-monotonic, since

$$ {}^\circ\mathbf{F}(\Sigma) \;=\; \bigcup_{s \in {}^\circ\Omega} \mathsf{SP}^s_{\mathsf{new}} \;=\; {}^\circ\mathbf{F}(\Theta) \quad \text{and} \quad \mathbf{F}(\Sigma)^\circ \;=\; \bigcup_{s \in \Omega^\circ} \mathsf{SP}^s_{\mathsf{new}} \;=\; \mathbf{F}(\Theta)^\circ. $$

Suppose now that $\chi = \Sigma^0\ \Sigma^1\ \ldots$ is a $\tilde{\sqsubseteq}$-chain of rec-boxes. Moreover, let Σ and Θ be as in the second part of the proof of proposition 5 whose careful inspection leads to a conclusion that it can be re-used with one exception, namely we have to re-prove that $S_{\mathbf{F}(\Sigma)} \subseteq S_\Theta$. This can be done in the following way.

By the equality of the sets of entry and exit places in the chain, we have that ${}^\circ\Sigma = {}^\circ(\Sigma^0)$ and $\Sigma^\circ = (\Sigma^0)^\circ$. Hence, for every $s \in S_\Omega$, $\mathsf{SP}^s_{\mathsf{new}}$ in $\mathbf{F}(\Sigma)$ is equal to $\mathsf{SP}^s_{\mathsf{new}}$ in $\mathbf{F}(\Sigma^0)$. Moreover, if $v \lhd q \in \mathsf{ST}^v_{\mathsf{new}}$ in $\mathbf{F}(\Sigma)$ and $\Delta_v = X$ then there is $i \geq 0$ such that q belongs to Σ^i and hence $v \lhd q \in \mathsf{ST}^v_{\mathsf{new}}$ in $\mathbf{F}(\Sigma^i)$; and if $\Delta_v \in \mathsf{Box}^{rec}$, $v \lhd q \in \mathsf{ST}^v_{\mathsf{new}}$ in $\mathbf{F}(\Sigma^0)$. Thus $S_{\mathbf{F}(\Sigma)} \subseteq S_\Theta$. □

Since the recursive equation induces a $\tilde{\sqsubseteq}$-continuous mapping $\mathbf{F}$, we can attempt to solve it by constructing a $\tilde{\sqsubseteq}$-chain of rec-boxes formed through successive applications of $\mathbf{F}$ to some initial approximation Υ of a solution.

For a rec-box Υ, we denote by $\mathbf{F}^*(\Upsilon)$ the sequence

$$ \mathbf{F}^0(\Upsilon)\ \mathbf{F}^1(\Upsilon)\ \mathbf{F}^2(\Upsilon)\ \ldots $$

where $\mathbf{F}^0(\Upsilon) = \Upsilon$ and $\mathbf{F}^n(\Upsilon) = \mathbf{F}(\mathbf{F}^{n-1}(\Upsilon))$, for every $n \geq 1$. We will call Υ a $\tilde{\sqsubseteq}$*-seed* for the equation (3) if $\Upsilon \tilde{\sqsubseteq} \mathbf{F}(\Upsilon)$; in such a case, by the $\tilde{\sqsubseteq}$-monotonicity of $\mathbf{F}$, we have that $\mathbf{F}^*(\Upsilon)$ is a $\tilde{\sqsubseteq}$-chain of rec-boxes. And such a chain always yields a solution.

Theorem 7. If Υ is a $\tilde{\sqsubseteq}$-seed then $\bigsqcup \mathbf{F}^*(\Upsilon)$ is a solution of (3).
Proof: By theorem 6, $\mathbf{F}(\bigsqcup \mathbf{F}^*(\Upsilon)) = \bigsqcup\ \mathbf{F}^1(\Upsilon)\ \mathbf{F}^2(\Upsilon)\ \mathbf{F}^3(\Upsilon)\ \ldots$. Moreover, by $\Upsilon \tilde{\sqsubseteq} \mathbf{F}(\Upsilon)$,

$$ \bigsqcup\ \mathbf{F}^1(\Upsilon)\ \mathbf{F}^2(\Upsilon)\ \mathbf{F}^3(\Upsilon)\ \ldots = \bigsqcup\ \Upsilon\ \mathbf{F}^1(\Upsilon)\ \mathbf{F}^2(\Upsilon)\ \mathbf{F}^3(\Upsilon)\ \ldots = \bigsqcup\ \mathbf{F}^*(\Upsilon). $$

Hence $\bigsqcup \mathbf{F}^*(\Upsilon)$ is a fixpoint of $\mathbf{F}$ and so a solution of (3). □

Moreover, it is sufficient to consider ex-boxes as the only seeds.

Theorem 8. Let Σ be a solution of (3) and

$$ \Upsilon \;=\; \Big({}^\circ\Sigma \cup \Sigma^\circ\ ,\ \emptyset\ ,\ \emptyset\ ,\ ({}^\circ\Sigma \times \{\mathsf{e}\}) \cup (\Sigma^\circ \times \{\mathsf{x}\})\ ,\ \emptyset \Big). $$

Then Υ is a $\tilde{\sqsubseteq}$-seed for (3) such that $\bigsqcup \mathbf{F}^*(\Upsilon) = \Sigma$.
Proof: We first observe that Υ is an unmarked ex-box and so also a rec-box. Moreover, by $\Sigma = \mathbf{F}(\Sigma)$ and the definition of net refinement, it is the case that ${}^\circ\Upsilon = {}^\circ\Sigma = {}^\circ\mathbf{F}(\Upsilon)$ and $\Upsilon^\circ = \Sigma^\circ = \mathbf{F}(\Upsilon)^\circ$. Hence $\Upsilon \tilde{\sqsubseteq} \mathbf{F}(\Upsilon)$ and so Υ is a $\tilde{\sqsubseteq}$-seed. Furthermore, ${}^\circ(\bigsqcup \mathbf{F}^*(\Upsilon)) = {}^\circ\Sigma$ and $(\bigsqcup \mathbf{F}^*(\Upsilon))^\circ = \Sigma^\circ$. Hence, by theorems 3 and 7, $\bigsqcup \mathbf{F}^*(\Upsilon) = \Sigma$. □

What we still lack to complete the discussion of the recursive equation (3) is a theorem saying that it always has a solution. This result will be postponed until section 5.4, after we have discussed in more detail seed boxes. First, however, we discuss the form of place and transition trees in nets solving (3) and then present an example.

5.2 Places and transitions in net solutions

A closer examination of the proof of lemma 2(1) indicates that the uniqueness of the transition set in a solution of (3) is essentially due to the fact that we only allow *finite* transition trees. If we allowed infinite ones, then transition sets could be different for different solutions. The uniqueness result may also be derived using the following argument.

Let τ be a transition or place name in $\mathsf{T_{tree}} \cup \mathsf{P_{tree}}$ and ξ be one of its nodes. The *level* of ξ is the length of the path from the root of τ to ξ defined as the number of edges on that path; in particular, the level of the root is 0. For a finite τ, its *depth*, denoted by $\mathsf{depth}(\tau)$, is the maximal level of the nodes.

Proposition 9 Let Υ and Υ' be $\tilde{\sqsubseteq}$-seeds for (3). Then, for every $n \geq 0$ and all $k, l \geq n$,

$$\{t \in T_{\mathbf{F}^k(\Upsilon)} \mid \mathsf{depth}(t) \leq n\} \;=\; \{t \in T_{\mathbf{F}^l(\Upsilon')} \mid \mathsf{depth}(t) \leq n\}.$$

Proof: Let $T_{nk} = \{t \in T_{\mathbf{F}^k(\Upsilon)} \mid \mathsf{depth}(t) \leq n\}$ and $T'_{nl} = \{t \in T_{\mathbf{F}^l(\Upsilon')} \mid \mathsf{depth}(t) \leq n\}$, for all $k, l \geq n \geq 0$.

We proceed by induction on n. In the base case $(n = 0)$ we have $T_{0k} = \emptyset = T'_{0l}$ which follows from the definition of net refinement (which never yields a transition of zero depth) and the following facts: $\mathbf{F}^0(\Upsilon) = \Upsilon \,\tilde{\sqsubseteq}\, \mathbf{F}(\Upsilon)$ and $\mathbf{F}^0(\Upsilon') = \Upsilon' \,\tilde{\sqsubseteq}\, \mathbf{F}(\Upsilon')$ and $\mathbf{F}^k(\Upsilon), \mathbf{F}^l(\Upsilon') \in \mathbf{F}(\mathsf{Box}^{rec})$, for all $k, l \geq 1$.

Suppose that $T_{nk} = T'_{nl}$ for some $n \geq 0$ and all $k, l \geq n$. Let $k, l \geq n{+}1$ and $u = (v, \alpha) \lhd R \in T_{(n+1)k}$. If $\Delta_v \neq X$ then $u \in T'_{(n+1)l}$ since $R \in \mathsf{mult}(T_{\Delta_v})$ and $\mathbf{F}^l(\Upsilon') \in \mathbf{F}(\mathsf{Box}^{rec})$. If $\Delta_v = X$ then $u \in T'_{(n+1)l}$ since $R \in \mathsf{mult}(T_{n(k-1)})$ (and so, by the induction hypothesis, $R \in \mathsf{mult}(T'_{n(l-1)})$) and $\mathbf{F}^l(\Upsilon') = \mathbf{F}(\mathbf{F}^{l-1}(\Upsilon'))$. Hence $T_{(n+1)k} \subseteq T'_{(n+1)l}$. By symmetry, $T'_{(n+1)l} \subseteq T_{(n+1)k}$ and so $T_{(n+1)k} = T'_{(n+1)l}$. □

Corollary 10 Let Σ and Θ be solutions of (3). Then, for every $n \geq 0$,

$$\{t \in T_\Sigma \mid \mathsf{depth}(t) \leq n\} \;=\; \{t \in T_\Theta \mid \mathsf{depth}(t) \leq n\}.$$

Proof: Follows immediately from proposition 9 after setting $\Upsilon = \Sigma$ and $\Upsilon' = \Theta$. □

The last result does not rely on the fact that transition trees are finite; hence if we allowed infinite transition trees, then every solution of (3) would contain all (finite) transition trees generated from the ex-box appearing in theorem 8. As a consequence, the solutions so constructed are *minimal* in terms of transition sets, and this minimality is sealed by the finiteness of the allowed transition trees.

It is not possible to obtain a similar result for places because restricting place trees to finite ones could lead to empty nets, which is not allowed by the ex-restrictedness of boxes (whereas the transition sets may be empty, see section 7). Thus one cannot be certain that there will be a unique minimal solution in terms of places. But it is possible to obtain a result similar to proposition 9, after allowing for the presence of infinite trees. Below, for every $n \geq 0$, the *n-prefix* of a (possibly infinite) $\tau \in \mathsf{P_{tree}}$ is the tree $\mathsf{pref}_n(\tau)$ obtained by deleting all nodes whose level is greater than n; in particular, $\mathsf{pref}_0(\tau)$ consists just of the root of τ.

Proposition 11 Let Υ and Υ' be $\tilde{\sqsubseteq}$-seeds for (3). Then

$$\mathsf{pref}_n(S_{\mathbf{F}^k(\Upsilon)}) = \mathsf{pref}_n(S_{\mathbf{F}^l(\Upsilon')}),$$

for every $n \geq 0$ and all $k, l \geq n+2$.

Proof: Let $S_k = S_{\mathbf{F}^k(\Upsilon)}$ and $S'_l = S_{\mathbf{F}^l(\Upsilon')}$, for all $k, l \geq 0$. In general, each place tree has either the form $s \lhd \mathcal{S}$ or $t_1 \lhd \ldots t_k \lhd s \lhd \mathcal{S}$, where $t_1, \ldots, t_k \in \mathsf{T_{root}}$ and $k \geq 1$ and $s \in \mathsf{P_{root}}$. Moreover, no non-empty prefix of a place of one form is a prefix of a place of the other form. As a result, we can split the proof in two parts. Below, S^I denotes the set of trees of the first form, and S^{II} of the second form.

Part 1: $\mathsf{pref}_n(S_k \cap S^I) = \mathsf{pref}_n(S'_l \cap S^I)$, for every $n \geq 0$ and all $k, l \geq \lfloor \frac{n}{2} \rfloor + 1$.
We proceed by induction on n. In the base case $(n \in \{0, 1\})$ we have $\mathsf{pref}_0(S_k \cap S^I) = S_\Omega = \mathsf{pref}_0(S'_l \cap S^I)$ and $\mathsf{pref}_1(S_k \cap S^I) = \{s \lhd ({}^\bullet s + s^\bullet) \mid s \in S_\Omega\} = \mathsf{pref}_1(S'_l \cap S^I)$, for all $k, l \geq 1$, which follows from the definition of net refinement and $\mathbf{F}^k(\Upsilon), \mathbf{F}^l(\Upsilon') \in \mathbf{F}(\mathsf{Box}^{rec})$.

Suppose that $\mathsf{pref}_{2n+1}(S_k \cap S^I) = \mathsf{pref}_{2n+1}(S'_l \cap S^I)$, for some $n \geq 0$ and all $k, l \geq n+1$. We will show that $\mathsf{pref}_{2n+3}(S_k \cap S^I) \subseteq \mathsf{pref}_{2n+3}(S'_l \cap S^I)$, for all $k, l \geq n+2$.

Let $k, l \geq n+2$ and $p = s \lhd \mathcal{S} \in S_k$. We have

$$p \;=\; s \lhd \Big(\{v \lhd p_v\}_{v \in {}^\bullet s \cap V} + \{v \lhd q_v\}_{v \in {}^\bullet s \cap U} + \{w \lhd r_w\}_{w \in s^\bullet \cap V} + \{w \lhd z_w\}_{w \in s^\bullet \cap U} \Big)$$

where $V = \{v \in T_\Omega \mid \Delta_v = X\}$ and $U = T_\Omega \backslash V$ and the following are satisfied: $p_v \in \mathbf{F}^{k-1}(\Upsilon)^\circ \subseteq S^I$, $r_w \in {}^\circ\mathbf{F}^{k-1}(\Upsilon) \subseteq S^I$, $q_v \in \Delta_v^\circ$, and $z_w \in {}^\circ\Delta_w$. Thus

$$\mathsf{pref}_{2n+3}(p) \;=\; s \lhd \begin{pmatrix} \{v \lhd \mathsf{pref}_{2n+1}(p_v)\}_{v \in {}^\bullet s \cap V} & + \{v \lhd q_v\}_{v \in {}^\bullet s \cap U} & + \\ \{w \lhd \mathsf{pref}_{2n+1}(r_w)\}_{w \in s^\bullet \cap V} & + \{w \lhd z_w\}_{w \in s^\bullet \cap U} & \end{pmatrix}.$$

By the induction hypothesis, for every $v \in {}^\bullet s \cap V$ there is $p'_v \in S'_{l-1} \cap S^I$ such that $\mathsf{pref}_{2n+1}(p_v) = \mathsf{pref}_{2n+1}(p'_v)$; similarly, for every $w \in s^\bullet \cap V$ there is $r'_w \in S'_{l-1} \cap S^I$ such that $\mathsf{pref}_{2n+1}(r_w) = \mathsf{pref}_{2n+1}(r'_w)$. Then the following place belongs to $S'_l \cap S^I$:

$$p' \;=\; s \lhd \Big(\{v \lhd p'_v\}_{v \in {}^\bullet s \cap V} + \{v \lhd q_v\}_{v \in {}^\bullet s \cap U} + \{w \lhd r'_w\}_{w \in s^\bullet \cap V} + \{w \lhd z_w\}_{w \in s^\bullet \cap U} \Big).$$

And $\mathsf{pref}_{2n+3}(p') = \mathsf{pref}_{2n+3}(p)$. Hence $\mathsf{pref}_{2n+3}(S_k \cap S^I) \subseteq \mathsf{pref}_{2n+3}(S'_l \cap S^I)$ and, since the reverse inclusion holds by symmetry, we have $\mathsf{pref}_{2n+3}(S_k \cap S^I) = \mathsf{pref}_{2n+3}(S'_l \cap S^I)$.

We have shown the result for all odd values of n. Clearly, the result holds then also for all even n.

Part 2: $\mathsf{pref}_n(S_k \cap S^{II}) = \mathsf{pref}_n(S'_l \cap S^{II})$, for every $n \geq 0$ and all $k, l \geq n+2$.
We first observe that if no net occurring in (3) has an internal place then the same is true of $\Psi = \bigsqcup \mathbf{F}^*(\Upsilon)$ (as well as $\Psi = \bigsqcup \mathbf{F}^*(\Upsilon')$). Indeed, since Ψ is a fixpoint of $\mathbf{F}$, any internal place is: (i) either $s \lhd \mathcal{S}$ in S^I with $s \in \ddot{\Omega}$, but $\ddot{\Omega}$ is empty; or (ii) $t_1 \lhd \ldots t_k \lhd s$ in S^{II} with $s \in \ddot{\Delta}_v$, for some $v \in T_\Omega$, but $\ddot{\Delta}_v$ is empty too; or (iii) $t_1 \lhd \ldots t_k \lhd s \lhd \mathcal{S}$ in S^{II} with $s \in \ddot{\Psi} \cap S^I$, but we have already argued in (i) that $\ddot{\Psi} \cap S^I$ is empty (the last two facts result from a straightforward induction on the number of t_i's). Hence $S_k \cap S^{II} = \emptyset = S'_l \cap S^{II}$, for all $k, l \geq 0$.

Otherwise, we proceed by induction on n. In the base case ($n = 0$) we have

$$\mathsf{pref}_0(S_k \cap S^{II}) \;=\; T_\Omega \backslash \{v \mid \Delta_v \neq X \;\wedge\; \ddot{\Delta}_v = \emptyset\} \;=\; \mathsf{pref}_0(S'_l \cap S^{II}),$$

for all $k, l \geq 2$. Indeed, if $\Delta_v \neq X$ and $\ddot{\Delta}_v = \emptyset$, then there is no internal place of the kind $v \lhd q$ in any $\mathbf{F}(\Psi)$, whatever the rec-box Ψ; in particular, not in $\mathbf{F}^k(\Upsilon)$ nor in $\mathbf{F}^l(\Upsilon')$ for $k, l \geq 1$. If $\Delta_v \neq X$ and $q \in \ddot{\Delta}_v \neq \emptyset$, then the internal place $v \lhd q$ does occur in every $\mathbf{F}(\Psi)$, whatever the rec-box Ψ; in particular, in $\mathbf{F}^k(\Upsilon)$ and $\mathbf{F}^l(\Upsilon')$ for $k, l \geq 1$. If $\Delta_v = X$ and $s \in \ddot{\Omega}$, then there is some internal place $s \lhd \mathcal{S}$ in $\mathbf{F}(\Upsilon)$, an internal place $v \lhd s \lhd \mathcal{S}$ in $\mathbf{F}^2(\Upsilon)$, and an internal place $v \lhd v \lhd v \lhd \cdots \lhd s \lhd \mathcal{S}$ in each $\mathbf{F}^k(\Upsilon)$ for $k \geq 2$, and similarly for $\mathbf{F}^l(\Upsilon')$, where $l \geq 2$. Finally, if $\Delta_v = X$, $\Delta_w \neq X$ and $q \in \ddot{\Delta}_w \neq \emptyset$ (for some w), then $w \lhd q$ does occur in every $\mathbf{F}(\Psi)$ and $v \lhd w \lhd q$ does occur in each $\mathbf{F}^k(\Upsilon)$ for $k \geq 2$; and similarly for $\mathbf{F}^l(\Upsilon')$, where $l \geq 2$.[8]

The inductive step results from Part 1, the induction hypothesis, and the fact that

$$\mathsf{pref}_n(S_k \cap S^{II}) \;=\; \{v \lhd q \mid \Delta_v \neq X \;\wedge\; q \in \ddot{\Delta}_v\} \cup \{v \lhd q \mid \Delta_v = X \;\wedge\; q \in \mathsf{pref}_{n-1}(\ddot{\Psi})\},$$

where $\Psi = \mathbf{F}^{k-1}(\Upsilon)$, for $n \geq 1$ and $k \geq n+2$ (and similarly for $\mathsf{pref}_n(S'_l \cap S^{II})$). □

Corollary 12 Let Σ and Θ be solutions of (3). Then $\mathsf{pref}_n(S_\Sigma) = \mathsf{pref}_n(S_\Theta)$, for every $n \geq 0$.
Proof: Follows immediately from proposition 11 after setting $\Upsilon = \Sigma$ and $\Upsilon' = \Theta$. □

In other words, all solutions have the same finite prefixes of place trees, and the n-prefixes are all present in the $(n+2)^{\text{nd}}$ iteration of the limit construction, whatever the seed.

It is possible to derive new places and transitions of a solution of (3) as *suffixes* of known ones, through a repeated application of the next result.

[8] Notice that $\mathsf{pref}_0(S_1 \cap S^{II}) = \{v \mid \Delta_v \neq X \;\wedge\; \ddot{\Delta}_v \neq \emptyset\}$ if $\ddot{\Upsilon} = \emptyset$, and $\mathsf{pref}_0(S'_1 \cap S^{II}) = \{v \mid \Delta_v \neq X \;\wedge\; \ddot{\Delta}_v \neq \emptyset\} \cup \{v \mid \Delta_v = X\}$ if $\ddot{\Upsilon}' \neq \emptyset$, so that it may happen that $\mathsf{pref}_0(S_1 \cap S^{II}) \neq \mathsf{pref}_0(S'_1 \cap S^{II})$.

Proposition 13 Let Σ be a solution of (3), and v be a transition in Ω such that $\Delta_v = X$.

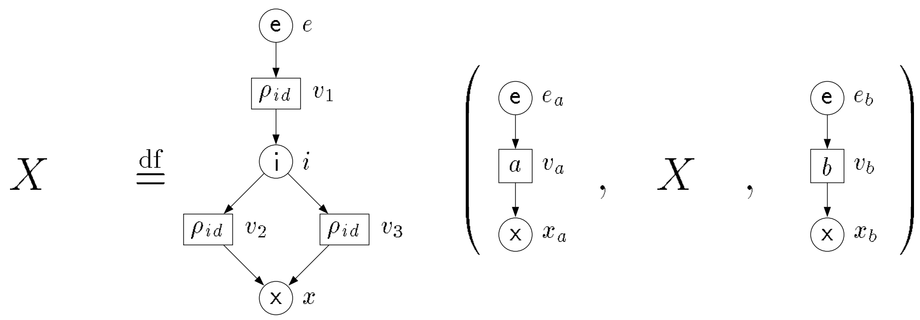
(1) If $(v, \alpha) \lhd (\{u\} + R)$ is a transition in Σ then u is also a transition in Σ.
(2) If $v \lhd q$ is a place in Σ then q is an internal place in Σ.
(3) If $s \lhd (\{v \lhd q\} + \mathcal{S})$ is a place in Σ then q is an entry or exit place in Σ.

Proof: Follows immediately from $\Sigma = \mathbf{F}(\Sigma)$ and the definition of net refinement. $\square$

And, conversely, by applying $\mathbf{F}$, or more generally $\mathbf{F}^n$, it may be possible to construct new places and transitions from known ones by suitably prefixing them. This may be useful to determine a minimal solution containing some place, as will be shown in section 6.1.

5.3 An example of the limit construction

Consider the following equation which intuitively corresponds to $X \stackrel{\mathrm{df}}{=} a; (X \,\square\, b)$:



(5)

Let us examine how a solution of the net equation (5) might be constructed using the limit construction. Suppose that there exists already an approximation Υ^0 given as a starting point, a hypothetical seed box. Then inserting that approximation in the place of the X on the right-hand side of (5) yields a new box $\Upsilon^1 = \Omega(\mathsf{N}_a, \Upsilon^0, \mathsf{N}_b)$, where Ω is the operator box in this equation, which is a slightly better approximation.

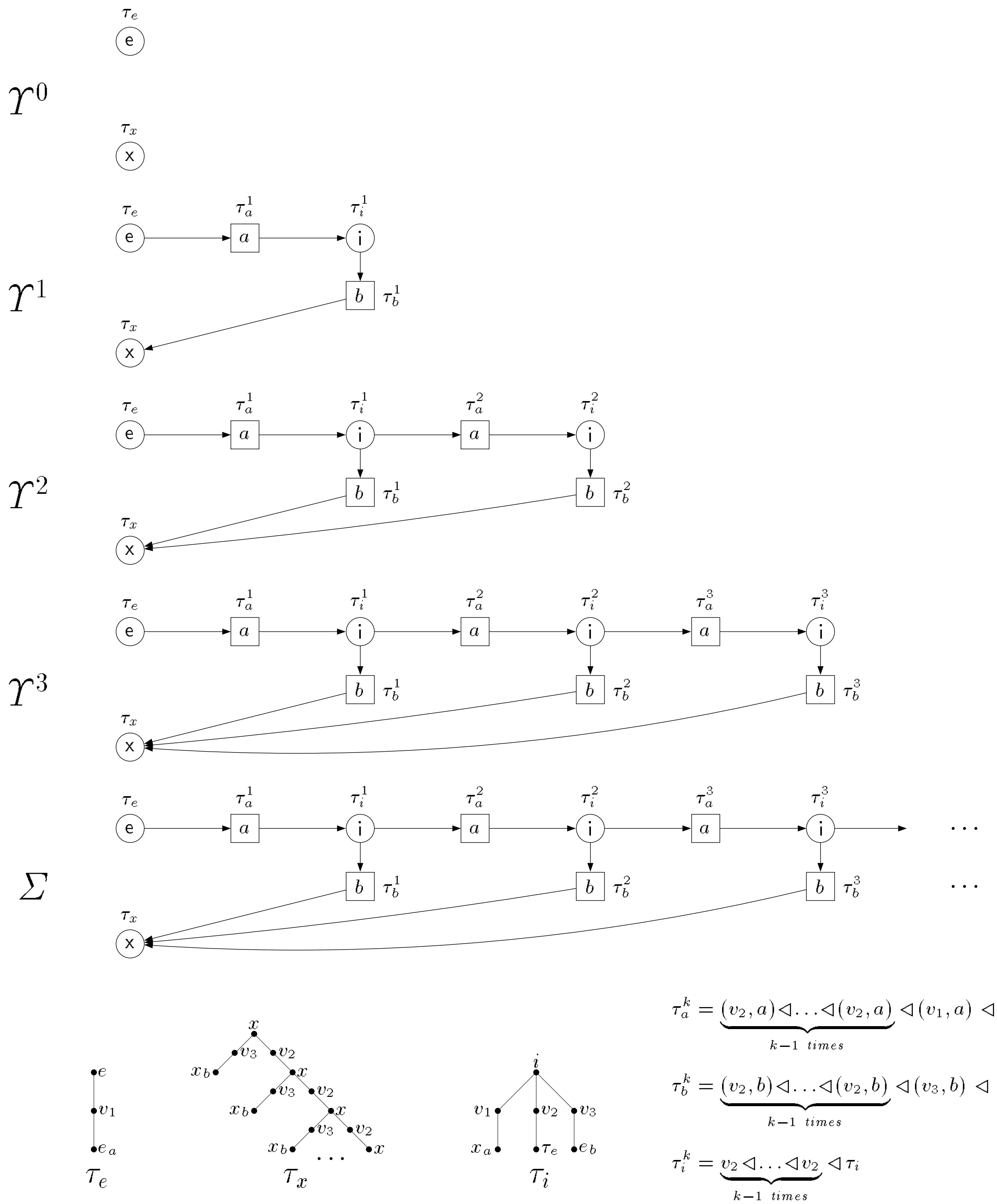
We then construct another approximation $\Upsilon^2 = \Omega(\mathsf{N}_a, \Upsilon^1, \mathsf{N}_b)$ by the same principle, and so on. We already know that if Υ^0 is a $\tilde{\sqsubseteq}$-seed then the limit of the sequence $\Upsilon^0\Upsilon^1\ldots$ so constructed exists and solves the equation (5). We now claim that the ex-box Υ^0 shown in figure 4 (with these particular place names) is a good seed for (5). Figure 4 shows also the first three approximations resulting from this seed and, finally, it shows $\Sigma = \bigsqcup \mathbf{F}^*(\Upsilon^0)$, which is the limit of the infinite series of boxes whose first three elements are depicted. The reader may check that this box satisfies (5) in the strictest possible sense of the word. To see this, replace the variable X on the right-hand side of the equation by the box Σ shown in figure 4, perform net refinement $\Omega(\mathsf{N}_a, \Sigma, \mathsf{N}_b)$, and check that the result is not just isomorphic but even *identical* to Σ, i.e., it has the same places and the same transitions with the same connections and the same labels.

Thus, once an appropriate starting point Υ^0 has been chosen, the approximation of a solution is straightforward (we just keep refining successive approximations into the right-hand side of the equation under consideration and compute the limit) and leads to a solution in all cases.

5.4 Deriving seed boxes

To find a $\tilde{\sqsubseteq}$-seed Υ for the recursive equation (3), we need to ensure that $\Upsilon \sqsubseteq \mathbf{F}(\Upsilon)$, ${}^\circ\Upsilon = {}^\circ\mathbf{F}(\Upsilon)$ and $\Upsilon^\circ = \mathbf{F}(\Upsilon)^\circ$ hold for Υ and the next approximation $\mathbf{F}(\Upsilon)$. One might ask why the rec-box Υ^0 was a suitable starting point for the equation (5). First, it is a good idea to start always with an ex-box, because ex-boxes are minimal with respect to the number of transitions they contain and because, as a consequence of theorem 8, all solutions can be so obtained. Moreover, one only needs to ensure that ${}^\circ\Upsilon = {}^\circ\mathbf{F}(\Upsilon)$ and $\Upsilon^\circ = \mathbf{F}(\Upsilon)^\circ$, since then the condition $\Upsilon \sqsubseteq \mathbf{F}(\Upsilon)$ is automatically fulfilled.[9] We will come

[9] It is even better (if possible) to start with an ex-box containing only one entry and one exit place, because such boxes are minimal with respect to the ordering $\sqsubseteq$.



$$\tau_a^k = \underbrace{(v_2,a)\lhd\ldots\lhd(v_2,a)}_{k-1\ times}\lhd(v_1,a)\ \lhd\ t_a$$

$$\tau_b^k = \underbrace{(v_2,b)\lhd\ldots\lhd(v_2,b)}_{k-1\ times}\lhd(v_3,b)\ \lhd\ t_b$$

$$\tau_i^k = \underbrace{v_2\lhd\ldots\lhd v_2}_{k-1\ times}\lhd\tau_i$$

Fig. 4. Four approximations for equation (5) and the resulting solution.

back to this point in section 6. Let us now fix a hypothetical starting ex-box

$$\Upsilon \;=\; \Big(\mathcal{P} \cup \mathcal{R}\ ,\ \emptyset\ ,\ \emptyset\ ,\ (\mathcal{P} \times \{\mathsf{e}\}) \cup (\mathcal{R} \times \{\mathsf{x}\})\ ,\ \emptyset \Big)$$

and develop conditions for it to be a $\tilde{\sqsubseteq}$-seed. First, because of ex-restrictedness, $\mathcal{P}$ and $\mathcal{R}$ should both be non-empty. They should also be disjoint, otherwise the labelling function would be ill-defined. Finally, with a view to ensure that ${}^\circ\Upsilon = {}^\circ\mathbf{F}(\Upsilon)$ and $\Upsilon^\circ = \mathbf{F}(\Upsilon)^\circ$, some conditions need to be imposed on $\mathcal{P}$ and $\mathcal{R}$, essentially by performing the refinement at the entry and exit places. As both place sets are sets of trees, these conditions can be translated into equalities over sets of place trees.

We define a mapping $\mathbf{f} \;:\; 2^{\mathsf{P}_{\mathsf{tree}}} \times 2^{\mathsf{P}_{\mathsf{tree}}} \;\to\; 2^{\mathsf{P}_{\mathsf{tree}}} \times 2^{\mathsf{P}_{\mathsf{tree}}}$ associated with $\mathbf{F}$, and through $\mathbf{F}$ with the recursive equation (3), so that for all $P, R \subseteq \mathsf{P}_{\mathsf{tree}}$,

$$\mathbf{f}(P,R) \;=\; \Big(\bigcup_{s \in {}^\circ\Omega} \mathbf{f}_s(P,R) \;\;,\;\; \bigcup_{s \in \Omega^\circ} \mathbf{f}_s(P,R) \Big) \tag{6}$$

where each $\mathbf{f}_s(P,R)$ comprises all place trees $s \lhd (\{v \lhd p_v\}_{v \in s^\bullet} + \{w \lhd r_w\}_{w \in {}^\bullet s})$ such that:

$$p_v \;\in\; \begin{cases} P & \text{if} \quad \Delta_v = X \\ {}^\circ\Delta_v & \text{if} \quad \Delta_v \neq X \end{cases} \qquad \text{and} \qquad r_w \;\in\; \begin{cases} R & \text{if} \quad \Delta_w = X \\ \Delta_w^\circ & \text{if} \quad \Delta_w \neq X. \end{cases}$$

Notice that if s is an isolated place then $\mathbf{f}_s(P,R) = \{s\}$, and if $P = \emptyset$ and $\Delta_v = X$ for some v (or $R = \emptyset$ and $\Delta_w = X$ for some w) then $\mathbf{f}_s(P,R) = \emptyset$.

A pair of sets $(P,R) \in 2^{\mathsf{P}_{\mathsf{tree}}} \times 2^{\mathsf{P}_{\mathsf{tree}}}$ is a *fixpoint* of $\mathbf{f}$ whenever $\mathbf{f}(P,R) = (P,R)$. A fixpoint (P,R) is *positive* if $P \neq \emptyset \neq R$; moreover (P,R) is *min/pos* if it is minimal among all positive fixpoints.[10] It is a matter of simple check to verify that the function $\mathbf{f}$ can be used to calculate the entry and exit places of $\mathbf{F}(\Upsilon)$:

$$\mathbf{f}(\mathcal{P}\ ,\ \mathcal{R}) \;=\; ({}^\circ\mathbf{F}(\Upsilon)\ ,\ \mathbf{F}(\Upsilon)^\circ). \tag{7}$$

As a result,

Lemma 14 The following are equivalent:

(1) Υ is a $\tilde{\sqsubseteq}$-seed.
(2) $(\mathcal{P},\mathcal{R})$ is a positive fixpoint of $\mathbf{f}$.

Proof: If Υ is a $\tilde{\sqsubseteq}$-seed then $\emptyset \neq \mathcal{P} = {}^\circ\Upsilon = {}^\circ\mathbf{F}(\Upsilon)$ and $\emptyset \neq \mathcal{R} = \Upsilon^\circ = \mathbf{F}(\Upsilon)^\circ$. This and (7) means that $(\mathcal{P},\mathcal{R})$ is a positive fixpoint of $\mathbf{f}$. The reverse implication follows directly from (7), $\mathcal{P} \neq \emptyset \neq \mathcal{R}$, $\mathcal{P} \cap \mathcal{R} = \emptyset$ (which in turn follows from ${}^\circ\Omega \cap \Omega^\circ = \emptyset$) and the definition of net refinement. □

Thus we have reduced the problem of finding a $\tilde{\sqsubseteq}$-seed for (3) to that of finding a positive fixpoint of $\mathbf{f}$. Using the notation introduced in (6), the latter amounts to finding non-empty sets $\mathcal{P}$ and $\mathcal{R}$ solving the following *tree equations*:

$$\mathcal{P} = \textstyle\bigcup_{s \in {}^\circ\Omega} \;\; \mathbf{f}_s(\mathcal{P},\mathcal{R})$$
$$\mathcal{R} = \textstyle\bigcup_{s \in \Omega^\circ} \;\; \mathbf{f}_s(\mathcal{P},\mathcal{R}).$$

Let us examine the function $\mathbf{f}$ and its fixpoints more closely. A first observation is that $(2^{\mathsf{P}_{\mathsf{tree}}} \times 2^{\mathsf{P}_{\mathsf{tree}}}, \subseteq)$ is a complete lattice with set union as the 'join' and intersection as the 'meet' operation, where all operations, including $\subseteq$, are carried out component-wise. Moreover, $\mathbf{f}$ is a monotonic mapping on this domain.

Lemma 15 $\mathbf{f}$ is monotonic.
Proof: Follows directly from the definition of $\mathbf{f}$. □

It therefore follows that Tarski's theorem ([14]) can be applied.

[10] That is, for every positive fixpoint (P',R') such that $P' \subseteq P$ and $R' \subseteq R$, it is the case that $(P',R') = (P,R)$.

Theorem 16. The set of fixpoints of $\mathbf{f}$ is a complete lattice whose maximal element is positive.
Proof: That the set of fixpoints of $\mathbf{f}$ is a complete lattice follows from [14] and lemma 15. To show the second part, it suffices to demonstrate that there are two trees in $\mathsf{P}_{\mathsf{tree}}$, τ_e and τ_x, such that $(\{\tau_e\},\{\tau_x\}) \subseteq \mathbf{f}(\{\tau_e\},\{\tau_x\})$. The following construction achieves the desired effect.

For every $v \in T_\Omega$ such that $\Delta_v \neq X$, take any basic names $e_v \in {}^\circ\Delta_v$ and $x_v \in \Delta_v^\circ$ (this is possible since rec-boxes are ex-restricted and $S_{\Delta_v} \subseteq \mathsf{P}_{\mathsf{root}}$). Moreover, for every $v \in T_\Omega$ such that $\Delta_v = X$, let $e_v = s$ and $x_v = r$, where $s \in {}^\circ\Omega$ and $r \in \Omega^\circ$ are arbitrary but fixed places (this is possible because Ω is ex-restricted). The nodes of τ_e and τ_x are all strings of the following form (together with their prefixes):

$$q_0\,(v_0,z_0)\,q_1\,(v_1,z_1)\,\ldots\,q_{k-1}\,(v_{k-1},z_{k-1})\,q_k \qquad (k \geq 0)$$

such that $q_0,q_1,\ldots,q_{k-1} \in \{s,r\}$ and, for every $i<k$, one of the following holds: $v_i \in {}^\bullet q_i \;\wedge\; z_i = x \;\wedge\; q_{i+1} = x_{v_i}$ or $v_i \in q_i^\bullet \;\wedge\; z_i = e \;\wedge\; q_{i+1} = e_{v_i}$. Then the nodes of τ_e are all strings beginning with s, and of τ_x those beginning with r. There is an arc from node y to node y' if and only if $y' = ya$ for some symbol a, where we treat each q_i and each (v_i,z_i) as a symbol. The label of a node ending with q_i is q_i, and with (v_i,z_i) is v_i.

It is easy to verify that τ_e and τ_x are valid place trees (notice that adding the z_i's was needed to cope with possible places s in Ω such that ${}^\bullet s \cap s^\bullet \neq \emptyset$), and that $(\{\tau_e\},\{\tau_x\}) \subseteq \mathbf{f}(\{\tau_e\},\{\tau_x\})$ since we define place trees up to isomorphism. □

We will call the two trees constructed in the last proof *spines* associated with $\mathbf{f}$; of course, to a given $\mathbf{f}$ there may be many different spines, depending on the choice of s, r, the e_v's and the x_v's. The construction is illustrated in figure 5, where it is assumed that ${}^\bullet s = \{v_1,v_2\}$, $s^\bullet = \{v_1,v_3\}$, ${}^\bullet r = \{v_2\}$, $r^\bullet = \emptyset$, $\Delta_{v_1} = \Delta_{v_2} = X$ and $\Delta_{v_3} \neq X$.

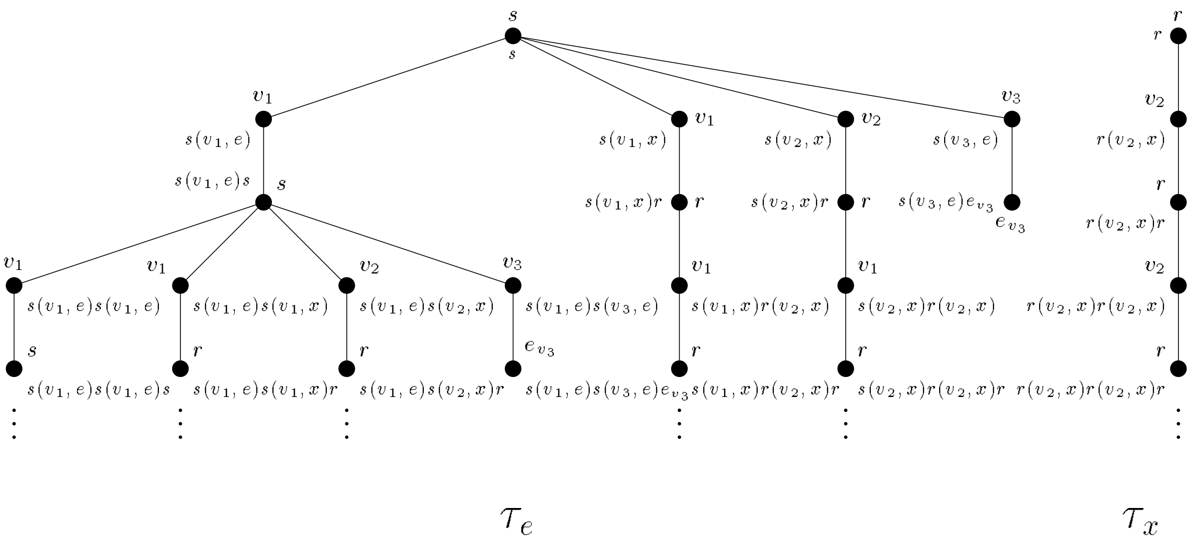


Fig. 5. Spine trees.

As a result of theorem 16, $\mathbf{f}$ has a unique minimal fixpoint, $\mathsf{fix}_{\mathsf{min}}$, and a unique maximal fixpoint, $\mathsf{fix}_{\mathsf{max}}$, which is positive. From lemma 14, using a positive fixpoint of $\mathbf{f}$ leads to a solution of (3) for $\mathbf{F}$. Putting all these together, yields:

Theorem 17. The recursive equation (3) has at least one solution. Moreover, the set of solutions has a unique maximal element.[11]
Proof: Follows from lemmata 2(2) and 14 and theorems 7 and 16. □

[11] We shall see in section 7.2 that the set of solutions of (3) does not necessarily possess a unique element.

One might be tempted to always take $\mathsf{fix}_{\mathsf{min}}$, and treat the corresponding solution as *the* solution of (3) since, by lemma 2(2), such a solution would be the minimal solution in the sense of the inclusion order $\sqsubseteq$. Unfortunately, the minimal fixpoint may fail to be positive, making it unsuitable for our purposes because boxes are ex-restricted. Below we characterise the solutions of recursive equation (3) by looking at various fixpoints of $\mathbf{f}$.

Unique solution: If fix is a unique positive fixpoint of $\mathbf{f}$ then (3) has a unique solution which can be derived from $(\mathcal{P},\mathcal{R}) = \mathsf{fix}$.

The minimal solution: If $\mathsf{fix}_{\mathsf{min}}$ is positive then (3) has a unique minimal solution with respect to $\sqsubseteq$ derived from $(\mathcal{P},\mathcal{R}) = \mathsf{fix}_{\mathsf{min}}$.

The maximal solution: There is *always* a unique maximal solution of (3) (cf. theorem 16) which can be derived from $(\mathcal{P},\mathcal{R}) = \mathsf{fix}_{\mathsf{max}}$.

Minimal solutions: If fix is a min/pos fixpoint of $\mathbf{f}$ then (3) has a minimal solution which can be derived from $(\mathcal{P},\mathcal{R}) = \mathsf{fix}$. That there is always at least one such fixpoint will be shown in section 5.5.

Guardedness: There is a simple structural condition guaranteeing that $\mathbf{f}$ is a constant mapping and therefore trivially has a unique fixpoint. The recursive equation (3) is *front-guarded* if no entry place of Ω has a direct access to the variable X; that is, if $\Delta_v \neq X$, for every $v \in {}^\bullet({}^\circ\Omega) \cup ({}^\circ\Omega)^\bullet$. Similarly, (3) is *rear-guarded* if $\Delta_v \neq X$, for every $v \in {}^\bullet(\Omega^\circ) \cup (\Omega^\circ)^\bullet$. Finally, (3) is *guarded* if it is both front- and rear-guarded.

Theorem 18. If (3) is guarded then it has a unique solution.
Proof: In the guarded case, $\mathbf{f}$ is a constant function, and so if Σ and Θ are two solutions of (3) then ${}^\circ\Sigma = {}^\circ\Theta$ and $\Sigma^\circ = \Theta^\circ$. Thus, by theorem 3, $\Sigma = \Theta$. □

The reverse implication does not hold, and we shall present a counterexample to that effect in section 7.

5.5 Minimal solutions

Every min/pos fixpoint of $\mathbf{f}$ determines a minimal solution of (3). We shall now show that there is at least one such fixpoint, and so the recursive net equation has at least one minimal solution. We first make a general observation concerning the fixpoints of $\mathbf{f}$. Recall that in the domain $(2^{\mathsf{P}_{\mathsf{tree}}} \times 2^{\mathsf{P}_{\mathsf{tree}}}, \subseteq)$, the operations of union and intersection are carried out component-wise.

Proposition 19 Let fix^i be a fixpoint of $\mathbf{f}$, for every $i \in I \neq \emptyset$. Then $\mathsf{fix} = \bigcap_{i\in I} \mathsf{fix}^i$ is a fixpoint of $\mathbf{f}$.
Proof: Let $\mathsf{fix} = (P,R)$, $\mathbf{f}(\mathsf{fix}) = (P',R')$ and $\mathsf{fix}^i = (P^i,R^i)$, for every $i \in I$. That $\mathbf{f}(\mathsf{fix}) \subseteq \mathsf{fix}$ follows immediately from the monotonicity of $\mathbf{f}$ and $\mathbf{f}(\mathsf{fix}^i) = \mathsf{fix}^i$, for every $i \in I$. To show the reverse inclusion, we observe that from $\mathbf{f}(\mathsf{fix}^i) = \mathsf{fix}^i$, for every $i \in I$, it follows that each $p \in P$ (if $P \neq \emptyset$) has the form (see also proposition 13(3))

$$p \;=\; s \lhd \Big(\{v \lhd p_v\}_{v\in s^\bullet} + \{w \lhd r_w\}_{w\in {}^\bullet s}\Big)$$

where: $s \in {}^\circ\Omega$, $p_v \in \bigcap_{i\in I} P^i$ if $\Delta_v = X$; $r_w \in \bigcap_{i\in I} R^i$ if $\Delta_w = X$; $p_v \in {}^\circ\Delta_v$ if $\Delta_v \neq X$; and $r_w \in \Delta_w^\circ$ if $\Delta_w \neq X$. Thus $p \in P'$ and so $P \subseteq P'$. By a symmetric argument, $R \subseteq R'$. Hence $\mathbf{f}(\mathsf{fix}) \supseteq \mathsf{fix}$. □

There are two immediate consequences of the last result.

If (P,R) and (P',R') are min/pos fixpoints of $\mathbf{f}$ such that $\emptyset \neq P \cap P' \notin \{P,P'\}$, then $R \cap R' = \emptyset$; similarly, if $\emptyset \neq R \cap R' \notin \{R,R'\}$ then $P \cap P' = \emptyset$.

Corollary 20 If $(P, R) \subseteq \mathsf{fix}_{\mathsf{max}}$ then there is a unique fixpoint of $\mathbf{f}$, later denoted by $\mathsf{sol}_{\mathsf{min}}(P, R)$, which is minimal among all those containing P and R.
Proof: From proposition 19 it follows that the intersection of all fixpoints of $\mathbf{f}$ containing P and R (and there is at least one such fixpoint, e.g., the maximal one) is a fixpoint of $\mathbf{f}$. $\square$

Moreover, it follows directly from the definitions of $\mathsf{sol}_{\mathsf{min}}$ and $\mathsf{fix}_{\mathsf{max}}$ that $(P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(P, R) \subseteq \mathsf{fix}_{\mathsf{max}}$, and that $\mathsf{sol}_{\mathsf{min}}$ is monotonic, i.e., $(P', R') \subseteq (P, R)$ implies $\mathsf{sol}_{\mathsf{min}}(P', R') \subseteq \mathsf{sol}_{\mathsf{min}}(P, R)$.

The result formulated as the last corollary is of an existential nature, but it is possible to give it a more constructive flavour. For $(P, R) \subseteq \mathsf{fix}_{\mathsf{max}}$, we define the *first-level suffix-set* to be the pair of sets $\mathsf{suf}(P, R) = (P', R')$ such that P' (respectively, R') comprises all places q such that there is $p \in P \cup R$ satisfying $p = s \lhd (\{v \lhd q\} + \mathcal{S})$ with $\Delta_v = X$ and the root of q is labelled by a place in ${}^\circ\Omega$ (respectively, in Ω°).[12] From $(P, R) \subseteq \mathsf{fix}_{\mathsf{max}} = \mathbf{f}(\mathsf{fix}_{\mathsf{max}})$, it follows that $\mathsf{suf}(P, R) \subseteq \mathsf{fix}_{\mathsf{max}}$ (cf. proposition 13(3)). We thus can define the *suffix-set* of (P, R) as

$$\mathsf{Suf}(P, R) \;=\; \bigcup_{i=0}^{\infty} \mathsf{suf}_i(P, R)$$

where $\mathsf{suf}_0(P, R) = (P, R)$ and $\mathsf{suf}_i(P, R) = \mathsf{suf}(\mathsf{suf}_{i-1}(P, R))$, for every $i > 0$. Notice that in $\mathsf{Suf}(P, R) = (P', R')$, the set P' (respectively, R') comprises all subtrees q of places in $P \cup R$ with the root q being labelled by a place in ${}^\circ\Omega$ (respectively, in Ω°).

Now, for a (possibly transfinite) ordinal[13] β and $K, M \subseteq \mathsf{P}_{\mathsf{tree}}$, let us define

$$\mathbf{f}^{\beta}(K, M) \;=\; \begin{cases} (K, M) & \text{if} \quad \beta = 0 \\ \mathbf{f}(\bigcup_{\gamma<\beta} \mathbf{f}^{\gamma}(K, M)) & \text{if} \quad \beta > 0. \end{cases}$$

Since $\mathbf{f}$ is monotonic, if $(K, M) \subseteq \mathbf{f}(K, M)$ then, by transfinite induction, we obtain that $(K, M) \subseteq \mathbf{f}^{\beta}(K, M) \subseteq \mathbf{f}(\mathbf{f}^{\beta}(K, M))$, for every ordinal β. Hence from Hartog's theorem [22] which says that for any cardinal there is an ordinal which is strictly greater, we obtain[14] that if $(K, M) \subseteq \mathbf{f}(K, M)$, then there is an ordinal α such that $\mathbf{f}^{\alpha}(K, M) = \mathbf{f}(\mathbf{f}^{\alpha}(K, M)) = \mathbf{f}^{\beta}(K, M)$, for every $\beta \geq \alpha$. We will denote such an α (for instance, the smallest one) by $\alpha(K, M)$; it follows that $\mathbf{f}^{\alpha(K,M)}(K, M)$ is the smallest fixpoint of $\mathbf{f}$ containing (K, M).

Proposition 21 $\mathsf{sol}_{\mathsf{min}}(P, R) = \mathbf{f}^{\alpha(\mathsf{Suf}(P,R))}(\mathsf{Suf}(P, R))$, for every pair $(P, R) \subseteq \mathsf{fix}_{\mathsf{max}}$.
Proof: We first show that $\mathsf{Suf}(P, R) \subseteq \mathbf{f}(\mathsf{Suf}(P, R))$. By the definitions of $\mathbf{f}$ and suf, we have $\mathsf{suf}_i(P, R) \subseteq \mathbf{f}(\mathsf{suf}_{i+1}(P, R))$, for every $i \geq 0$. Hence

$$\begin{aligned} \mathsf{Suf}(P, R) \;&=\; \bigcup_{i=0}^{\infty} \mathsf{suf}_i(P, R) \quad \subseteq \bigcup_{i=0}^{\infty} \mathbf{f}(\mathsf{suf}_{i+1}(P, R)) \subseteq \bigcup_{i=0}^{\infty} \mathbf{f}(\mathsf{suf}_i(P, R)) \\ &\subseteq_{(*)} \mathbf{f}\Big(\bigcup_{i=0}^{\infty} \mathsf{suf}_i(P, R)\Big) = \mathbf{f}(\mathsf{Suf}(P, R)) \end{aligned}$$

where (*) follows from $\mathbf{f}$ being monotonic. Thus $\alpha(\mathsf{Suf}(P, R))$ is well defined and $\mathbf{f}^{\alpha(\mathsf{Suf}(P,R))}(\mathsf{Suf}(P, R))$ is the smallest fixpoint of $\mathbf{f}$ containing $\mathsf{Suf}(P, R)$. As a result, to complete the proof, it suffices to show that

$$(P, R) \;\subseteq\; \mathsf{Suf}(P, R) \;\subseteq\; \mathsf{sol}_{\mathsf{min}}(P, R).$$

The first inclusion is obvious. We prove the second one by induction, in the following way. Clearly, $\mathsf{suf}_0(P, R) = (P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(P, R)$. Then, assuming $\mathsf{suf}_i(P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(P, R)$, we obtain

$$\mathsf{suf}_{i+1}(P, R) \;=\; \mathsf{suf}(\mathsf{suf}_i(P, R)) \;\subseteq_{(**)}\; \mathsf{suf}(\mathsf{sol}_{\mathsf{min}}(P, R)) \;\subseteq_{(***)}\; \mathsf{sol}_{\mathsf{min}}(P, R)$$

where (**) follows from the induction hypothesis and suf being monotonic, and (***) follows from $\mathsf{sol}_{\mathsf{min}}(P, R)$ being a fixpoint of $\mathbf{f}$, more precisely, from $\mathsf{sol}_{\mathsf{min}}(P, R) \subseteq \mathbf{f}(\mathsf{sol}_{\mathsf{min}}(P, R))$. $\square$

[12] Notice that, by the definition of $\mathbf{f}$, the root of each place in $\mathsf{fix}_{\mathsf{max}}$ is labelled by a place in ${}^\circ\Omega$ or Ω°, and that p is a place in $\mathbf{f}(\mathsf{fix}_{\mathsf{max}})$.
[13] The definition is well-formed since there is no infinite descending chain of ordinals.
[14] This is a general result of the fixpoint theory.

Corollary 22 Let $(P, R) \subseteq \mathsf{fix}_{\mathsf{max}}$. Then, for every place p in $\mathsf{sol}_{\mathsf{min}}(P, R)$, on every infinite path (if any) from the root of p there is a node ξ such that $\mathsf{subtr}(\xi)$ belongs to $\mathsf{Suf}(P, R)$, where $\mathsf{subtr}(\xi)$ denotes the *subtree* appended at ξ.
Proof: Follows by transfinite induction from proposition 21. Another argument is that if $\mathsf{sol}_{\mathsf{min}}(P, R)$ were to contain trees p not fulfilling the condition, by dropping all of them we would get a strictly smaller set, which would still be a fixpoint of $\mathbf{f}$ containing P and R, contradicting the minimality of $\mathsf{sol}_{\mathsf{min}}(P, R)$. $\square$

Let $(P, R) \subseteq \mathsf{fix}_{\mathsf{max}}$ be a pair of non-empty sets of places. Although $\mathsf{sol}_{\mathsf{min}}(P, R)$ is a positive fixpoint of $\mathbf{f}$ and minimal among those containing P and R, it may not be minimal among all positive fixpoints of $\mathbf{f}$ (even if P and R are singletons). We therefore need to be careful in choosing P and R to guarantee that $\mathsf{sol}_{\mathsf{min}}(P, R)$ is a min/pos fixpoint. Interestingly, the spine trees constructed in theorem 16 will prove useful in looking for the right P and R. Below, we use an auxiliary mapping $g \ : \ {}^\circ\Omega \cup \Omega^\circ \to 2^{\{e,x\}}$ such that $e \in g(s)$ if and only if $\{v \in s^\bullet \mid \Delta_v = X\} \neq \emptyset$, and $x \in g(s)$ if and only if $\{v \in {}^\bullet s \mid \Delta_v = X\} \neq \emptyset$. Intuitively, $g(s)$ reflects the non-guardedness of an entry or exit place s in the operator box Ω; in particular, the recursive equation (3) is guarded if and only if $g({}^\circ\Omega \cup \Omega^\circ) = \{\emptyset\}$.

Theorem 23. $\mathbf{f}$ has a min/pos fixpoint.
Proof: In this proof, we will make several applications of the spine trees, τ_e and τ_x, defined as in the proof of theorem 16 and illustrated in figure 5, for different pairs of places $s \in {}^\circ\Omega$ and $r \in \Omega^\circ$. As already observed, we have $(T_e, T_x) \subseteq \mathbf{f}(T_e, T_x) \subseteq \mathsf{fix}_{\mathsf{max}}$, where here and later in this proof we use T_e and T_x to denote $\{\tau_e\}$ and $\{\tau_x\}$, respectively. Moreover, it is easy to see that $\mathsf{Suf}(T_e, T_x) = (T_e, T_x)$. Thus, by corollary 22,

for every place p in $\mathsf{sol}_{\mathsf{min}}(T_e, T_x)$, on every infinite path (if any) from the root of p there is a node ξ such that $\mathsf{subtr}(\xi) \in \{\tau_e, \tau_x\}$. (*)

Suppose first that $\mathbf{f}(\emptyset, \emptyset) = (K, M) \neq (\emptyset, \emptyset)$. Without loss of generality, we assume that $K \neq \emptyset$. Then there is $s \in {}^\circ\Omega$ such that $g(s) = \emptyset$, and we consider two cases.

Case 1: There is $r \in \Omega^\circ$ such that $x \notin g(r)$. Let τ_e and τ_x be defined for the current s and r. Then $(T_e, T_x) \subseteq \mathbf{f}^2(\emptyset, \emptyset)$ and so $\mathsf{fix}_{\mathsf{min}} = \mathsf{sol}_{\mathsf{min}}(\emptyset, \emptyset)$ is positive. Clearly, in such a case, it is the only min/pos fixpoint of $\mathbf{f}$, since $\mathsf{sol}_{\mathsf{min}}(\emptyset, \emptyset)$ is always the minimal fixpoint of $\mathbf{f}$.

Case 2: For all $r \in \Omega^\circ$, $x \in g(r)$. Take any $r \in \Omega^\circ$. Let τ_e and τ_x be defined for the current s and r. Suppose that $(P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(T_e, T_x)$ is a positive fixpoint. Clearly, $\tau_e \in P$. To show that $\tau_x \in R$, take any $q \in R$. From $\mathsf{Suf}(P, R) \subseteq (P, R) = \mathbf{f}(P, R)$ and $x \in g(r)$, for all $r \in \Omega^\circ$, it follows that there is an infinite path ϕ from the root of q such that no node of ϕ is labelled by a place in ${}^\circ\Omega$. Moreover, by (*), there is a node ξ on ϕ such that $\mathsf{subtr}(\xi) \in \{\tau_e, \tau_x\}$. Hence, since the root of τ_e belongs to ${}^\circ\Omega$, we obtain that τ_x belongs to $\mathsf{Suf}(\emptyset, \{q\})$. Hence $(T_e, T_x) \subseteq \mathsf{Suf}(P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(P, R) = (P, R)$. Thus, by $\mathsf{sol}_{\mathsf{min}}$ being monotonic, $\mathsf{sol}_{\mathsf{min}}(T_e, T_x) \subseteq (P, R)$, and so $\mathsf{sol}_{\mathsf{min}}(T_e, T_x) = (P, R)$ is a min/pos fixpoint of $\mathbf{f}$.

Having shown the result for the case $\mathbf{f}(\emptyset, \emptyset) \neq (\emptyset, \emptyset)$, we now assume the opposite, i.e., $\mathbf{f}(\emptyset, \emptyset) = (\emptyset, \emptyset)$. This means that $\emptyset \notin g({}^\circ\Omega \cup \Omega^\circ)$ which in turn implies that

every tree in $\mathsf{fix}_{\mathsf{max}}$ has an infinite path from its root. (**)

Suppose there are $s \in {}^\circ\Omega$ and $r \in \Omega^\circ$ such that $x \in g(s)$ and $e \in g(r)$. Let τ_e and τ_x be defined for the current s and r. Moreover, let $(P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(T_e, T_x)$ be a positive fixpoint of $\mathbf{f}$ and $p \in P$. By $\emptyset \notin g({}^\circ\Omega \cup \Omega^\circ)$ and (*) and (**), $\tau_e \in P'$ or $\tau_x \in R'$ for $(P', R') = \mathsf{Suf}(\{p\}, \emptyset)$. Moreover, for the current s and r it is the case that $\mathsf{Suf}(\emptyset, T_x) = \mathsf{Suf}(T_e, \emptyset) = (T_e, T_x)$. Hence $(T_e, T_x) \subseteq \mathsf{Suf}(P, R)$ and, as in Case 2, $\mathsf{sol}_{\mathsf{min}}(T_e, T_x)$ is a min/pos fixpoint of $\mathbf{f}$.

If s and r as above cannot be found, we assume without loss of generality, that $g({}^\circ\Omega) = \{\{e\}\}$. Then we choose any $s \in {}^\circ\Omega$ and consider two cases.

Case 3: There is $r \in \Omega^\circ$ such that $g(r) = \{e\}$. Let τ_e and τ_x be defined for the current s and r. Notice that we have in this case: $\mathsf{Suf}(T_e, \emptyset) = (T_e, \emptyset)$, $\mathsf{Suf}(\emptyset, T_x) = (T_e, T_x)$ and $(T_e, T_x) \subseteq \mathbf{f}(T_e, \emptyset) \subseteq \mathsf{sol}_{\mathsf{min}}(T_e, \emptyset)$; hence $\mathsf{sol}_{\mathsf{min}}(T_e, T_x) = \mathsf{sol}_{\mathsf{min}}(T_e, \emptyset) = \mathsf{sol}_{\mathsf{min}}(\emptyset, T_x)$. It can then be seen that $\mathsf{sol}_{\mathsf{min}}(T_e, T_x)$ is a min/pos fixpoint of $\mathbf{f}$. Indeed, let $(P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(T_e, T_x)$ be a positive fixpoint. By (*) and (**), $\tau_e \in P'$ or $\tau_x \in R'$ for $(P', R') = \mathsf{Suf}(\{p\}, \emptyset)$. Thus, by $\mathsf{Suf}(\emptyset, T_x) = (T_e, T_x)$, $\tau_e \in P'$. Hence $(T_e, \emptyset) \subseteq \mathsf{Suf}(P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(P, R) = (P, R)$ and, by $\mathsf{sol}_{\mathsf{min}}(T_e, T_x) = \mathsf{sol}_{\mathsf{min}}(T_e, \emptyset)$, $\mathsf{sol}_{\mathsf{min}}(T_e, T_x) = (P, R)$.

Case 4: For all $r \in \Omega^\circ$, $x \in g(r)$. Take any $r \in \Omega^\circ$. Let τ_e and τ_x be defined for the current s and r. Notice that we still have in this case $\mathsf{Suf}(T_e, \emptyset) = (T_e, \emptyset)$. Let $(P, R) \subseteq \mathsf{sol}_{\mathsf{min}}(T_e, T_x)$ be a positive fixpoint of $\mathbf{f}$. Take any $p \in P$. By (*) and (**) and the fact that no node in p is labelled by a place in Ω°, we obtain $(T_e, \emptyset) \subseteq \mathsf{Suf}(P, R)$. Similarly, $(\emptyset, T_x) \subseteq \mathsf{Suf}(P, R)$ follows from (*) and (**) and the fact that there is an infinite path in r such that no node along this path is labelled by a place in ${}^\circ\Omega$. Hence $(T_e, T_x) \subseteq \mathsf{Suf}(P, R)$ and, again, $\mathsf{sol}_{\mathsf{min}}(T_e, T_x)$ is a min/pos fixpoint of $\mathbf{f}$. □

The first part of the proof has identified a sufficient condition for the existence of a unique min/pos fixpoint of $\mathbf{f}$.

Corollary 24 Suppose that the operator box Ω in (3) is such that one of the following holds:

- $\emptyset \in g({}^\circ\Omega) \cap g(\Omega^\circ)$.
- $\emptyset \in g({}^\circ\Omega)$ and $\{e\} \in g(\Omega^\circ)$.
- $\emptyset \in g(\Omega^\circ)$ and $\{x\} \in g({}^\circ\Omega)$.

Then there is a unique min/pos fixpoint of $\mathbf{f}$. □

6 Finitary equations and finite operator boxes

We now look what conclusions can be drawn from assuming that (3) is an X-finitary equation (cf. section 5.1) or, still stronger, that the operator box appearing in it is finite. The interest in the latter is motivated by the application of our theory to giving a compositional net semantics to the standard PBC [4], where all the operators are finite.

6.1 Finitary equation

For an X-finitary equation the auxiliary mapping $\mathbf{f}$ is not only monotonic but also continuous (cf. proposition 5 for $\mathbf{F}$).

Proposition 25 If (3) is X-finitary then $\mathbf{f}$ is continuous.
Proof: Let $P^0 \subseteq P^1 \subseteq \ldots$ be a chain in $(2^{\mathsf{P}_{\mathsf{tree}}} \times 2^{\mathsf{P}_{\mathsf{tree}}}, \subseteq)$ with the limit $P = \bigcup_{i=0}^\infty P^i$. The inclusion $\mathbf{f}(P) \supseteq \bigcup_{i=0}^\infty \mathbf{f}(P^i)$ follows from the monotonicity of $\mathbf{f}$ (see lemma 15). As for the reverse inclusion, suppose that

$$p = s \lhd (\{v \lhd p_v\}_{v \in {}^\bullet s} + \{w \lhd r_w\}_{w \in s^\bullet})$$

is a place such that, without loss of generality, $(\{p\}, \emptyset) \subseteq \mathbf{f}(P)$. Let $U = \{u \in {}^\bullet s \cup s^\bullet \mid \Delta_u = X\}$. Since U is finite (as (3) is X-finitary), from the definition of the chain it follows that there is $i \geq 0$ such that for every $u \in U$, each p_u and/or r_u belongs to P^i. Hence $(\{p\}, \emptyset) \subseteq \mathbf{f}(P^i)$ and so $(\{p\}, \emptyset) \subseteq \bigcup_{i=0}^\infty \mathbf{f}(P^i)$. Thus $\mathbf{f}(P) \subseteq \bigcup_{i=0}^\infty \mathbf{f}(P^i)$. □

Continuity is useful as we can calculate some fixpoints of $\mathbf{f}$ by taking $(\mathcal{P}, \mathcal{R}) = \bigcup_{i=0}^\infty \mathbf{f}^i(\{\tau_e\}, \{\tau_x\})$ provided that τ_e and τ_x are place trees satisfying $(\{\tau_e\}, \{\tau_x\}) \subseteq \mathbf{f}(\{\tau_e\}, \{\tau_x\})$, without resorting to a transfinite construction (see proposition 30 below). And recall that the existence of such trees has been demonstrated in the proof of theorem 16. But the discussion of X-finitary equations does not end here. As we have shown in proposition 5, if (3) is X-finitary then the mapping $\mathbf{F}$ is $\sqsubseteq$-continuous rather than only $\tilde{\sqsubseteq}$-continuous. An immediate consequence of this fact is that the definition of a seed box can be changed, in the following way.

A rec-box Υ is a $\sqsubseteq$-*seed* for equation (3) if $\Upsilon \sqsubseteq \mathbf{F}(\Upsilon)$. What is interesting is that the results obtained previously for $\tilde{\sqsubseteq}$-seed boxes carry over to the modified setting.

Theorem 26. If Υ is a $\sqsubseteq$-seed for an X-finitary equation (3) then $\bigsqcup \mathbf{F}^*(\Upsilon)$ is a solution.
Proof: Similar to the proof of theorem 7. □

The search for $\sqsubseteq$-seeds can be carried out similarly as in section 5.4.

Lemma 27 Let Υ be an ex-box and $\mathcal{P} = {}^\circ\Upsilon$ and $\mathcal{R} = \Upsilon^\circ$. Then the following are equivalent:

(1) Υ is a $\sqsubseteq$-seed.
(2) The sets $\mathcal{P}$ and $\mathcal{R}$ are non-empty and $(\mathcal{P}, \mathcal{R}) \subseteq \mathbf{f}(\mathcal{P}, \mathcal{R})$.

Proof: Similar to the proof of lemma 14. □

What is more, from the proof of theorem 16, it is always possible to find a pair of *singleton* sets $\mathcal{P} = \{\tau_e\}$ and $\mathcal{R} = \{\tau_x\}$ satisfying $(\mathcal{P}, \mathcal{R}) \subseteq \mathbf{f}(\mathcal{P}, \mathcal{R})$. It therefore follows that in the X-finitary case it is always possible to obtain a solution of (3) by starting from an ex-box with exactly two places. In terms of approximating a solution we obtain the following.

Theorem 28. If (3) is X-finitary then it has a solution which can be approximated from a $\sqsubseteq$-seed with exactly two places.
Proof: Follows from theorem 26 and lemma 27 and the above remark. □

However, this does not mean that *any* solution can be so approximated. Since there is a solution which can be approximated by starting from a seed box with exactly one entry and one exit place, if all boxes occurring in the recursive equation are countable, then all rec-boxes in the chain derived from such a seed box will have countably many places, and the solution given by the limit will also have countably many places. However, the same reasoning cannot be made for transitions, due to the fact that there may be uncountably infinitely many labels α such that $(\lambda_{\Sigma_v}(R), \alpha)$ belongs to $\lambda_\Omega(v)$, and so each such α may give rise to a separate transition $(v, \alpha) \lhd R$. This motivates the following definition: a relabelling ρ is *image-countable* if, for every finite multiset of labels Γ, $\{\alpha \mid (\Gamma, \alpha) \in \rho\}$ is countable. An immediate consequence of the last definition is

Theorem 29. Let (3) be an X-finitary recursive equation such that all boxes occurring in it are countable and all relabellings of the operator box are image-countable. Then the equation has a countable solution which can be approximated by a $\sqsubseteq$-chain of countable rec-boxes which begins with an ex-box comprising exactly two places. □

Another interesting feature of the X-finitary case is that it is not necessary to use transfinite constructions to obtain min/pos fixpoints containing some sets of places, whose existence was shown in section 5.5. Indeed, in the X-finitary case, from the continuity of the auxiliary function $\mathbf{f}$ it follows that $\alpha(\mathsf{Suf}(P, R)) \le \omega$, where ω is the first transfinite ordinal.

Proposition 30 If (3) is X-finitary then $\mathsf{sol}_{\mathsf{min}}(P, R) = \bigcup_{i=0}^{\infty} \mathbf{f}^i(\mathsf{Suf}(P, R))$, for every pair $(P, R) \subseteq \mathsf{fix}_{\mathsf{max}}$.
Proof: Follows immediately from propositions 21 and 25. □

Corollary 31 If all boxes occurring in an X-finitary (3) are countable then each min/pos fixpoint of $\mathbf{f}$ is countable.
Proof: Let (P, R) be a min/pos fixpoint of $\mathbf{f}$, and $p \in P$ and $r \in R$. Since the equation is X-finitary, for each countable $(K, M) \subseteq \mathsf{fix}_{\mathsf{max}}$, the first-level suffix $\mathsf{suf}(K, M)$ is countable too. Hence $\mathsf{Suf}(\{p\}, \{r\})$ is countable and, since all boxes occurring in (3) are countable and the equation is X-finitary, so is each $\mathbf{f}^i(\mathsf{Suf}(\{p\}, \{r\}))$. Thus, by proposition 30, $\mathsf{sol}_{\mathsf{min}}(\{p\}, \{r\}) = (P, R)$ is countable. □

6.2 Finite operator box

We now assume that the operator box in (3) is finite (hence X-finitary), and that each box in the Ω–tuple Δ is finite too. Our aim is to look what conditions need to be satisfied to guarantee that the equation has countable solutions which can be approximated by $\sqsubseteq$-chains of finite boxes.

To begin with, we can apply theorem 28 to conclude that there is a solution which can be approximated by starting from a $\sqsubseteq$-seed with exactly two places and no transitions. Hence, since all nets occurring in (3) are finite, all rec-boxes in the chain derived from such a seed will have finitely many places. And the solution given by the limit will have at most countably many places. However, again, the same reasoning cannot be made for transitions. Indeed, since the definition of net refinement $\Omega(\Sigma)$ uses finite multisets

R to construct new transitions of the form $(v, \alpha) \lhd R$, there is a possibility of constructing infinitely many transitions even if Ω and Σ only have finitely many transitions. As a matter of fact, there are two potential sources of infinity.

The first one is the possibility of there being infinitely many labels α such that $(\lambda_{\Sigma_v}(R), \alpha)$ belongs to $\lambda_\Omega(v)$, since each such α may give rise to a separate transition $(v, \alpha) \lhd R$. This motivates the following definition, strengthening the one already used in the previous section: a relabelling ρ is *image-finitary* if, for every finite multiset of labels Γ, $\{\alpha \mid (\Gamma, \alpha) \in \rho\}$ is finite. And it is worth noting in this context that all relabellings used to define operator boxes in the translation of PBC expressions are image-finitary [4, 3, 6, 7]. The second source of infinity is that we allow R to be a multiset of transitions rather than a set. Hence, it may be interesting to reinforce the condition above: a relabelling ρ is *image-finite* if, for every finite subset Lab' of Lab, $\{(\Gamma, \alpha) \in \rho \mid \Gamma \in \mathsf{mult}(\mathsf{Lab}')\}$ is finite. An immediate consequence of the last definition is

Theorem 32. If all boxes occurring in (3) are finite, and all relabellings in Ω are image-finite, then the equation has a countable solution which can be approximated by a $\sqsubseteq$-chain of finite rec-boxes which begins with a $\sqsubseteq$-seed comprising exactly two places and no transitions.
Proof: We observe that if Ω is finite with image-finite relabellings and all boxes in Δ have finitely many transitions, if Σ is a rec-box with finitely many transitions, then $\mathbf{F}(\Sigma)$ has finitely many transitions too. One then only needs to use theorem 28. □

Note that taking the limit of a $\sqsubseteq$-chain of boxes will typically produce an infinite box even if the equation was based on finite boxes.

In practical applications, initial fragments of the limit construction can be used to derive finite approximations of an infinite solution. We also expect that the results obtained here will provide a basis for high-level net solutions derived by folding infinite labelled nets into finite high-level representations.

7 Three examples

Let us return to the first limit example (5) for whom four initial approximations were shown in figure 4. It leads to the following tree equations for the entry and exit places of the starting box:

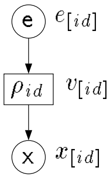
$$\begin{aligned} \mathcal{P} &= \{\, e \lhd v_1 \lhd e_a \,\} \\ \mathcal{R} &= \{\, x \lhd \{v_2 \lhd r \,,\ v_3 \lhd x_b\} \mid r \in \mathcal{R} \,\}. \end{aligned}$$

In such a case $\mathbf{f}$ has a unique (positive) fixpoint $(\mathcal{P}, \mathcal{R}) = (\{\tau_e\}, \{\tau_x\})$, where τ_e and τ_x are as in figure 4. And we can conclude that the net equation (5) has a unique solution, namely the box Σ shown in figure 4.

7.1 Extreme unguardedness

As the next equation – which informally corresponds to $X \stackrel{\mathrm{df}}{=} X$ – consider the following:

$$X \quad \stackrel{\mathrm{df}}{=}$$



$$\left(\quad X \quad \right) \tag{8}$$

It leads to the following tree equations for the entry and exit places of the starting box:

$$\begin{aligned} \mathcal{P} &= \{\, e_{[id]} \lhd v_{[id]} \lhd p \mid p \in \mathcal{P} \,\} \\ \mathcal{R} &= \{\, x_{[id]} \lhd v_{[id]} \lhd r \mid r \in \mathcal{R} \,\}. \end{aligned}$$

Clearly, in such a case $\mathbf{f}$ has a unique positive fixpoint

$$(\mathcal{P} \,,\, \mathcal{R}) \;=\; \Big(\;\{\, e_{[id]} \lhd v_{[id]} \lhd e_{[id]} \lhd v_{[id]} \lhd \ldots \} \;\;,\;\; \{\, x_{[id]} \lhd v_{[id]} \lhd x_{[id]} \lhd v_{[id]} \lhd \ldots \} \;\Big).$$

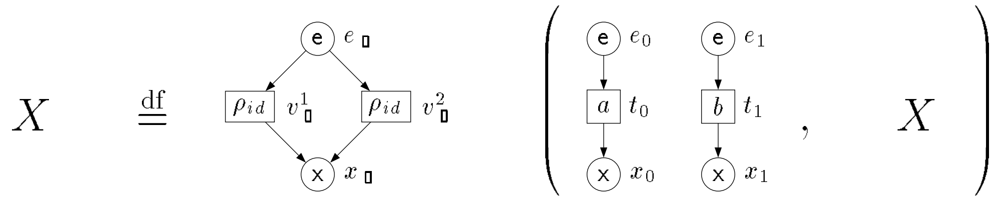
Hence approximation never changes anything, and the box solving the last net equation has exactly one entry place, one exit place, and no transitions. Note that despite the colloquial meaning of $X \stackrel{\mathrm{df}}{=} X$, no box with a transition solves (8). This results from the fact that we only allow finite transition names. Indeed, if we allowed infinite ones, we could accept a solution with transitions $(v_{[id]}, \alpha) \lhd (v_{[id]}, \alpha) \lhd (v_{[id]}, \alpha) \lhd \ldots$ for any label α; the reader may verify that this would be faithfully reproduced by $\mathbf{F}$. Thus, solving recursive equation by boxes with least behaviour is already in-built in our approach.

We feel that this gives an intuitive semantics to the 'recursive call' $X \stackrel{\mathrm{df}}{=} X$ because we do not model procedure entry and procedure exit as separate (silent) actions. Thus, $X \stackrel{\mathrm{df}}{=} X$ essentially means that X is continually 'rewritten' by itself and effects no behaviour at all, be it silent or observable. This is quite in contrast to the view that $X \stackrel{\mathrm{df}}{=} X$ implies continual entry to a procedure and thus an infinite succession of silent moves; to model such behaviour, we would rather use $X \stackrel{\mathrm{df}}{=} \emptyset; X$ (or even $X \stackrel{\mathrm{df}}{=} \emptyset; X; \emptyset$, to model exit as well as entry, even though none of the transitions coming from the $\emptyset$ following X would be executable).

The literature is divided over the treatment of unboundedly recursive calls such as $X \stackrel{\mathrm{df}}{=} X$. For instance, [1] agrees with our interpretation; other authors [11] advocate that $X \stackrel{\mathrm{df}}{=} X$ generates an infinite sequence of silent steps (modelling continuous entry to the procedure); and still other authors (such as [12]) consider $X \stackrel{\mathrm{df}}{=} X$ to denote 'chaos', in the sense of generating all possible infinite sequences as the behaviour.

7.2 Concurrency within unbounded choice

As another example, consider



(9)

which corresponds to $X \stackrel{\mathrm{df}}{=} (a\|b) \,\square\, X$. Applying the same rules as previously leads us to the following tree equations:

$$\begin{aligned} \mathcal{P} &= \{\, e_\square \lhd \{v^1_\square \lhd q,\, v^2_\square \lhd p\} \;\mid\; q \in \{e_0, e_1\} \;\wedge\; p \in \mathcal{P} \,\} \\ \mathcal{R} &= \{\, x_\square \lhd \{v^1_\square \lhd q,\, v^2_\square \lhd r\} \;\mid\; q \in \{x_0, x_1\} \;\wedge\; r \in \mathcal{R} \,\}. \end{aligned} \tag{10}$$

This case is interesting because, as it turns out, the maximal fixpoint of $\mathbf{f}$ has uncountably many trees. The generic trees for $\mathsf{fix}_{\mathsf{max}}$ are shown in figure 6 where $n_i \in \{e_0, e_1\}$ and $m_i \in \{x_0, x_1\}$, for every i.

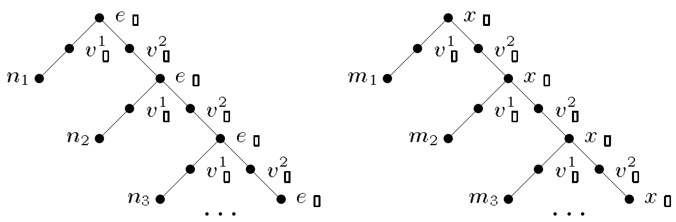


Fig. 6. Generic trees $\tau^e_{n_1 n_2 n_3 \ldots}$ and $\tau^x_{m_1 m_2 m_3 \ldots}$ for the entry and exit places.

Clearly, starting the approximation with an uncountably large seed box corresponding to $\mathsf{fix}_{\mathsf{max}}$ yields an uncountably large solution of the equation (9) (see also [3, 15]). Nevertheless, (9) also has countable solutions, since the relabellings in the the operator box are image-finite and both boxes occurring in it are finite (see theorem 29). Moreover, we know that such solutions can be obtained by starting the approximation with $\sqsubseteq$-seeds Υ having exactly one entry and one exit place. To determine such seeds, we construct spines as in the proof of theorem 16; figure 7 shows two of them, for the entry and exit place of Υ. Let Υ be the $\sqsubseteq$-seed ex-box defined by the spines in figure 7, and Υ' be the $\sqsubseteq$-seed ex-box defined by the spines obtained from those in figure 7 by replacing each node label e_0 by e_1, and x_0 by x_1. Starting either with Υ or with Υ' yields a countable solution of (9). Due to lemma 2(1), the two solutions have the same transitions and, according to the results contained in [7], also the same behaviour. But they are not only incomparable in the sense of $\sqsubseteq$ but also non-isomorphic. This is despite the fact that at every step in the approximation, isomorphic nets are obtained (albeit with incompatible isomorphisms). Note also that the maximal solution cannot be obtained by starting with a countable seed (*a fortiori*, neither can it be obtained from a two-place seed).

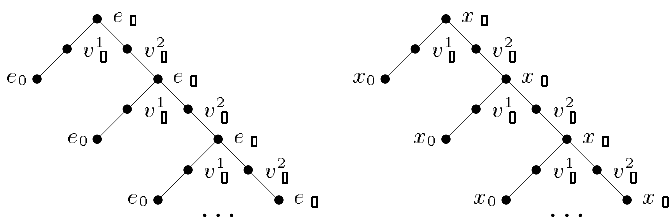


Fig. 7. Two spines for equation (10).

This example is of further interest since it has uncountably many different minimal solutions. For consider the trees of the maximal fixpoint, $\mathsf{fix}_{\mathsf{max}}$, shown in figure 6. It is easy to see (from proposition 30) that if we take any $P = \{\tau^e_{n_1 n_2 n_3 \ldots}\}$ and $R = \{\tau^x_{m_1 m_2 m_3 \ldots}\}$ then

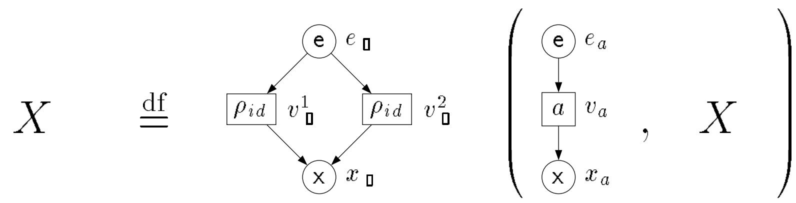
$$\mathsf{sol}_{\mathsf{min}}(P, R) \;=\; \begin{pmatrix} \{\tau^e_{l_1 \ldots l_k n_i n_{i+1} \ldots} & \mid i \geq 1 \wedge l_1, \ldots, l_k \in \{e_0, e_1\}\} \\ \{\tau^x_{l_1 \ldots l_k m_i m_{i+1} \ldots} & \mid i \geq 1 \wedge l_1, \ldots, l_k \in \{x_0, x_1\}\} \end{pmatrix}$$

is a countable min/pos fixpoint of $\mathbf{f}$. Let Sol be the set of all such fixpoints.

Proposition 33 Sol is an uncountable set of min/pos fixpoints of $\mathbf{f}$.
Proof: We have $\bigcup_{\mathsf{sol}_{\mathsf{min}}(P,R) \in \mathsf{Sol}} \mathsf{sol}_{\mathsf{min}}(P, R) = \mathsf{fix}_{\mathsf{max}}$. Since $\mathsf{fix}_{\mathsf{max}}$ is uncountable and each $\mathsf{sol}_{\mathsf{min}}(P, R) \in \mathsf{Sol}$ countable, it follows that Sol is uncountable. □

7.3 (Non-)use of empty nets in the limit construction

Finally, consider the following equation:



(11)

which corresponds to $X \stackrel{\mathrm{df}}{=} a \,□\, X$. This equation can be dealt with in a similar way as equation (10), and so we omit the details. The reason why we give the example, is to show that the empty net is not suitable as a starting point of the approximation, and so it justifies the introduction of ex-restrictedness in our framework. For consider that the empty net were taken as a seed box. Refinement would work as usual, and we would obtain as solution a net containing infinitely many isolated a-labelled transitions. Such a solution would allow an arbitrarily large step of a-labelled transitions, which is both counterintuitive and inconsistent with the usual operational semantics of the choice construct [7]. If the approximation is instead realised in the way we have described, then we obtain as solution a choice between infinitely many a-labelled transitions, which is both intuitively correct and consistent with the operational semantics of □.

8 Multiple equation systems

Let $\mathcal{X}$ be a non-empty, possibly infinite, set of *recursion variables*. We consider a system of *equations on boxes* of the following form (one equation per variable X in $\mathcal{X}$):

$$X \stackrel{\mathrm{df}}{=} \Omega_X(\mathbf{\Delta}_X). \tag{12}$$

On the right-hand side, Ω_X is an operator box, and $\mathbf{\Delta}_X$ is an Ω_X–tuple of recursion variables and rec-boxes, $\mathbf{\Delta}_X : T_{\Omega_X} \to \mathcal{X} \cup \mathsf{Box}^{rec}$. Now, given a mapping $\mathsf{sol} : \mathcal{X} \to \mathsf{Box}^{rec}$ assigning a rec-box to every variable in $\mathcal{X}$, we will denote by $\mathbf{\Delta}_X[\mathsf{sol}]$ the Ω_X–tuple obtained from $\mathbf{\Delta}_X$ by replacing each variable Y by $\mathsf{sol}(Y)$. That is, $\mathbf{\Delta}_X[\mathsf{sol}] : T_{\Omega_X} \to \mathsf{Box}^{rec}$ is such that for every $v \in T_{\Omega_X}$,

$$\mathbf{\Delta}_X[\mathsf{sol}](v) = \begin{cases} \mathsf{sol}(Y) & \text{if } \Delta_{X,v} = Y \in \mathcal{X} \\ \Delta_{X,v} & \text{if } \Delta_{X,v} \in \mathsf{Box}^{rec}. \end{cases}$$

Then, a *solution* of the system of equations (12) is an assignment sol such that, for every variable $X \in \mathcal{X}$, $\mathsf{sol}(X) = \Omega_X(\mathbf{\Delta}_X[\mathsf{sol}])$.

The results obtained for a single recursive equation (3) are easily lifted to the general case, i.e., to the recursive system (12), using the standard techniques on Cartesian product spaces (for a finite set of equations) or on function spaces (for an infinite set of equations). Below we consider various $\mathcal{X}$*–families* (tuples)[15] of boxes, sets, functions, etc. An $\mathcal{X}$–family will be denoted as $\check{\mathcal{F}}$, and its member corresponding to $X \in \mathcal{X}$ by $\mathcal{F}_X$. The inclusion orders $\sqsubseteq$ and $\tilde{\sqsubseteq}$, and more generally any predicate $\sim$, will be extended to $\mathcal{X}$–families, by stipulating that $\check{\mathcal{F}} \sim \check{\mathcal{G}}$ if $\mathcal{F}_X \sim \mathcal{G}_X$, for every $X \in \mathcal{X}$. Similarly as for (3), we assume that all rec-boxes occurring in (12) have their transitions and places in $\mathsf{T_{root}} \setminus \bigcup_{X\in\mathcal{X}} T_{\Omega_X}$ and $\mathsf{P_{root}} \setminus \bigcup_{X\in\mathcal{X}} S_{\Omega_X}$, respectively.

A $\tilde{\sqsubseteq}$-seed for the system of equations (12) is an $\mathcal{X}$–family of rec-boxes $\check{\Upsilon}$ such that $\check{\Upsilon} \,\tilde{\sqsubseteq}\, \check{\Theta}$, where, for every $X \in \mathcal{X}$, $\Theta_X = \mathbf{F}_X(\check{\Upsilon}) = \Omega_X(\mathbf{\Delta}_X[\check{\Upsilon}])$. A $\sqsubseteq$-seed satisfies the same property, but with $\tilde{\sqsubseteq}$ replaced by $\sqsubseteq$. As for a single equation, $\check{\mathbf{F}}$ is a $\sqsubseteq$-monotonic and $\tilde{\sqsubseteq}$-continuous $\mathcal{X}$–family of mappings.[16] And it is $\sqsubseteq$-continuous if and only if the system of equations (12) is $\mathcal{X}$*-finitary*, i.e., for every $X \in \mathcal{X}$ and every $s \in S_{\Omega_X}$, $\{v \in {}^\bullet s \cup s^\bullet \mid \Delta_{X,v} \in \mathcal{X}\}$ is finite. Moreover, if $\check{\Upsilon}$ is a $\tilde{\sqsubseteq}$-seed for (12) (or a $\sqsubseteq$-seed, if the system is $\mathcal{X}$-finitary), then $\bigsqcup \check{\mathbf{F}}^*(\check{\Upsilon})$ is a solution of this system of equations. For the same reasons as before, the transitions and finite prefixes of places of a solution are unique (for each $X \in \mathcal{X}$), and the solution generated from a $\tilde{\sqsubseteq}$-seed only depends on the entry and exit places of the latter. It is always possible to start from ex-boxes as seeds, and their entry/exit place sets have to satisfy the same kind of properties as for a single equation, but lifted to $\mathcal{X}$–families, $(\check{\mathcal{P}}, \check{\mathcal{R}}) = \check{\mathbf{f}}(\check{\mathcal{P}}, \check{\mathcal{R}})$ (or $(\check{\mathcal{P}}, \check{\mathcal{R}}) \subseteq \check{\mathbf{f}}(\check{\mathcal{P}}, \check{\mathcal{R}})$ for $\sqsubseteq$-seeds). Here, for all $X \in \mathcal{X}$, and all $\mathcal{X}$–families of place trees $\check{P}$ and $\check{R}$,

$$\mathbf{f}_X(\check{P}, \check{R}) = \Big(\bigcup_{s\in{}^\circ\Omega_X} \mathbf{f}_{X,s}(\check{P}, \check{R}) \;,\; \bigcup_{s\in\Omega_X^\circ} \mathbf{f}_{X,s}(\check{P}, \check{R}) \Big)$$

where each $\mathbf{f}_{X,s}(\check{P}, \check{R})$ comprises all place trees $s \lhd (\{v \lhd p_v\}_{v\in s^\bullet} + \{w \lhd r_w\}_{w\in{}^\bullet s})$ such that:

$$p_v \in \begin{cases} P_Y & \text{if } \Delta_{X,v} = Y \in \mathcal{X} \\ {}^\circ\Delta_{X,v} & \text{if } \Delta_{X,v} \notin \mathcal{X} \end{cases} \quad \text{and} \quad r_w \in \begin{cases} R_Y & \text{if } \Delta_{X,w} = Y \in \mathcal{X} \\ \Delta_{X,w}^\circ & \text{if } \Delta_{X,w} \notin \mathcal{X}. \end{cases}$$

That there is always a unique positive maximal solution, and possibly many min/pos ones, arises from a generalisation of the construction of spine trees, τ_e and τ_x, in theorem 16. We first build a labelled directed graph G, in the following way.

Let us choose, for every $X \in \mathcal{X}$, an entry place $e_X \in {}^\circ\Omega_X$ and an exit place $x_X \in \Omega_X^\circ$, and for every $\Delta_{X,v} \notin \mathcal{X}$, an entry place $e_{X,v} \in {}^\circ\Delta_{X,v}$ and an exit place $x_{X,v} \in \Delta_{X,v}^\circ$. The set of nodes of G is given by:

$$\{\epsilon_X \mid X \in \mathcal{X}\} \cup \{\epsilon_{X,v} \mid X \in \mathcal{X} \wedge v \in e_X^\bullet \cup x_X^\bullet\} \cup \{\epsilon'_{X,v} \mid X \in \mathcal{X} \wedge v \in e_X^\bullet \cup x_X^\bullet \wedge \Delta_{X,v} \notin \mathcal{X}\} \cup$$
$$\{\xi_X \mid X \in \mathcal{X}\} \cup \{\xi_{X,v} \mid X \in \mathcal{X} \wedge v \in {}^\bullet e_X \cup {}^\bullet x_X\} \cup \{\xi'_{X,v} \mid X \in \mathcal{X} \wedge v \in {}^\bullet e_X \cup {}^\bullet x_X \wedge \Delta_{X,v} \notin \mathcal{X}\}.$$

[15] Formally, an $\mathcal{X}$–family is a mapping from $\mathcal{X}$ to some set of objects.

[16] Where the limit of an $\mathcal{X}$–family of $\sqsubseteq$-chains is defined component-wise (and is itself an $\mathcal{X}$–family).

Each node ϵ_X is labelled by e_X, each ξ_X by x_X, each $\epsilon_{X,v}$ and $\xi_{X,v}$ by v, each $\epsilon'_{X,v}$ by $e_{X,v}$ and each $\xi'_{X,v}$ by $x_{X,v}$. For every node ϵ_X there is an arc to $\epsilon_{X,v}$ (for $v \in e_X^\bullet$) and to $\xi_{X,v}$ (for $v \in {}^\bullet e_X$). Similarly, for every node ξ_X there is an arc to $\epsilon_{X,v}$ (for $v \in x_X^\bullet$) and to $\xi_{X,v}$ (for $v \in {}^\bullet x_X$). For every node $\epsilon_{X,v}$ there is an arc to ϵ_Y if $\Delta_{X,v} = Y$ and to $\epsilon'_{X,v}$ if $\Delta_{X,v} \notin \mathcal{X}$ (similarly, for every node $\xi_{X,v}$ there is an arc to ξ_Y if $\Delta_{X,v} = Y$ and to $\xi'_{X,v}$ if $\Delta_{X,v} \notin \mathcal{X}$). There are no arcs outgoing from the nodes $\epsilon'_{X,v}$ and $\xi'_{X,v}$.

Having defined the graph G, for every $X \in \mathcal{X}$, the spine tree $\tau_{e,X}$ (respectively, $\tau_{x,X}$) has as its nodes all finite directed paths ϕ in G starting from the node ϵ_X (respectively, ξ_X), with the label of the last visited node (of G) as its own label, and arcs $\phi \to \phi'$ if and only if $\phi' = \phi \cdot \theta$, for some node θ (of G). One can see that $(\{\breve{\tau}_e\}, \{\breve{\tau}_x\}) \subseteq \breve{\mathbf{f}}(\{\breve{\tau}_e\}, \{\breve{\tau}_x\})$.

Figure 8 shows the graph G which would give rise to the spine trees shown in figure 5. The two different ways of defining spine trees (i.e., that defined in the proof of theorem 16, and the one just described for the case of a single equation) are equivalent, but the one based on the graph G immediately generalises to systems of equations.

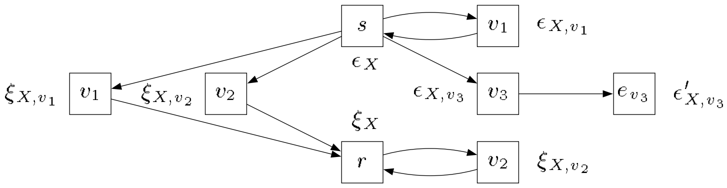


Fig. 8. Graph generating spine trees.

9 Concluding remarks

In this paper we did not address the issue of the behavioural properties of boxes obtained as solutions of recursive systems. This has been done in [7] where it was shown that for a wide class of operator boxes and rec-boxes, solving recursive net equations always yields nets which are safe and satisfy further series of desirable properties.

References

1. J. Baeten and W. P. Weijland: *Process Algebra.* Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press (1990).
2. E. Best and R. Devillers: Sequential and Concurrent Behaviour in Petri Net Theory. *Theoretical Computer Science* 55 (1988) 87-136.
3. E. Best, R. Devillers and J. Esparza: General Refinement and Recursion Operators for the Petri Box Calculus. Proc. of *STACS 93, 10th Annual Symposium on Theoretical Aspects of Computer Science*, P. Enjalbert, A. Finkel and K.W. Wagner (Eds.). Springer-Verlag, Lecture Notes in Computer Science 665 (1993) 130-140.
4. E. Best, R. Devillers and J. Hall: The Petri Box Calculus: a New Causal Algebra with Multilabel Communication. In: *Advances in Petri Nets1992*, G. Rozenberg (Ed.). Springer-Verlag, Lecture Notes in Computer Science 609 (1992) 21-69.
5. E. Best, R. Devillers and M. Koutny: The Box Algebra - a Model of Nets and Process Expressions. Proc. of *ICATPN'99*, S. Donatelli and J. Kleijn. Springer-Verlag, Lecture Notes in Computer Science 1639 (1999) 344-363.
6. E. Best, R. Devillers and M. Koutny: Petri Nets, Process Algebras and Concurrent Programming Languages. In: *Advances in Petri Nets. Lectures on Petri Nets II: Applications*, W. Reisig and G. Rozenberg (Eds.). Springer-Verlag, Lecture Notes in Computer Science 1492 (1998) 1-84.
7. E. Best, R. Devillers and M. Koutny: *Petri Net Algebra (Working Title).* Manuscript (1999).
8. E. Best, R. Devillers and M. Koutny: The Box Algebra = Petri Nets + Process Expressions. Submitted paper (1999).

9. E. Best and M. Koutny: Solving Recursive Net Equations. Proc. of *Automata, Languages and Programming, 22nd International Colloquium, ICALP 95*, Z. Fulöp and F. Gécseg (Ed.). Springer-Verlag, Lecture Notes in Computer Science 944 (1995) 605-623.
10. G. Boudol and I. Castellani: Flow Models of Distributed Computations: Event Structures and Nets. Teachnical Report RR-1484, INRIA, Sophia Antipolis (1991).
11. S. D. Brookes: The Essence of Parallel Algol. Proc. of *LICS'96*, IEEE Computer Society Press (1996) 164-173.
12. J. W. de Bakker: *Operational and Denotational Semantics of Unguarded Recursion.* Manuscript (1996).
13. P. Degano, R. De Nicola and U. Montanari: A Distributed Operational Semantics for CCS Based on C/E Systems. *Acta Informatica* 26 (1988) 59-91.
14. B. A. Davey and H. A. Priestley: *Introduction to Lattices and Order.* Cambridge Mathematical Textbook (1990).
15. R. Devillers: S-invariant Analysis of General Recursive Petri Boxes. *Acta Informatica* 32 (1995) 313-345.
16. R. Devillers and M. Koutny: Recursive Nets in the Box Algebra. Proc. of *CSD'98: International Conference on Application of Concurrency to System Design*, IEEE Press (1998) 239-249.
17. U. Goltz: On Representing CCS Programs by Finite Petri Nets. Proc. of *MFCS'88*, M. P. Chytil, L. Janiga and V. Koubek (Eds.). Springer-Verlag, Lecture Notes in Computer Science 324 (1988) 339-350.
18. U. Goltz and R. Loogen: A Non-interleaving Semantic Model for Nondeterministic Concurrent Processes. *Fundamentae Informaticae* 14 (1991) 39-73.
19. C. A. R. Hoare: *Communicating Sequential Processes.* Prentice Hall (1985).
20. R. Janicki and P. E. Lauer: *Specification and Analysis of Concurrent Systems - the COSY Approach.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1992).
21. M. Koutny and E. Best: Fundamental Study: Operational and Denotational Semantics for the Box Algebra. *Theoretical Computer Science* 211 (1999) 1-83.
22. A. Levy: *Basic Set Theory.* Springer-Verlag (1979).
23. R. Milner: *Communication and Concurrency.* Prentice Hall (1989).
24. T. Murata: Petri Nets: Properties, Analysis and Applications. *Proc. of IEEE* 77 (1989) 541-580.
25. E. R. Olderog: *Nets, Terms and Formulas.* Cambridge Tracts in Theoretical Computer Science 23, Cambridge University Press (1991).
26. G. D. Plotkin: A Structural Approach to Operational Semantics. Teachnical Report FN-19, Computer Science Department, University of Aarhus (1981).
27. W. Reisig: *Petri Nets. An Introduction.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1985).
28. D. Taubner: *Finite Representation of CCS and TCSP Programs by Automata and Petri Nets.* Springer-Verlag, Lecture Notes in Computer Science 369 (1989).