Structure Preserving Bisimilarity, Supporting an Operational Petri Net Semantics of CCSP

Rob J. van Glabbeek[1,2]

[1] NICTA*, Sydney, Australia
[2] Computer Science and Engineering, UNSW, Sydney, Australia

Abstract. In 1987 Ernst-Rüdiger Olderog provided an operational Petri net semantics for a subset of CCSP, the union of Milner's CCS and Hoare's CSP. It assigns to each process term in the subset a labelled, safe place/transition net. To demonstrate the correctness of the approach, Olderog established agreement (1) with the standard interleaving semantics of CCSP up to strong bisimulation equivalence, and (2) with standard denotational interpretations of CCSP operators in terms of Petri nets up to a suitable semantic equivalence that fully respects the causal structure of nets. For the latter he employed a linear-time semantic equivalence, namely having the same causal nets.

This paper strengthens (2), employing a novel branching-time version of this semantics—*structure preserving bisimilarity*—that moreover preserves inevitability. I establish that it is a congruence for the operators of CCSP.

1 Introduction

The system description languages CCS and CSP have converged to one theory of processes which—following a suggestion of M. Nielsen—was called "CCSP" in [26]. The standard semantics of this language is in terms of labelled transition systems modulo strong bisimilarity, or some coarser semantic equivalence. In the case of CCS, a labelled transition system is obtained by taking as states the closed CCS expressions, and as transitions those that are derivable from a collection of rules by induction on the structure of these expressions [24]; this is called a *(structural) operational semantics* [30]. The semantics of CSP was originally given in quite a different way [3,20], but [28] provided an operational semantics of CSP in the same style as the one of CCS, and showed its consistency with the original semantics.

Such semantics abstract from concurrency relations between actions by reducing concurrency to interleaving. An alternative semantics, explicitly modelling concurrency relations, requires models like Petri nets [33] or event structures [25,36]. In [36,21] non-interleaving semantics for variants of CCSP are given in terms of event structures. However, infinite event structures are needed to model simple systems involving loops, whereas Petri nets, like labelled transition systems, offer finite representations for some such systems. Denotational semantics in terms of Petri nets of the essential CCSP operators are given in [18,35,16]—see [27] for more references. Yet a satisfactory denotational Petri net semantics treating recursion has to my knowledge not been proposed.

* NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

Olderog [26,27] closed this gap by giving an operational net semantics in the style of [30,24] for a subset of CCSP including recursion—to be precise: *guarded* recursion. To demonstrate the correctness of his approach, Olderog proposed two fundamental properties such a semantics should have, and established that both of them hold [27]:

- *Retrievability*. The standard interleaving semantics for process terms should be retrievable from the net semantics.
- *Concurrency*. The net semantics should represent the intended concurrency of process terms.

The second requirement was not met by an earlier operational net semantics from [5].

To formalise the first requirement, Olderog notes that a Petri net induces a labelled transition system through the firing relation between markings—the *interleaving case graph*—and requires that the interpretation of any CCSP expression as a state in a labelled transition system through the standard interleaving semantics of CCSP should be strongly bisimilar to the interpretation of this expression as a marking in the interleaving case graph induced by its net semantics.

To formalise the second requirement, he notes that the intended concurrency of process terms is clearly represented in the standard denotational semantics of CCSP operators [18,35,16], and thus requires that the result of applying a CCSP operator to its arguments according to this denotational semantics yields a similar result as doing this according to the new operational semantics. The correct representation of recursion follows from the correct representation of the other operators through the observation that a recursive call has the very same interpretation as a Petri net as its unfolding.

A crucial parameter in this formalisation is the meaning of "similar". A logical choice would be semantic equivalence according to one of the non-interleaving equivalences found in the literature, where a finer or more discriminating semantics gives a stronger result. To match the concurrency requirement, this equivalence should *respect concurrency*, in that it only identifies nets which display the same concurrency relations. In this philosophy, the semantics of a CCSP expression is not so much a Petri net, but a semantic equivalence class of Petri nets, i.e. a Petri net after abstraction from irrelevant differences between nets. For this idea to be entirely consistent, one needs to require that the chosen equivalence is a congruence for all CCSP constructs, so that the meaning of the composition of two systems, both represented as equivalence classes of nets, is independent of the choice of representative Petri nets within these classes.

Instead of selecting such an equivalence, Olderog instantiates "similar" in the above formalisation of the second requirement with *strongly bisimilar*, a new relation between nets that should not be confused with the traditional relation of strong bisimilarity between labelled transition systems. As shown in [1], strong bisimilarity fails to be an equivalence: it is reflexive and symmetric, but not transitive.

As pointed out in [27, Page 37] this general shortcoming of strong bisimilarity "does not affect the purpose of this relation" in that book: there it "serves as an auxiliary notion in proving that structurally different nets are causally equivalent". Here *causal equivalence* means having the same causal nets, where *causal nets* [29,34] model concurrent computations or executions of Petri nets. So in effect Olderog does choose a semantic equivalence on Petri nets, namely having the same concurrent computations as modelled by causal nets. This equivalence fully respects concurrency.

1.1 Structure preserving bisimilarity

The contribution of the present paper is a strengthening of this choice of a semantic equivalence on Petri nets. I propose the novel *structure preserving bisimulation* equivalence on Petri nets, and establish that the result of applying a CCSP operator to its arguments according to the standard denotational semantics yields a structure preserving bisimilar result as doing this according to Olderog's operational semantics. The latter is an immediate consequence of the observation that structure preserving bisimilarity between two nets is implied by Olderog's strong bisimilarity.

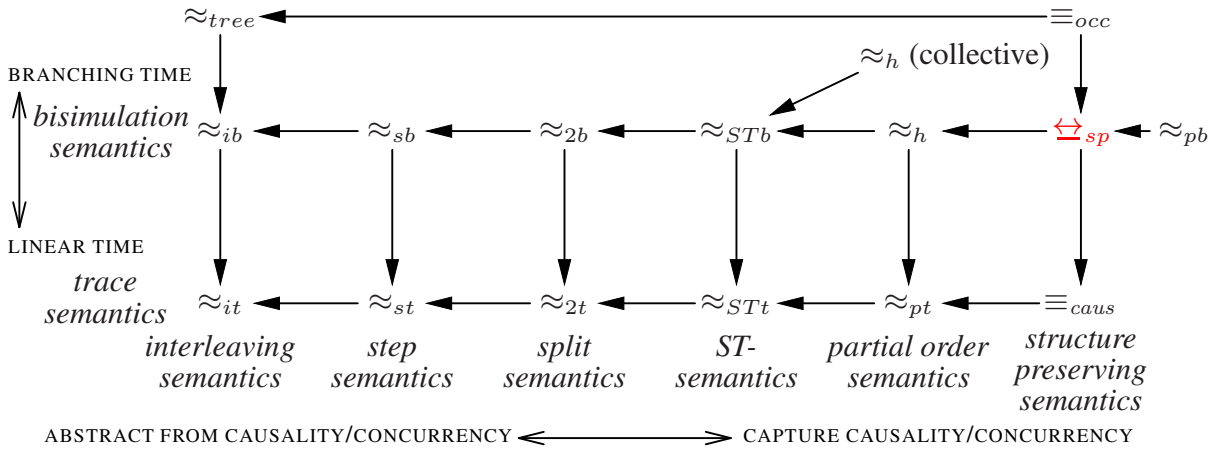


Fig. 1. A spectrum of semantic equivalences on Petri nets

Figure 1 shows a map of some equivalence relations on nets found in the literature, in relation to the new structure preserving bisimilarity, $\leftrightarrows_{sp}$. The equivalences become finer when moving up or to the right; thus coarser or less discriminating when following the arrows. The rectangle from $\approx_{it}$ to $\approx_h$ is taken from [10]. The vertical axis is the *linear time – branching time spectrum*, with *trace equivalence* at the bottom and *(strong) bisimulation equivalence*, or *bisimilarity*, at the top. A host of intermediate equivalences is discussed in [11]. The key difference is that *linear time* equivalences, like trace equivalence, only consider the set of possible executions of a process, whereas *branching time* equivalences, like bisimilarity, additionally take into account at which point the choice between two executions is made. The horizontal axis indicates to what extent concurrency information is taken into account. *Interleaving* equivalences—on the left—fully abstract from concurrency by reducing it to arbitrary interleaving; *step* equivalences additionally take into account the possibility that two concurrent actions happen at exactly the same moment; *split* equivalences recognise the beginning and end of actions, which here are regarded to be durational, thereby capturing some information about their overlap in time; *ST-* or *interval* equivalences fully capture concurrency information as far as possible by considering durational actions overlapping in time; and *partial order* equivalences capture the causal links between actions, and thereby all concurrency. By taking the product of these two axes, one obtains a two-dimensional spectrum of equivalence relations, with entries like *interleaving bisimulation* equivalence $\approx_{ib}$ and *partial order trace* equivalence $\approx_{pt}$. For the right upper corner several partial order bisimulation equivalences were proposed in the literature; according to [13]

the *history preserving bisimulation* equivalence $\approx_h$, originally proposed by [32], is the coarsest one that fully captures the interplay between causality and branching time.

The causal equivalence employed by Olderog, $\equiv_{caus}$, is a linear time equivalence strictly finer than $\approx_{pt}$. Since it preserves information about the number of preplaces of a transition, it is specific to a model of concurrency based on Petri nets; i.e. there is no obvious counterpart in terms of event structures. I found only two equivalences in the literature that are finer than both $\equiv_{caus}$ and $\approx_h$, namely *occurrence net equivalence* [16]—$\equiv_{occ}$—and the *place bisimilarity* $\approx_{pb}$ of [1]. Two nets are occurrence net equivalent iff they have isomorphic unfoldings. The *unfolding*, defined in [25], associates with a given safe Petri net N a loop-free net—an *occurrence net*—that combines all causal nets of N, together with their branching structure. This unfolding is similar to the unfolding of a labelled transition system into a tree, and thus the interleaving counterpart of occurrence net equivalence is *tree equivalence* [11], identifying two transition systems iff their unfoldings are isomorphic. The place bisimilarity was inspired by Olderog's strong bisimilarity, but adapted to make it transitive, and thus an equivalence relation. My new equivalence $\leftrightarrows_{sp}$ will be shown to be strictly coarser than $\equiv_{occ}$ and $\approx_{pb}$, yet finer than both $\equiv_{caus}$ and $\approx_h$.

The equivalences discussed above (without the diagonal line in Figure 1) are all defined on safe Petri nets. Additionally, the definitions generalise to unsafe Petri nets. However, there are two possible interpretations of unsafe Petri nets, called the *collective token* and the *individual token* interpretation [12], and this leads to two versions of history preserving bisimilarity. The history preserving bisimilarity based on the individual token interpretation was first defined for Petri nets in [2], under the name *fully concurrent bisimulation* equivalence. At the level of ST-semantics the collective and individual token interpretations collapse. The unfolding of unsafe Petri nets, and thereby occurrence net equivalence, has been defined for the individual token interpretation only [7,23,12], and likewise causal equivalence can be easily generalised within the individual token interpretation. The new structure preserving bisimilarity falls in the individual token camp as well.

1.2 Criteria for choosing this semantic equivalence

In selecting a new semantic equivalence for reestablishing Olderog's agreement of operational and denotational interpretations of CCSP operators, I consider the following requirements on such a semantic equivalence (with subsequent justifications):

1. it should be a branching time equivalence,
2. it should fully capture causality relations and concurrency (and the interplay between causality and branching time),
3. it should respect *inevitability* [22], meaning that if two systems are equivalent, and in one the occurrence of a certain action is inevitable, then so is it in the other,
4. it should be *real-time consistent* [16], meaning that for every association of execution times to actions, assuming that actions happen as soon as they can, the running times associated with computations in equivalent systems should be the same,
5. it should be *preserved under action refinement* [4,13], meaning that if in two equivalent Petri nets the same substitutions of nets for actions are made, the resulting nets should again be equivalent,

6. it should be finer than Olderog's causal equivalence,
7. it should not distinguish systems whose behaviours are patently the same, such as Petri nets that differ only in their unreachable parts,
8. it should be a congruence for the constructs of CCSP,
9. and it should allow to establish agreement between the operational and denotational interpretations of CCSP operators.

Requirement 1 is the driving force behind this contribution. It is motivated by the insight that branching time equivalences better capture phenomena like deadlock behaviour. Since in general a stronger result on the agreement between operational and denotational semantics is obtained when employing a finer semantics, I aim for a semantics that fully captures branching time information, and thus is at least as discriminating as interleaving bisimilarity.

Requirement 2 is an obvious choice when the goal of the project is to capture concurrency explicitly. The combination of Requirements 1 and 2 then naturally asks for an equivalence that is at least as fine as $\approx_h$. One might wonder, however, for what reason one bothers to define a semantics that captures concurrency information. In the literature, various practical reasons have been given for preferring a semantics that (partly) respects concurrency and causality over an interleaving semantics. Three of the more prominent of these reasons are formulated as requirements 3, 4 and 5 above.

Requirement 3 is manifestly useful when considering liveness properties of systems. Requirement 4 obviously has some merit when timing is an issue. Requirement 5 is useful in system design based on stepwise refinement [13].

Requirement 6 is only there so that I can truthfully state to have strengthened Olderog's agreement between the denotational and operational semantics, which was stated in terms of causal equivalence. This requirement will not be needed in my justification for introducing a new semantic equivalence—and neither will Requirement 2.

Requirement 7 is hardly in need of justification. The paper [1] lists as a desirable property of semantic equivalences—one that is not met by their own proposal $\approx_{pb}$—that they should not distinguish nets that have isomorphic unfoldings, given that unfolding a net should not be regarded as changing it behaviour. When working within the individual token interpretation of nets I will take this as a suitable formalisation of Requirement 7.

The argument for Requirement 8 has been given earlier in this introduction, and Requirement 9 underlies my main motivation for selecting a semantic equivalence in the first place.

1.3 Applying the criteria

Table 1 tells which of these requirements are satisfied by the semantic equivalences from Section 1.1 (not considering the one collective token equivalence there). The first two rows, reporting which equivalences satisfy Requirements 1 and 2, are well-known; these results follow directly from the definitions. The third row, reporting on respect for inevitability, is a contribution of this paper, and will be discussed in Section 1.4, and delivered in Sections 11–14.

Table 1. Which requirements are satisfied by the various semantic equivalences

Equivalence			$\approx_{tree}$									$\equiv_{occ}$		$\approx_{pb}$
		$\approx_{ib}$			$\approx_{sb}$		$\approx_{2b}$		$\approx_{STb}$		$\approx_{h}$	$\leftrightarrows_{sp}$		
Requirement	$\approx_{it}$			$\approx_{st}$		$\approx_{2t}$		$\approx_{STt}$		$\approx_{pt}$		$\equiv_{caus}$		
1. Branching time	×	✓	✓	×	✓	×	✓	×	✓	×	✓	×	✓	✓	✓
2. Causality	×	×	×	×	×	×	×	×	×	✓	✓	✓	✓	✓	✓
3. Inevitability	×	×	×	×	×	×	×	×	×	×	×	×	✓	✓	✓
4. Real-time consistency	×	×	×	×	×	×	×	×	✓	×	✓	×	✓	✓	✓
5. Action refinement	×	×	×	×	×	×	×	✓	✓	✓	✓	✓?	✓?	✓?	
6. Finer than $\equiv_{caus}$	×	×	×	×	×	×	×	×	×	×	×	✓	✓	✓	✓
7. Coarser than $\equiv_{occ}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
8. Congruence	✓	✓											✓		
9. Operat. $\equiv$ denotat.	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	

Regarding Row 4, In [16] it is established that ST-bisimilarity is real-time consistent. Moreover, the formal definition is such that if a semantic equivalence $\approx$ is real-time consistent, then so is any equivalence finer than $\approx$. Linear time equivalences are not real-time consistent, and neither is $\approx_{2b}$ [17].

In [13] it is established that $\approx_{pt}$ and $\approx_{h}$ are preserved under action refinement, but interleaving and step equivalences are not, because they do not capture enough information about concurrency. In [10] it is shown that $\approx_{STt}$ and $\approx_{STb}$ are already preserved under action refinement, whereas by [17] split semantics are not. I conjecture that $\equiv_{caus}$ and $\equiv_{occ}$ are also preserved under action refinement, but I have not seen a formal proof. I also conjecture that the new $\leftrightarrows_{sp}$ is preserved under action refinement.

Rows 6 and 7 follow as soon as I have formally established the implications of Figure 1 (in Section 10). As for Row 8, I will show in Section 7 that $\leftrightarrows_{sp}$ is a congruence for the operators of CCSP. That also $\approx_{it}$ and $\approx_{ib}$ are congruences for CCSP is well known. The positive results in Row 9 follow from the fact that Olderog's strong bisimilarity implies $\leftrightarrows_{sp}$, which will be established in Section 6.

Requirements 1 and 6 together limit the search space for suitable equivalence relations to $\equiv_{occ}$, $\approx_{pb}$ and the new $\leftrightarrows_{sp}$. When dropping Requirement 6, but keeping 2, also $\approx_{h}$ becomes in scope. When also dropping 2, but keeping 4, I gain $\approx_{STb}$ as a candidate equivalence. However, both $\approx_{h}$ and $\approx_{STb}$ will fall pray to Requirement 3, so also without Requirements 2 and 6 the search space will be limited to $\equiv_{occ}$, $\approx_{pb}$ and the new $\leftrightarrows_{sp}$.

Requirement 7 rules out $\approx_{pb}$, as that equivalence makes distinctions based on unreachable parts of nets [1]. The indispensable Requirement 9 rules out $\equiv_{occ}$, since that equivalence distinguishes the operational and denotational semantics of the CCSP expression $a0 + a0$. According to the operational semantics this expression has only one transition, whereas by the denotational semantics it has two, and $\equiv_{occ}$ does not collapse identical choices. The same issue plays in interleaving semantics, where the operational and denotational transition system semantics of CCSP do not agree up to tree equivalence. This is one of the main reasons that bisimilarity is often regarded as the top of the linear time – branching time spectrum.

This constitutes the justification for the new equivalence $\leftrightarrows_{sp}$.

1.4 Inevitability

The meaning of Requirement 3 depends on which type of progress or fairness property one assumes to guarantee that actions that are due to occur will actually happen. Lots of fairness assumption are mentioned in the literature, but, as far as I can tell, they can be classified in exactly 4 groups: *progress*, *justness*, *weak fairness* and *strong fairness* [15]. These four groups form a hierarchy, in the sense that one cannot consistently assume strong fairness while objecting to weak fairness, or justness while objecting to progress.

Strong and weak fairness deal with choices that are offered infinitely often. Suppose you have a shop with only two customers A and B that may return to the shop to buy something else right after they are served. Then it is unfair to only serve customer A again and again, while B is continuously waiting to be served. In case B is not continuously ready to be served, but sometimes goes home to sleep, yet always returns to wait for his turn, it is weakly fair to always ignore customer B in favour of A, but not strongly fair.

Weak and strong fairness assumptions can be made *locally*, pertaining to *some* repeating choices of the modelled system but not to others, or *globally*, pertaining to all choices of a given type. Since the real world is largely unfair, strong and weak fairness assumptions need to be made with great caution, and they will not appear in this paper.

Justness and progress assumptions, on the other hand, come only in the global variant, and can be safely assumed much more often. A progress assumption says that if a system can do some action (that is not contingent on external input) it will do an action. In the example of the shop, if there is a customer continuously ready to be served, and the clerk stands pathetically behind the counter staring at the customer but not serving anyone, there is a failure of progress. Without assuming progress, no action is inevitable, because it is always possible that a system will remain in its initial state without ever doing anything. Hence the concept of inevitability only makes sense when assuming at least progress.

Justness [8,15] says roughly that if a parallel component can make progress (not contingent on input from outside of this component) it will do so. Suppose the shop has two counters, each manned by a clerk, and, whereas customer A is repeatedly served at counter 1, customer B is ready to be served by counter 2, but is only stared at by a pathetic clerk. This is not a failure of progress, as in any state of the system someone will be served eventually. Yet it counts as a failure of justness. In the context of Petri nets, a failure of justness can easily be formalised as an execution, during which, from some point onwards, all preplaces of a given transition remain marked, yet the transition never fires [14]. One could argue that, when taking concurrency seriously, justness should be assumed whenever one assumes progress.

Inevitability can be easily expressed in temporal logics like LTL [31] or CTL [6], and it is well known that strongly bisimilar transition systems satisfy the same temporal formulas. This suggests that interleaving bisimilarity already respects inevitability. However, this conclusion is warranted only when assuming progress but not justness, or perhaps also when assuming some form of weak or strong fairness. The system $C := \langle X|X = aX + bX\rangle$—using the CCSP syntax of Section 2—repeatedly choosing between the actions a and b, is interleaving bisimilar to the system $D := \langle Y|Y = aY\rangle \| \langle Z|Z = bZ\rangle$, which in parallel performs infinitely many as and infinitely

many bs. Yet, when assuming justness but not weak fairness, the execution of the action b is inevitable in D, but not in C. This shows that when assuming justness but not weak fairness, interleaving bisimilarity does not respect inevitability. The paper [22], which doesn't use Petri nets as system model, leaves the precise formulation of a justness assumption for future work—this task is undertaken in the different context of CCS in [15]. Also, respect of inevitability as a criterion for judging semantic equivalences does not occur in [22], even though "the partial order approach" is shown to be beneficial.

In this paper, assuming justness but not strong or weak fairness, I show that neither $\approx_h$ nor $\equiv_{caus}$ respects inevitability (using infinite nets in my counterexample). Hence, respecting concurrency appears not quite enough to respect inevitability. Respect for inevitability, like real-time consistency, is a property that holds for any equivalence relation finer than one for which it is known to hold already. So also none of the ST- or interleaving equivalences respects inevitability. I show that the new equivalence $\leftrightarrows_{sp}$ respects inevitability. This makes it the coarsest equivalence of Figure 1 that does so.

2 CCSP

CCSP is parametrised by the choice of an infinite set Act of actions, that I will assume to be fixed for this paper. Just like the version of CSP from Hoare [20], the version of CCSP used here is a typed language, in the sense that with every CCSP process P an explicit alphabet $\alpha(P) \subseteq Act$ is associated, which is a superset of the set of all actions the process could possibly perform. This alphabet is exploited in the definition of the parallel composition $P\|Q$: actions in the intersection of the alphabets of P and Q are required to synchronise, whereas all other actions of P and Q happen independently. Because of this, processes with different alphabets may never be identified, even if they can perform the same set of actions and are alike in all other aspects. It is for this reason that I interpret CCSP in terms of *typed* Petri nets, with an alphabet as extra component.

I also assume an infinite set V of *variable names*. A *variable* is a pair X_A with $X \in V$ and $A \subseteq Act$. The syntax of (my subset of) CCSP is given by

$$P ::= 0_A \mid aP \mid P + P \mid P\|P \mid R(P) \mid X_A \mid \langle X_A|\mathcal{S}\rangle \text{ (with } X_A \in V_{\mathcal{S}})$$

with $A \subseteq Act$, $a \in Act$, $R \subseteq Act \times Act$, $X \in V$ and $\mathcal{S}$ a *recursive specification*: a set of equations $\{Y_B = \mathcal{S}_{Y_B} \mid Y_B \in V_{\mathcal{S}}\}$ with $V_{\mathcal{S}} \subseteq V \times Act$ (the *bound variables* of $\mathcal{S}$) and $\mathcal{S}_{Y_B}$ a CCSP expression satisfying $\alpha(\mathcal{S}_{Y_B}) = B$ for all $Y_B \in V_{\mathcal{S}}$ (were $\alpha(\mathcal{S}_{Y_B})$ is defined below). The constant 0_A represents a process that is unable to perform any action. The process aP first performs the action a and then proceeds as P. The process $P + Q$ will behave as either P or Q, $\|$ is a partially synchronous parallel composition operator, R a renaming, and $\langle X_A|\mathcal{S}\rangle$ represents the X_A-component of a solution of the system of recursive equations $\mathcal{S}$. A CCSP expression P is *closed* if every occurrence of a variable X_A occurs in a subexpression $\langle Y_B|\mathcal{S}\rangle$ of P with $X_A \in V_{\mathcal{S}}$.

The constant 0 and the variables are indexed with an alphabet. The alphabet of an arbitrary CCSP expression is given by:

- $\alpha(0_A) = \alpha(X_A) = \alpha(\langle X_A|\mathcal{S}\rangle) = A$
- $\alpha(aP) = \{a\} \cup \alpha(P)$

Table 2. Structural operational interleaving semantics of CCSP

$$aP \xrightarrow{a} P \qquad \frac{P \xrightarrow{a} P'}{P\|Q \xrightarrow{a} P'\|Q}\ (a \notin \alpha(Q)) \qquad \frac{P \xrightarrow{a} P'}{R(P) \xrightarrow{b} R(P')}\ ((a,b) \in R)$$

$$\frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \qquad \frac{P \xrightarrow{a} P',\ Q \xrightarrow{a} Q'}{P\|Q \xrightarrow{a} P'\|Q'}\ (a \in \alpha(P) \cap \alpha(Q))$$

$$\frac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'} \qquad \frac{Q \xrightarrow{a} Q'}{P\|Q \xrightarrow{a} P\|Q'}\ (a \notin \alpha(P)) \qquad \frac{\langle \mathcal{S}_{X_A} | \mathcal{S} \rangle \xrightarrow{a} P'}{\langle X_A | \mathcal{S} \rangle \xrightarrow{a} P'}$$

- $\alpha(P + Q) = \alpha(P\|Q) = \alpha(P) \cup \alpha(Q)$
- $\alpha(R(P)) = \{b \mid \exists a \in \alpha(P) : (a,b) \in R\}$.

Substitutions of expressions for variables are allowed only if the alphabets match. For this reason a recursive specification $\mathcal{S}$ is declared syntactically incorrect if $\alpha(\mathcal{S}_{Y_B}) \neq B$ for some $Y_B \in V_{\mathcal{S}}$. The interleaving semantics of CCSP is given by the labelled transition relation $\rightarrow \subseteq \mathrm{T}_{\mathrm{CCSP}} \times Act \times \mathrm{T}_{\mathrm{CCSP}}$ on the set $\mathrm{T}_{\mathrm{CCSP}}$ of closed CCSP terms, where the transitions $P \xrightarrow{a} Q$ (on arbitrary CCSP expressions) are derived from the rules of Table 2. Here $\langle P|\mathcal{S}\rangle$ for P an expression and $\mathcal{S}$ a recursive specification denotes the expression P in which $\langle Y_B|\mathcal{S}_{Y_B}\rangle$ has been substituted for the variable Y_B for all $Y_B \in V_{\mathcal{S}}$.

A CCSP expression is *well-typed* if for any subexpression of the form aP one has $a \in \alpha(P)$ and for any subexpression of the form $P + Q$ one has $\alpha(P) = \alpha(Q)$. Thus $a0_{\{a\}} + bX_{\emptyset}$ is not well-typed, although the equivalent expression $a0_{\{a,b\}} + bX_{\{a,b\}}$ is. A recursive specification $\langle X_A|\mathcal{S}\rangle$ is *guarded* if each occurrence of a variable $Y_B \in V_{\mathcal{S}}$ in a term $\mathcal{S}_{Z_C}$ for some $Z_C \in V_{\mathcal{S}}$ lays within a subterm of $\mathcal{S}_{Z_C}$ of the form aP. Following [27] I henceforth only consider well-typed CCSP expressions with guarded recursion.

In Olderog's subset of CCSP, each recursive specification has only one equation, and renamings must be functions instead of relations. Here I allow mutual recursion and relational renaming, where an action may be renamed into a choice of several actions—or possibly none. This generalisation does not affect any of the proofs in [27].

Example 1. The behaviour of the customer from Section 1.4 could be given by the recursive specification $\mathcal{S}_{\mathrm{CUS}}$:

$$\mathrm{CUS}_{Cu} = \textit{enter buy leave}\ \mathrm{CUS}_{Cu}$$

indicating that the customer keeps coming back to the shop to buy more things. Here *enter*, *buy*, *leave* $\in Act$ and $\mathrm{CUS} \in V$. The customer's alphabet Cu is $\{\textit{enter}, \textit{buy}, \textit{leave}\}$. Likewise, the behaviour of the store clerk could be given by the specification $\mathcal{S}_{\mathrm{CLK}}$:

$$\mathrm{CLK}_{Cl} = \textit{serve}\ \mathrm{CLK}_{Cl}$$

where $Cl = \{\textit{serve}\}$. The CCSP processes representing the customer and the clerk, with their reachable states and labelled transitions between them, are displayed in Figure 2.

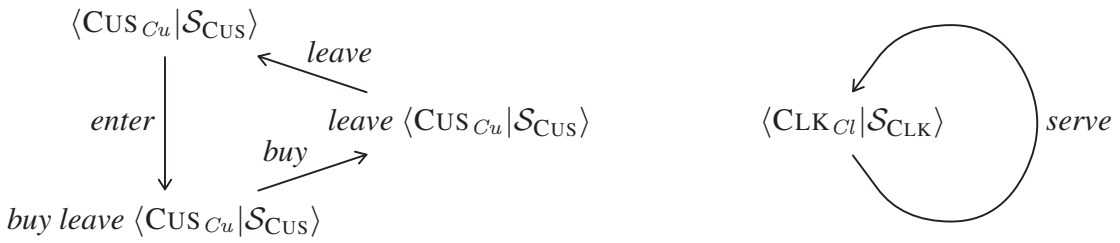


Fig. 2. Labelled transition semantics of customer and clerk

In order to ensure that the parallel composition synchronises the *buy*-action of the customer with the *serve*-action of the clerk, I apply renaming operators R_{CUS} and R_{CLK} with $R_{\text{CUS}}(\mathit{buy}) = \mathit{serves}$ and $R_{\text{CLK}}(\mathit{serve}) = \mathit{serves}$ and leaving all other actions unchanged, where *serves* is a joint action of the renamed customer and the renamed clerk. The total CCSP specification of a store with one clerk and one customer is

$$R_{\text{CUS}}(\langle \text{CUS}_{Cu} | \mathcal{S}_{\text{CUS}} \rangle) \| R_{\text{CLK}}(\langle \text{CLK}_{Cl} | \mathcal{S}_{\text{CLK}} \rangle)$$

and the relevant part of the labelled transition system of CCSP is displayed below.

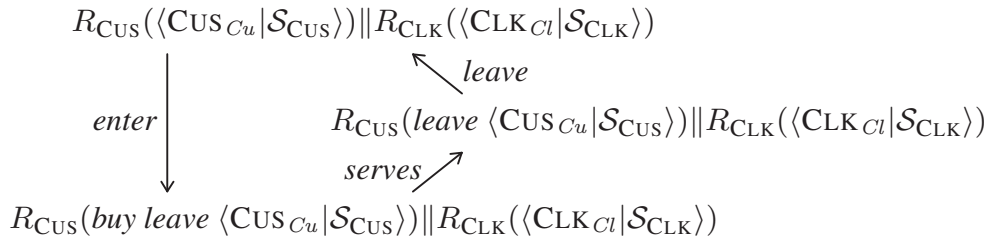


Fig. 3. Labelled transition semantics of the 1-customer 1-clerk store

One possible behaviour of this system is the sequence of actions *enter serves leave enter*, followed by eternal stagnation. This behaviour is ruled out by the progress assumption of Section 1.4. The only behaviour compatible with this assumption is the infinite sequence of actions $(\mathit{enter\ serves\ leave})^\infty$.

To model a store with two customers (A and B) and 2 clerks (I and II), I introduce a relational renaming for each of them, defined by

$$\begin{array}{lll} R_A(\mathit{enter}) = A\,\mathit{enters} & R_A(\mathit{buy}) = \{\text{I}\,\mathit{serves}\,A, \text{II}\,\mathit{serves}\,A\} & R_A(\mathit{leave}) = A\,\mathit{leaves} \\ R_B(\mathit{enter}) = B\,\mathit{enters} & R_B(\mathit{buy}) = \{\text{I}\,\mathit{serves}\,B, \text{II}\,\mathit{serves}\,B\} & R_B(\mathit{leave}) = B\,\mathit{leaves} \\ & R_{\text{I}}(\mathit{serve}) = \{\text{I}\,\mathit{serves}\,A, \text{I}\,\mathit{serves}\,B\} & \\ & R_{\text{II}}(\mathit{serve}) = \{\text{II}\,\mathit{serves}\,A, \text{II}\,\mathit{serves}\,B\}. & \end{array}$$

The CCSP specification of a store with two clerks and two customers is

$$\big(R_A(\langle\!\langle \text{CUS}_{Cu} | \mathcal{S}_{\text{CUS}} \rangle\!\rangle) \| R_B(\langle\!\langle \text{CUS}_{Cu} | \mathcal{S}_{\text{CUS}} \rangle\!\rangle)\big) \| \big(R_{\text{I}}(\langle\!\langle \text{CLK}_{Cl} | \mathcal{S}_{\text{CLK}} \rangle\!\rangle) \| R_{\text{II}}(\langle\!\langle \text{CLK}_{Cl} | \mathcal{S}_{\text{CLK}} \rangle\!\rangle)\big)$$

and the part of the labelled transition system of CCSP reachable from that process has $3 \times 3 \times 1 \times 1 = 9$ states and $6 \times 4 = 24$ transitions.

3 Petri nets

A *multiset* over a set S is a function $C\colon S \to \mathbb{N}$, i.e. $C \in \mathbb{N}^S$; let $|C| := \sum_{x\in X} C(x)$; $x \in S$ is an *element of* C, notation $x \in C$, iff $C(x) > 0$.
The function $\emptyset\colon S \to \mathbb{N}$, given by $\emptyset(x) := 0$ for all $x \in S$, is the *empty* multiset over S.
For multisets C and D over S one writes $C \leq D$ iff $C(x) \leq D(x)$ for all $x \in S$;
$C \cap D$ denotes the multiset over S with $(C \cap D)(x) := \min(C(x), D(x))$,
$C + D$ denotes the multiset over S with $(C + D)(x) := C(x) + D(x)$; and
the multiset $C - D$ is only defined if $D \leq C$ and then $(C - D)(x) := C(x) - D(x)$.
A multiset C with $C(x) \leq 1$ for all x is identified with the (plain) set $\{x \mid C(x) = 1\}$.
The construction $C := \{f(x_1, ..., x_n) \mid x_i \in D_i\}$ of a set C out of sets D_i $(i = 1, ..., n)$ generalises naturally to multisets C and D_i, taking the multiplicity $C(x)$ of an element x to be $\sum_{f(x_1,...,x_n)=x} D_1(x_1) \cdot ... \cdot D_n(x_n)$.

Definition 1. A (*typed*) *Petri net* is a tuple $N = (S, T, F, M_0, A, \ell)$ with
- S and T disjoint sets (of *places* and *transitions*),
- $F : ((S \times T) \cup (T \times S)) \to \mathbb{N}$ (the *flow relation* including *arc weights*),
- $M_0 : S \to \mathbb{N}$ (the *initial marking*),
- A a set of *actions*, the *type* of the net, and
- $\ell : T \to A$ (the *labelling function*).

Petri nets are depicted by drawing the places as circles and the transitions as boxes, containing their label. Identities of places and transitions are displayed next to the net element. For $x, y \in S \cup T$ there are $F(x, y)$ arrows (*arcs*) from x to y. When a Petri net represents a concurrent system, a global state of this system is given as a *marking*, a multiset M of places, depicted by placing $M(s)$ dots (*tokens*) in each place s. The initial state is M_0.

The behaviour of a Petri net is defined by the possible moves between markings M and M', which take place when a transition t *fires*. In that case, t consumes $F(s, t)$ tokens from each place s. Naturally, this can happen only if M makes all these tokens available in the first place. Moreover, t produces $F(t, s)$ tokens in each place s. Definition 2 formalises this notion of behaviour.

Definition 2. Let $N = (S, T, F, M_0, A, \ell)$ be a Petri net and $x \in S \cup T$. The multisets ${}^\bullet x,\ x^\bullet\colon S{\cup}T \to \mathbb{N}$ are given by ${}^\bullet x(y) = F(y, x)$ and $x^\bullet(y) = F(x, y)$ for all $y{\in}S{\cup}T$; for $t \in T$, the elements of ${}^\bullet t$ and $t^\bullet$ are called *pre-* and *postplaces* of t, respectively. Transition $t \in T$ is *enabled* from the marking $M \in \mathbb{N}^S$—notation $M[t\rangle$—if ${}^\bullet t \leq M$. In that case firing t yields the marking $M' := M - {}^\bullet t + t^\bullet$—notation $M[t\rangle M'$.

A *path* π of a Petri net N is an alternating sequence $M_0 t_1 M_1 t_2 M_2 t_3 \ldots$ of markings and transitions, starting from the initial marking M_0 and either being infinite or ending in a marking M_n, such that $M_k[t_k\rangle M_{k+1}$ for all $k\,(<n)$. A marking is *reachable* if it occurs in such a path. The Petri net N is *safe* if all reachable markings M are plain sets, meaning that $M(s) \leq 1$ for all places s. It has *bounded parallelism* [16] if there is no reachable marking M and infinite multiset of transitions U such that $\sum_{t\in U} {}^\bullet t \leq M$. In this paper I consider Petri nets with bounded parallelism only, and call them *nets*.

4 An operational Petri net semantics of CCSP

This section recalls the operational Petri net semantics of CCSP, given by Olderog [26,27]. It associates a net $[\![P]\!]$ with each closed CCSP expression P.

The standard operational semantics of CCSP, presented in Section 2, yields one big labelled transition system for the entire language.[1] Each individual closed CCSP expression P appears as a state in this LTS. If desired, a *process graph*—an LTS enriched with an initial state—for P can be extracted from this system-wide LTS by appointing P as the initial state, and optionally deleting all states and transitions not reachable from P. In the same vein, an operational Petri net semantics yields one big Petri net for the entire language, but without an initial marking. I call such a Petri net *unmarked*. Each process $P \in \mathrm{T}_{\mathrm{CCSP}}$ corresponds with a marking $dex(P)$ of that net. If desired, a Petri net $[\![P]\!]$ for P can be extracted from this system-wide net by appointing $dex(P)$ as its initial marking, taking the type of $[\![P]\!]$ to be $\alpha(P)$, and optionally deleting all places and transitions not reachable from $dex(P)$.

The set S_{CCSP} of places in the net is the smallest set including:

0_A	*inaction*	aP	*prefixing*	$\mu + \nu$	*choice*
$\mu\|\|_A$	*left parallel component*	${}_A\|\|\mu$	*right component*	$R(\mu)$	*renaming*

for $A \subseteq Act$, $P \in \mathrm{T}_{\mathrm{CCSP}}$, $a \in Act$, $\mu, \nu \in S_{\mathrm{CCSP}}$ and renamings R. The mapping $dex : \mathrm{T}_{\mathrm{CCSP}} \to \mathscr{P}(S_{\mathrm{CCSP}})$ decomposing and expanding a process expression into a set of places is inductively defined by:

$$\begin{array}{lcllcl} dex(0_A) & = & \{0_A\} & & & \\ dex(aP) & = & \{aP\} & dex(R(P)) & = & R(dex(P)) \\ dex(P+Q) & = & dex(P) + dex(Q) & dex(\langle X_A|\mathcal{S}\rangle) & = & dex(\langle \mathcal{S}_{X_A}|\mathcal{S}\rangle) \\ dex(P\|Q) & = & dex(P)\|_A \cup {}_A\|dex(Q) \text{ where } A = \alpha(P) \cap \alpha(Q). & & & \end{array}$$

Here $H\|_A$, ${}_A\|H$, $R(H)$ and $H+K$ for $H, K \subseteq S_{\mathrm{CCSP}}$ are defined element by element; e.g. $R(H) = \{R(\mu) \mid \mu \in H\}$. The binding matters, so that $({}_A\|H)\|_B \neq {}_A\|(H\|_B)$. Since I deal with guarded recursion only, dex is well-defined.

Following [27], I construct the unmarked Petri net (S, T, F, Act, ℓ) of CCSP with $S := S_{\mathrm{CCSP}}$, specifying the triple (T, F, ℓ) as a ternary relation $\to \subseteq \mathbb{N}^S \times Act \times \mathbb{N}^S$. An element $H \xrightarrow{a} J$ of this relation denotes a transition $t \in T$ with $\ell(t) = a$ such that ${}^\bullet t = H$ and $t^\bullet = J$. The transitions $H \xrightarrow{\alpha} J$ are derived from the rules of Table 3.

Note that there is no rule for recursion. The transitions of a recursive process $\langle X_A|\mathcal{S}\rangle$ are taken care of indirectly by the decomposition $dex(\langle X_A|\mathcal{S}\rangle) = dex(\langle \mathcal{S}_{X_A}|\mathcal{S}\rangle)$, which expands the decomposition of a recursive call into a decomposition of an expression in which each recursive call is guarded by an action prefix.

Example 2. The Petri net semantics of the 2-customer 2-clerk store from Section 2 is displayed in Figure 4. It is more compact than the 9-state 24-transition labelled transition system. The name of the bottom-most place is ${}_{Ser}\|\,\emptyset\|\, R_{\mathrm{II}}(\mathit{serve}\,\langle \mathrm{CLK}_{Cl}|\mathcal{S}_{\mathrm{CLK}}\rangle)$ where Ser is the alphabet $\{\mathrm{I}\,\mathit{serves}\,A, \mathrm{I}\,\mathit{serves}\,B, \mathrm{II}\,\mathit{serves}\,A, \mathrm{II}\,\mathit{serves}\,B\}$.

[1] A *labelled transition system* (LTS) is given by a set S of *states* and a *transition relation* $T \subseteq S \times \mathscr{L} \times S$ for some set of labels $\mathscr{L}$. The LTS generated by CCSP has $S := \mathrm{T}_{\mathrm{CCSP}}$, $\mathscr{L} := Act$ and $T := \to$.

Table 3. Operational Petri net semantics of CCSP

$$\{aP\} \xrightarrow{a} dex(P)$$

$$\frac{H \xrightarrow{a} J}{R(H) \xrightarrow{b} R(J)} \ ((a,b) \in R) \qquad \frac{H \xrightarrow{a} J}{H\|_A \xrightarrow{a} J\|_A} \ (a \notin A)$$

$$\frac{H \dot\cup K \xrightarrow{a} J}{H \cup (K + dex(Q)) \xrightarrow{a} J} \qquad \frac{H \xrightarrow{a} J \qquad K \xrightarrow{a} L}{H\|_A \cup {}_A\|K \xrightarrow{a} J\|_A \cup {}_A\|L} \ (a \in A)$$

$$\frac{H \dot\cup K \xrightarrow{a} J}{H \cup (dex(P) + K) \xrightarrow{a} J} \qquad \frac{H \xrightarrow{a} J}{{}_A\|H \xrightarrow{a} {}_A\|J} \ (a \notin A)$$

A progress assumption, as discussed in Section 1.4, disallows runs that stop after finitely many actions. So in each run some of the actions from *Ser* will occur infinitely often. When assuming strong fairness, each of those actions will occur infinitely often. When assuming only weak fairness, it is possible that II *serves* A and II *serves* B will never occur, as long as I *serves* A and I *serves* B each occur infinitely often, for in such a run the actions II *serves* A and II *serves* B are not enabled in every state (from some point onwards). However, it is not possible that I *serves* B and II *serves* B never occur, because in such a run, from some point onwards, the action I *serves* B is enabled in every state.

When assuming justness but not weak fairness, a run that bypasses any two serving actions is possible, but a run that bypasses I *serves* B, II *serves* A and II *serves* B is excluded, because in such a run, from some point onwards, the action II *serves* B is perpetually enabled, in the sense that both tokens in its preplaces never move away.

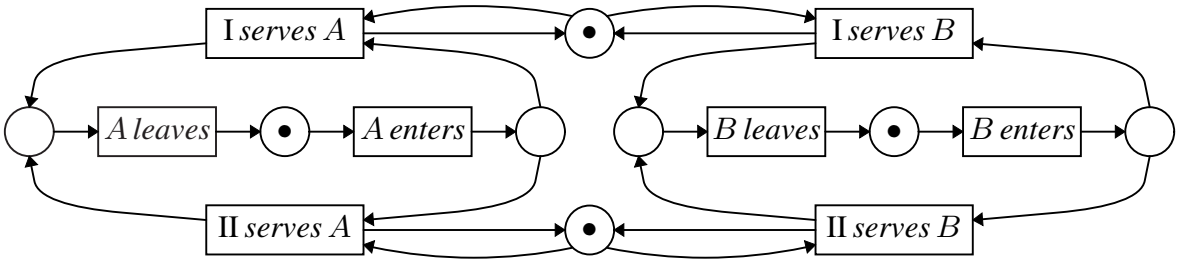


Fig. 4. Petri net semantics of the 2-customer 2-clerk store

Olderog [26,27] shows that the Petri net $[\![P]\!]$ associated to a closed CCSP expression P is safe, and that all its reachable markings are finite; the latter implies that it has bounded parallelism. The following result, from [26,27], shows that the standard interleaving semantics of CCSP is retrievable from the net semantics; it establishes a strong bisimulation relating any CCSP expression (seen as a state in a labelled transition system) with its interpretation as a marking in the Petri net of CCSP.

Theorem 1. There exists a relation $\mathscr{B}$ between closed CCSP expressions and markings in the unmarked Petri net of CCSP, such that

- $P \mathscr{B} dex(P)$ for each closed, well-typed CCSP expression with guarded recursion,
- if $P \mathscr{B} M$ and $P \xrightarrow{a} P'$ then there is a marking M' and transition t with $\ell(t) = a$, $M[t\rangle M'$ and $P \mathscr{B} M'$, and
- if $P \mathscr{B} M$ and $M[t\rangle M'$ then there is CCSP process P' with $P \xrightarrow{\ell(t)} P'$ and $P \mathscr{B} M'$.

To formalise the concurrency requirement for his net semantics Olderog defines for each n-ary CCSP operator op an n-ary operation $op_{\mathcal{N}}$ on safe Petri nets, inspired by proposals from [18,35,16], and requires that

$$\begin{array}{lrcl} (1) & [\![op(P_1,\ldots,P_n)]\!] & \approx & op_{\mathcal{N}}([\![P_1]\!],\ldots,[\![P_n]\!]) \\ (2) & [\![\langle X_A|\mathcal{S}\rangle]\!] & \approx & [\![\langle \mathcal{S}_{X_A}|\mathcal{S}\rangle]\!] \end{array}$$

for a suitable relation $\approx$. In fact, (2) turns out to hold taking for $\approx$ the identity relation. He establishes (1) taking for $\approx$ a relation he calls *strong bisimilarity*, whose definition will be recalled in Section 6. When a relation $\equiv$ includes $\approx$, and (1) holds for $\approx$, then it also holds for $\equiv$.

The operations $op_{\mathcal{N}}$ (i.e. $(0_A)_{\mathcal{N}}$ for $A \subseteq Act$, $a_{\mathcal{N}}$ for $a \in Act$, $R_{\mathcal{N}}$ for $R \subseteq Act \times Act$, $\|_{\mathcal{N}}$ and $+_{\mathcal{N}}$) are defined only up to isomorphism, but this is no problem as isomorphic nets are strongly bisimilar. The definition is recalled below—it generalises verbatim to non-safe nets, except that $+_{\mathcal{N}}$ is defined only for nets whose initial markings are nonempty plain sets.

Definition 3. [27] The net 0_A has type A and consists of a single place, initially marked: $(0_A)_{\mathcal{N}} := (\{0_A\}, \emptyset, \emptyset, \{0_A\}, A, \emptyset)$.

Given a net $N = (S, T, F, M, A, \ell)$ and $a \in Act$, take $s_0, t_a \notin S \cup T$. Then the net $a_{\mathcal{N}} N$ is obtained from N by the addition of the fresh place s_0 and the fresh transition t_a, labelled a, such that ${}^\bullet t_a = \{s_0\}$ and ${t_a}^\bullet = M$. The type of $a_{\mathcal{N}} N$ will be $A \cup \{a\}$ and the initial marking $\{s_0\}$.

Given a net $N = (S, T, F, M, A, \ell)$ and a renaming operator $R(\_)$, the net $R_{\mathcal{N}}(N)$ has type $R(A) := \{b \in Act \mid \exists a \in A, (a, b) \in R\}$, the same places and initial marking as N, and transitions t_b for each $t \in T$ and $b \in Act$ with $(\ell(t), b) \in R$. One has ${}^\bullet t_b := {}^\bullet t$, ${t_b}^\bullet := t^\bullet$, and the label of t_b will be b.

Given two nets $N_i = (S_i, T_i, F_i, M_i, A_i, \ell_i)$ $(i = 1, 2)$, their parallel composition $N_1 \|_{\mathcal{N}} N_2 = (S, T, F, M, A, \ell)$ is obtained from the disjoint union of N_1 and N_2 by the omission of all transitions t of $T_1 \cup T_2$ with $\ell(t) \in A_1 \cap A_2$, and the addition of fresh transitions (t_1, t_2) for all pairs $t_i \in T_i$ $(i = 1, 2)$ with $\ell_1(t_1) = \ell_2(t_2) \in A_1 \cap A_2$. Take ${}^\bullet(t_1, t_2) = {}^\bullet t_1 + {}^\bullet t_2$, $(t_1, t_2)^\bullet = {t_1}^\bullet + {t_2}^\bullet$, $\ell(t_1, t_2) = \ell(t_1)$, and $A := A_1 \cup A_2$.

Given nets $N_i = (S_i, T_i, F_i, M_i, A_i, \ell_i)$ with $M_i \neq \emptyset$ a plain set $(i = 1, 2)$, the net $N_1 +_{\mathcal{N}} N_2$ with type $A_1 \cup A_2$ is obtained from the disjoint union of N_1 and N_2 by the addition of the set of fresh places $M_1 \times M_2$—this set will be the initial marking of $N_1 +_{\mathcal{N}} N_2$—and the addition of fresh transitions t_i^K for any $t_i \in T_i$ and $\emptyset \neq K \leq {}^\bullet t_i \cap M_i$. $\ell(t_i^K) = \ell_i(t)$, ${}^\bullet t_1^K = {}^\bullet t_1 - K + (K \times M_2)$, ${}^\bullet t_2^K = {}^\bullet t_2 - K + (M_1 \times K)$ and $(t_i^K)^\bullet = {t_i}^\bullet$.

5 Structure preserving bisimulation equivalence

This section presents structure preserving bisimulation equivalence on nets.

Definition 4. Given two nets $N_i=(S_i,T_i,F_i,M_i,A_i,\ell_i)$, a *link* is a pair $(s_1, s_2)\in S_1\times S_2$ of places. A *linking* $l\in\mathbb{N}^{S_1\times S_1}$ is a multiset of links; it can be seen as a pair of markings with a bijection between them. Let $\pi_i(l)\in\mathbb{N}^{S_i}$ be these markings, given by $\pi_1(l)(s_1) = \sum_{s_2\in S_2} l(s_1, s_2)$ for all $s_1 \in S_1$ and $\pi_2(l)(s_2) = \sum_{s_1\in S_1} l(s_1, s_2)$ for all $s_2 \in S_2$. A *structure preserving bisimulation* (*sp-bisimulation*) is a set $\mathscr{B}$ of linkings, such that

- if $c \leq l \in \mathscr{B}$ and $\pi_1(c) = {}^\bullet t_1$ for $t_1 \in T_1$ then there are a transition $t_2 \in T_2$ with $\ell(t_2) = \ell(t_1)$ and $\pi_2(c) = {}^\bullet t_2$, and a linking $\bar{c}$ such that $\pi_1(\bar{c}) = t_1{}^\bullet$, $\pi_2(\bar{c}) = t_2{}^\bullet$ and $\bar{l} := l - c + \bar{c} \in \mathscr{B}$.
- if $c \leq l \in \mathscr{B}$ and $\pi_2(c) = {}^\bullet t_2$ then there are a t_1 and a $\bar{c}$ with the same properties.

N_1 and N_2 are *structure preserving bisimilar*, notation $N_1 \leftrightarrows_{sp} N_2$, if $A_1=A_2$ and there is a linking l in a structure preserving bisimulation with $M_1 = \pi_1(l)$ and $M_2 = \pi_2(l)$.

Note that if $\mathscr{B}$ is an sp-bisimulation, then so is its downward closure $\{k \mid \exists l\in\mathscr{B}.\, k \leq l\}$. Moreover, if $\mathscr{B}$ is an sp-bisimulation between two nets, then the set of those linkings $l \in \mathscr{B}$ for which $\pi_1(l)$ and $\pi_2(l)$ are reachable markings is also an sp-bisimulation.

If $\mathcal{B}$ is a set of a links, let $\overline{\mathcal{B}}$ be the set of *all* linkings that are multisets over $\mathcal{B}$.

Proposition 1. Structure preserving bisimilarity is an equivalence relation.

Proof. The relation $\overline{Id}$, with Id the identity relation on places, is an sp-bisimulation, showing that $N \leftrightarrows_{sp} N$ for any net N.

Given an sp-bisimulation $\mathscr{B}$, also $\{l^{-1} \mid l \in \mathscr{B}\}$ is an sp-bisimulation, showing symmetry of $\leftrightarrows_{sp}$.

Given linkings $h \in \mathbb{N}^{S_1\times S_3}$, $k \in \mathbb{N}^{S_1\times S_2}$ and $l \in \mathbb{N}^{S_2\times S_3}$, write $h \in k; l$ if there is a multiset $m \in \mathbb{N}^{S_1\times S_2\times S_3}$ of triples of places, with $k(s_1, s_2) = \sum_{s_3\in S} m(s_1, s_2, s_3)$, $l(s_2, s_3) = \sum_{s_1\in S} m(s_1, s_2, s_3)$ and $h(s_1, s_3) = \sum_{s_2\in S} m(s_1, s_2, s_3)$. Now, for sp-bisimulations $\mathscr{B}$ and $\mathscr{B}'$, also $\mathscr{B}; \mathscr{B}' := \{h\in k; l \mid k\in\mathscr{B}\wedge l\in\mathscr{B}'\}$ is an sp-bisimulation, showing transitivity of $\leftrightarrows_{sp}$. □

6 Strong bisimilarity

As discussed in the introduction and at the end of Section 4, Olderog defined a relation of *strong bisimilarity* on safe Petri nets.

Definition 5. For $\mathcal{B} \subseteq S_1 \times S_2$ a binary relation between the places of two safe nets $N_i = (S_i, T_i, F_i, M_i, A_i, \ell_i)$, write $\widehat{\mathcal{B}}$ for the set of all linkings $l \subseteq \mathcal{B}$ such that $\pi_i(l)$ is a reachable marking of N_i for $i = 1, 2$ and $\mathcal{B} \cap \big(\pi_1(l) \times \pi_2(l)\big) = l$. Now a *strong bisimulation* as defined in [27] can be seen as a structure preserving bisimulation of the form $\widehat{\mathcal{B}}$. The nets N_1 and N_2 are *strongly bisimilar* if $A_1 = A_2$ and there is a linking l in a strong bisimulation with $M_1 = \pi_1(l)$ and $M_2 = \pi_2(l)$.

This reformulation of the definition from [27] makes immediately clear that strong bisimilarity of two safe Petri nets implies their structure preserving bisimilarity. Consequently, the concurrency requirement for the net semantics from Olderog, as formalised by Requirements (1) and (2) in Section 4, holds for structure preserving bisimilarity.

7 Compositionality

In this section I show that structure preserving bisimilarity is a congruence for the operators of CCSP, or, in other words, that these operators are compositional up to $\leftrightarrows_{sp}$.

Theorem 2. If $N_1 \leftrightarrows_{sp} N_2$, $a \in Act$ and $R \subseteq Act \times Act$, then $a_{\mathcal{N}} N_1 \leftrightarrows_{sp} a_{\mathcal{N}} N_2$ and $R_{\mathcal{N}}(N_2) \leftrightarrows_{sp} R_{\mathcal{N}}(N_2)$. If $N_1^l \leftrightarrows_{sp} N_2^l$ and $N_1^r \leftrightarrows_{sp} N_2^r$ then $N_1^l \| _{\mathcal{N}} N_1^r \leftrightarrows_{sp} N_2^l \| _{\mathcal{N}} N_2^r$ and, if the initial markings of N_i^l and N_i^r are nonempty sets, $N_1^l +_{\mathcal{N}} N_1^r \leftrightarrows_{sp} N_2^l +_{\mathcal{N}} N_2^r$.

Proof. Let $N_i = (S_i, T_i, F_i, M_i, A_i, \ell_i)$ for $i = 1, 2$, and let s_i and u_i be the fresh place and transition introduced in the definition of $a_{\mathcal{N}} N_i$. From $N_1 \leftrightarrows_{sp} N_2$ it follows that $A_1 = A_2$ and hence $A_1 \cup \{a\} = A_2 \cup \{a\}$.

Let $\mathscr{B}$ be an sp-bisimulation containing a linking k with $M_i = \pi_i(k)$ for $i = 1, 2$. Let $\mathscr{B}_a := \mathscr{B} \cup \{h\}$, with $h = \{(s_1, s_2)\}$. Then h links the initial markings of $a_{\mathcal{N}} N_1$ and $a_{\mathcal{N}} N_2$. Hence it suffices to show that $\mathscr{B}_a$ is an sp-bisimulation. So suppose $c \leq h$ and $\pi_1(c) = {}^\bullet t_1$ for some $t_1 \in T_1$. Then $c = h$ and $t_1 = u_1$. Take $t_2 := u_2$ and $\bar{h} := \bar{c} := k$.

To show that $R_{\mathcal{N}}(N_2) \leftrightarrows_{sp} R_{\mathcal{N}}(N_2)$ it suffices to show that $\mathscr{B}$ also is an sp-bisimulation between $R_{\mathcal{N}}(N_2)$ and $R_{\mathcal{N}}(N_2)$, which is straightforward.

Now let $N_i^l = (S_i^l, T_i^l, F_i^l, M_i^l, A_i^l, \ell_i^l)$ and $N_i^r = (S_i^r, T_i^r, F_i^r, M_i^r, A_i^r, \ell_i^r)$ for $i = 1, 2$. Let $A := A_1^l \cap A_1^r = A_2^l \cap A_2^r$. Create the disjoint union of N_i^l and N_i^r in the definition of $N_i^l \| _{\mathcal{N}} N_i^r$ by renaming all places s and transitions t of N_i^l into $s\|_A$ and $t\|_A$, and all places s and transitions t of N_i^r into ${}_A\|s$ and ${}_A\|t$. Let $\mathscr{B}^l$ and $\mathscr{B}^r$ be sp-bisimulations containing linkings k^l and k^r, respectively, with $M_i^l = \pi_i(k^l)$ and $M_i^r = \pi_i(k^r)$, for $i = 1, 2$. Take $\mathscr{B} := \{(h^l\|_A) + ({}_A\|h^r) \mid h^l \in \mathscr{B}^l \wedge h^r \in \mathscr{B}^r\}$, where $h^l\|_A := \{(s_1\|_A, s_2\|_A) \mid (s_1, s_2) \in h^l\}$, and ${}_A\|h^r$ is defined likewise. Then $\pi_i((k^l\|_A) + ({}_A\|k^r)) = \pi_i(k^l)\|_A + {}_A\|\pi_i(k^r) = M_i^l\|_A + {}_A\|M_i^r$ is the initial marking of $N_i^l \| _{\mathcal{N}} N_i^r$ for $i = 1, 2$, so it suffices to show that $\mathscr{B}$ is an sp-bisimulation.

So suppose $c \leq (h^l\|_A) + ({}_A\|h^r) \in \mathscr{B}$ with $h^l \in \mathscr{B}^l \wedge h^r \in \mathscr{B}^r$ and $\pi_1(c) = {}^\bullet t_1$ for t_1 a transition of $N_1^l \| _{\mathcal{N}} N_1^r$. Then c has the form $(c^l\|_A) + ({}_A\|c^r)$ for $c^l \leq h^l \in \mathscr{B}^l$ and $c^r \leq h^r \in \mathscr{B}^r$, and t_1 has the form (i) $t_1^l\|_A$ for $t_1^l \in T_1^l$ with $\ell_1^l(t_1^l) \notin A$, or (ii) $(t_1^l\|_A, {}_A\|t_1^r)$ for $t_1^l \in T_1^l$ and $t_1^r \in T_1^r$ with $\ell_1^l(t_1^l) = \ell_1^r(t_1^r) \in A$, or (iii) ${}_A\|t_1^r$ for $t_1^r \in T_1^r$ with $\ell_1^r(t_1^r) \notin A$. In case (i) one has $c^r = \emptyset$ and $\pi_1(c^l) = {}^\bullet t_1^l$, whereas in case (ii) $\pi_1(c^l) = {}^\bullet t_1^l$ and $\pi_1(c^r) = {}^\bullet t_1^r$. I only elaborate case (ii); the other two proceed likewise. Since $\mathscr{B}^l$ is an sp-bisimulation, there are a transition t_2^l with $\ell_2^l(t_2^l) = \ell_1^l(t_1^l)$ and $\pi_2(c^l) = {}^\bullet t_2^l$, and a linking $\bar{c}^l$ such that $\pi_1(\bar{c}^l) = t_1^{l\bullet}$, $\pi_2(\bar{c}^l) = t_2^{l\bullet}$ and $\bar{h}^l := h^l - c^l + \bar{c}^l \in \mathscr{B}^l$. Likewise, since $\mathscr{B}^r$ is an sp-bisimulation, there are a transition t_2^r with $\ell_2^r(t_2^r) = \ell_1^r(t_1^r)$ and $\pi_2(c^r) = {}^\bullet t_2^r$, and a linking $\bar{c}^r$ such that $\pi_1(\bar{c}^r) = t_1^{r\bullet}$, $\pi_2(\bar{c}^r) = t_2^{r\bullet}$ and $\bar{h}^r := h^r - c^r + \bar{c}^r \in \mathscr{B}^r$. Take $t_2 := (t_2^l\|_A, {}_A\|t_2^r)$. This transition has the same label as t_2^l, t_2^r, t_1^l, t_1^r and $(t_1^l\|_A, {}_A\|t_1^r) = t_1$. Moreover, $\pi_2(c) = \pi_2(c^l)\|_A + {}_A\|\pi_2(c^r) = {}^\bullet t_2^l\|_A + {}_A\|{}^\bullet t_2^r = {}^\bullet t_2$. Take $\bar{c} := (\bar{c}^l\|_A) + ({}_A\|\bar{c}^r)$. Then $\pi_1(\bar{c}) = t_1{}^\bullet$, $\pi_2(\bar{c}) = t_2{}^\bullet$ and $\bar{h} := (h^l\|_A) + ({}_A\|h^r) - c + \bar{c} = (\bar{h}^l\|_A) + ({}_A\|\bar{h}^r) \in \mathscr{B}$.

Let $N_i^l = (S_i^l, T_i^l, F_i^l, M_i^l, A_i^l, \ell_i^l)$ and $N_i^r = (S_i^r, T_i^r, F_i^r, M_i^r, A_i^r, \ell_i^r)$ for $i = 1, 2$, with M_i^l and M_i^r nonempty plain sets, but this time I assume the nets to already be disjoint, and such that all the places and transitions added in the construction of $N_i^l +_{\mathcal{N}} N_i^r$ are fresh. Let $\mathscr{B}^l$ and $\mathscr{B}^r$ be as above. Without loss of generality I may assume that the linkings h in $\mathscr{B}^l$ and $\mathscr{B}^r$ have the property that $\pi_i(h)$ is a reachable marking for

$i = 1, 2$, so that the restriction of $\pi_i(h)$ to M_i^l or M_i^r is a plain set. Define

$$\begin{aligned}\mathscr{B}^+ := \{&h_\bullet^l + (h_+^l \otimes k^r) \mid h_\bullet^l + h_+^l \in \mathscr{B}^l \wedge h_+^l \lneqq k^l\}\\ \{&h_\bullet^r + (k^l \otimes h_+^r) \mid h_\bullet^r + h_+^r \in \mathscr{B}^r \wedge h_+^r \lneqq k^r\} \cup \{k^l \otimes k^r\}\end{aligned}$$

where $h^l \otimes h^r := \{((s_1^l, s_1^r), (s_2^l, s_2^r)) \mid (s_1^l, s_2^l) \in h^l \wedge (s_1^r, s_2^r) \in h^r\}$. Now $\pi_i(k^l \otimes k^r) = \pi_i(k^l) \times \pi_i(k^r) = M_i^l \times M_i^r$ is the initial marking of $N_i^l +_{\mathcal{N}} N_i^r$, so again it suffices to show that $\mathscr{B}^+$ is an sp-bisimulation.

So suppose $c \leq h_\bullet^l + (h_+^l \otimes k^r) \in \mathscr{B}^+$ with $h_\bullet^l + h_+^l \in \mathscr{B}^l$, $h_+^l \lneqq k^l$ and $\pi_1(c) = {}^\bullet t_1$ for t_1 a transition of $N_1^l +_{\mathcal{N}} N_1^r$.

First consider the case that $c \leq h_\bullet^l$. Then $c \leq h_\bullet^l \leq h_\bullet^l + h_+^l \in \mathscr{B}^l$. Since $\mathscr{B}^l$ is an sp-bisimulation, there are a transition $t_2 \in T_2^l$ with $\ell_2^l(t_2) = \ell_1^l(t_1)$ and $\pi_2(c) = {}^\bullet t_2$, and a linking $\bar{c}$ such that $\pi_1(\bar{c}) = t_1{}^\bullet$, $\pi_2(\bar{c}) = t_2{}^\bullet$ and $h_\bullet^l + h_+^r - c + \bar{c} \in \mathscr{B}^l$. Now $h_\bullet^l + (h_+^l \otimes k^r) - c + \bar{c} = (h_\bullet^l - c + \bar{c}) + (h_+^r \otimes k_2) \in \mathscr{B}^+$ because $(h_\bullet^l - c + \bar{c}) + h_+^r \in \mathscr{B}^l$.

In the remaining case $\pi_1(c)$ contains a place $(s_1^l, s_1^r) \in M_1^l \times M_1^r$, so t_1 must have either the form t_{1l}^K with $\emptyset \neq K \leq {}^\bullet t_1^l \cap M_1^l$ for some $t_1^l \in T_1^l$, or t_{1r}^K with $\emptyset \neq K \leq {}^\bullet t_1^r \cap M_1^r$ for some $t_1^r \in T_1^r$. First assume, towards a contradiction, that $t_1 = t_{1r}^K$. Then $M_1^l \times K \leq {}^\bullet t_{1r}^K = \pi_1(c) \leq \pi_1(h_\bullet^l) + \pi_1(h_+^l \otimes k^r)$. Since the places in $M_1^l \times K \subseteq M_1^l \times M_1^r$ are fresh, it follows that $M_1^l \times K \leq \pi_1(h_+^l \otimes k^r) \leq \pi_1(h_+^l) \times \pi_1(k^r) \leq \pi_1(h_+^l) \times M_1^r$, implying that $M_1^l \leq \pi_1(h_+^l)$ and $K \leq M_1^r$—here I use that $M_1^l \neq \emptyset \neq K$ and $\pi_1(h_+^l)$ and M_1^r are plain sets. However, the condition $h_+^l \lneqq k^l$ implies that $\pi_1(h_+^l) \lneqq \pi_1(k^l) = M_1^l$, yielding a contradiction. Hence t_1 is of the form t_{1l}^K.

Since $\pi_1(c) = {}^\bullet t_{1l}^K = {}^\bullet t_1^l - K + (K \times M_1^r)$, the linking c must have the form $c_\bullet + c'$ with $\pi_1(c_\bullet) = {}^\bullet t_1^l - K$ and $\pi_1(c') = K \times M_1^r$. As no place in ${}^\bullet t_1^l - K$ can be in $M_1^l \times M_1^r \supseteq \pi_1(h_+^l \otimes k^r)$, it follows that $c_\bullet \leq h_\bullet^l$. Likewise, as none of the places in $K \times M_1^r$ can be in $\pi_1(h_\bullet^l)$, it follows that $c' \leq h_+^l \otimes k^r$. Thus $K \times M_1^r = \pi_1(c') \leq \pi_1(h_+^l \otimes k^r) \leq \pi_1(h_+^l) \times \pi_1(k^r) \leq \pi_1(h_+^l) \times M_1^r$, implying $K \leq \pi_1(h_+^l)$—again using that $\pi_1(h_+^l)$ and $M_1^r \neq \emptyset$ are plain sets. The linking $h_+^l \otimes k^r$ has the property that its projection $\pi_1(h_+^l \otimes k^r)$ is a plain set. Since a subset c'' of a such linking is completely determined by its first projection $\pi_1(c'')$, it follows that $c' = c_+ \otimes k^r$ for the unique linking $c_+ \leq h_+^l$ with $\pi_1(c_+) = K$.

Now $c_\bullet + c_+ \leq h_\bullet^l + h_+^l \in \mathscr{B}^l$ and $\pi_1(c_\bullet + c_+) = ({}^\bullet t_1^l - K) + K = {}^\bullet t_1^l$. Since $\mathscr{B}^l$ is an sp-bisimulation, there are a transition $t_2^l \in T_2^l$ with $\ell_2^l(t_2^l) = \ell_1^l(t_1^l)$ and $\pi_2(c_\bullet + c_+) = {}^\bullet t_2^l$, and a linking $\bar{c}$ such that $\pi_1(\bar{c}) = t_1^l{}^\bullet$, $\pi_2(\bar{c}) = t_2^l{}^\bullet$ and $h_\bullet^l + h_+^l - (c_\bullet + c_+) + \bar{c} \in \mathscr{B}^l$. Let $L := \pi_2(c_+)$. Then $L \neq \emptyset$ since $K \neq \emptyset$, $L = \pi_2(c_+) \leq \pi_2(h_+^l) \leq \pi_2(k^l) = M_2^l$ and $L = \pi_2(c_+) \leq \pi_2(c_\bullet + c_+) = {}^\bullet t_2^l$. By Definition 3 $N_2^l +_{\mathcal{N}} N_2^r$ has a transition t_{2l}^L with $\ell(t_{2l}^L) = \ell_2^l(t_2^l) = \ell_1^l(t_1^l) = \ell(t_{1l}^L)$, ${}^\bullet t_{2l}^L = {}^\bullet t_2^l - L + (L \times M_2^l) = \pi_2(c_\bullet + c_+) - \pi_2(c_+) + (\pi_2(c_+) \times \pi_1(k^r)) = \pi_2(c_\bullet + (c_+ \otimes k^r)) = \pi_2(c)$ and $t_{2l}^L{}^\bullet = t_2^l{}^\bullet = \pi_2(\bar{c})$. Moreover, $\pi_1(\bar{c}) = t_1^l{}^\bullet = t_1^K{}^\bullet$. Finally, $h_\bullet^l + (h_+^l \otimes k^r) - c + \bar{c} = (h_\bullet^l - c_\bullet + \bar{c}) + ((h_+^l - c_+) \otimes k^r) \in \mathscr{B}^+$ since $(h_\bullet^l - c_\bullet + c') + (h_+^l - c_+) \in \mathscr{B}^l$ and $h_+^l - c_+ \leq h_+^l \lneqq k^l$.

The case supposing $c \leq h_\bullet^r + (k^r \otimes h_+^r) \in \mathscr{B}^+$ follows by symmetry, whereas the case $c \leq k^l \otimes k^r$ proceeds by simplification of the other two cases. □

8 Processes of nets and causal equivalence

A *process* of a net N [29,9,19] is essentially a conflict-free, acyclic net together with a mapping function to N. It can be obtained by unwinding N, choosing one of the alternatives in case of conflict. It models a run, or concurrent computation, of N. The acyclic nature of the process gives rise to a notion of causality for transition firings in the original net via the mapping function. A conflict present in the original net is represented by the existence of multiple processes, each representing one possible way to decide the conflict. This notion of process differs from the one used in process algebra; there a "process" refers to the entire behaviour of a system, including all its choices.

Definition 6. A *causal net*[2] is a net $\mathcal{N} = (\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0, \mathcal{A}, \ell_{\mathcal{N}})$ satisfying

- $\forall s \in \mathcal{S}. |{}^\bullet s| \leq 1 \geq |s^\bullet| \wedge \mathcal{M}_0(s) = \begin{cases} 1 & \text{if } {}^\bullet s = \emptyset \\ 0 & \text{otherwise,} \end{cases}$
- $\mathcal{F}$ is acyclic, i.e., $\forall x \in \mathcal{S} \cup \mathcal{T}. (x,x) \notin \mathcal{F}^+$, where $\mathcal{F}^+$ is the transitive closure of $\{(x,y) \mid \mathcal{F}(x,y) > 0\}$,
- and $\{t \in \mathcal{T} \mid (t,u) \in \mathcal{F}^+\}$ is finite for all $u \in \mathcal{T}$.

A *folding* from a net $\mathcal{N} = (\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0, \mathcal{A}, \ell_{\mathcal{N}})$ into a net $N = (S, T, F, M_0, A, \ell)$ is a function $\rho : \mathcal{S} \cup \mathcal{T} \to S \cup T$ with $\rho(\mathcal{S}) \subseteq S$ and $\rho(\mathcal{T}) \subseteq T$, satisfying

- $\mathcal{A} = A$ and $\ell_{\mathcal{N}}(t) = \ell(\rho(t))$ for all $t \in \mathcal{T}$,
- $\rho(\mathcal{M}_0) = M_0$, i.e. $M_0(s) = |\rho^{-1}(s) \cap \mathcal{M}_0|$ for all $s \in S$, and
- $\forall t \in \mathcal{T}, s \in S. F(s, \rho(t)) = |\rho^{-1}(s) \cap {}^\bullet t| \wedge F(\rho(t), s) = |\rho^{-1}(s) \cap t^\bullet|$. [3]

A pair $\mathcal{P} = (\mathcal{N}, \rho)$ of a causal net $\mathcal{N}$ and a folding of $\mathcal{N}$ into a net N is a *process* of N. $\mathcal{P}$ is called *finite* if $\mathcal{T}$ is finite.

Note that if N has bounded parallelism, than so do all of its processes.

Definition 7. [27] A net $\mathcal{N}$ is called a causal net *of* a net N if it is the first component of a process $(\mathcal{N}, \rho)$ of N. Two nets N_1 and N_2 are *causal equivalent*, notation $\equiv_{caus}$, if they have the same causal nets.

Olderog shows that his relation of strong bisimilarity is included in $\equiv_{caus}$ [27], and thereby establishes the concurrency requirement (1) from Section 4 for $\equiv_{caus}$.

For $\mathcal{N} = (\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0, \mathcal{A}, \ell_{\mathcal{N}})$ a causal net, let $\mathcal{N}^\circ := \{s \in \mathcal{S} \mid s^\bullet = \emptyset\}$. The following result supports the claim that finite processes model finite runs.

Proposition 2. [19, Theorems 3.5 and 3.6] M is a reachable marking of a net N iff N has a finite process $(\mathcal{N}, \rho)$ with $\rho(\mathcal{N}^\circ) = M$. Here $\rho(\mathcal{N}^\circ)(s) = |\rho^{-1}(s) \cap \mathcal{N}^\circ|$.

[2] A causal net [29,34] is traditionally called an *occurrence net* [9,19,33]. Here, following [27], I will not use the terminology "occurrence net" in order to avoid confusion with the occurrence nets of [25,36]; the latter extend causal nets with forward branching places, thereby capturing all runs of the represented system, together with the branching structure between them.

[3] For $H \subseteq \mathcal{S}$, the multiset $\rho(H) \in \mathbb{N}^S$ is defined by $\rho(H)(s) = |\rho^{-1}(s) \cap H|$. Using this, these conditions can be reformulated as $\rho({}^\bullet t) = {}^\bullet \rho(t)$ and $\rho(t^\bullet) = \rho(t)^\bullet$.

A process is not required to represent a completed run of the original net. It might just as well stop early. In those cases, some set of transitions can be added to the process such that another (larger) process is obtained. This corresponds to the system taking some more steps and gives rise to a natural order between processes.

Definition 8. Let $\mathcal{P} = ((\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0, \mathcal{A}, \ell), \rho)$ and $\mathcal{P}' = ((\mathcal{S}', \mathcal{T}', \mathcal{F}', \mathcal{M}'_0, \mathcal{A}', \ell'), \rho')$ be two processes of the same net. $\mathcal{P}'$ is a *prefix* of $\mathcal{P}$, notation $\mathcal{P}' \leq \mathcal{P}$, and $\mathcal{P}$ an *extension* of $\mathcal{P}'$, iff $\mathcal{S}' \subseteq \mathcal{S}$, $\mathcal{T}' \subseteq \mathcal{T}$, $\mathcal{M}'_0 = \mathcal{M}_0$, $\mathcal{F}' = \mathcal{F} \restriction (\mathcal{S}' \times \mathcal{T}' \cup \mathcal{T}' \times \mathcal{S}')$ and $\rho' = \rho \restriction (\mathcal{S}' \cup \mathcal{T}')$. (This implies that $\mathcal{A}' = \mathcal{A}$ and $\ell' = \ell \restriction \mathcal{T}$.)

The requirements above imply that if $\mathcal{P}' \leq \mathcal{P}$, $(x, y) \in \mathcal{F}^+$ and $y \in \mathcal{S}' \cup \mathcal{T}'$ then $x \in \mathcal{S}' \cup \mathcal{T}'$. Conversely, any subset $\mathcal{T}' \subseteq \mathcal{T}$ satisfying $(t, u) \in \mathcal{F}^+ \wedge u \in \mathcal{T}' \Rightarrow t \in \mathcal{T}'$ uniquely determines a prefix of $\mathcal{P}$. A process $(\mathcal{N}, \rho)$ of a net N is *initial* if $\mathcal{N}$ contains no transitions; then $\rho(\mathcal{N}^\circ)$ is the initial marking of N. Any process has an initial prefix.

Proposition 3. [19, Theorem 3.17] If $\mathcal{P}_i = ((\mathcal{S}_i, \mathcal{T}_i, \mathcal{F}_i, \mathcal{M}_{0i}, \mathcal{A}_i, \ell_i), \rho_i)$ $(i \in \mathbb{N})$ is a chain of processes of a net N, satisfying $\mathcal{P}_i \leq \mathcal{P}_j$ for $i \leq j$, then there exists a unique process $\mathcal{P} = ((\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{M}_0, \mathcal{A}, \ell), \rho)$ of N with $\mathcal{S} = \bigcup_{i \in \mathbb{N}} \mathcal{S}_i$ and $\mathcal{T} = \bigcup_{i \in \mathbb{N}} \mathcal{T}_i$—the *limit* of this chain—such that $\mathcal{P}_i \leq \mathcal{P}$ for all $i \in \mathbb{N}$. □

In [29,9,19] processes were defined without the third requirement of Definition 6. Goltz and Reisig [19] observed that certain processes did not correspond with runs of systems, and proposed to restrict the notion of a process to those that can be obtained as the limit of a chain of finite processes [19, end of Section 3]. By [19, Theorems 3.18 and 2.14], for processes of finite nets this limitation is equivalent with imposing the third bullet point of Definition 6. My restriction to nets with bounded parallelism serves to recreate this result for processes of infinite nets.

Proposition 4. Any process of a net can be obtained as the limit of a chain of finite approximations.

Proof. Define the *depth* of a transition u in a causal net as one more than the maximum of the depth of all transitions t with tF^+u. Since the set of such transitions t is finite, the depth of a transition u is a finite integer. Now, given a process $\mathcal{P}$, the approximation $\mathcal{P}_i$ is obtained by restricting to those transitions in $\mathcal{P}$ of depth $\leq i$, together with all their pre- and postplaces, and keeping the initial marking. Clearly, these approximations form a chain, with limit $\mathcal{P}$. By induction on i one shows that $\mathcal{P}_i$ is finite. For $\mathcal{P}_0$ this is trivial, as it has no transitions. Now assume $\mathcal{P}_i$ is finite but $\mathcal{P}_{i+1}$ is not. Executing, in $\mathcal{P}_{i+1}$, all transitions of $\mathcal{P}_i$ one by one leads to a marking of $\mathcal{P}_{i+1}$ in which all remaining transitions of $\mathcal{P}_{i+1}$ are enabled. As these transitions cannot have common preplaces, this violates the assumption that $\mathcal{P}_{i+1}$ has bounded parallelism. □

9 A process-based characterisation of sp-bisimilarity

This section presents an alternative characterisation of sp-bisimilarity that will be instrumental in obtaining Theorems 4 and 5, saying that $\leftrightarrows_{sp}$ is a finer semantic equivalence than $\equiv_{caus}$ and $\approx_h$. This characterisation could have been presented as the original definition; however, the latter is instrumental in showing that $\leftrightarrows_{sp}$ is coarser than $\approx_{pb}$ and $\equiv_{occ}$, and implied by Olderog's strong bisimilarity.

Definition 9. A *process-based sp-bisimulation* between two nets N_1 and N_2 is a set $\mathscr{R}$ of triples $(\rho_1, \mathcal{N}, \rho_2)$ with $(\mathcal{N}, \rho_i)$ a finite process of N_i, for $i = 1, 2$, such that

- $\mathscr{R}$ contains a triple $(\rho_1, \mathcal{N}, \rho_2)$ with $\mathcal{N}$ a causal net containing no transitions,
- if $(\rho_1, \mathcal{N}, \rho_2) \in \mathscr{R}$ and $(\mathcal{N}', \rho_i')$ with $i \in \{1, 2\}$ is a fin. proc. of N_i extending $(\mathcal{N}, \rho_i)$ then N_j with $j := 3-i$ has a process $(\mathcal{N}', \rho_j') \geq (\mathcal{N}, \rho_j)$ such that $(\rho_1', \mathcal{N}', \rho_2') \in \mathscr{R}$.

Theorem 3. Two nets are sp-bisimilar iff there exists a process-based sp-bisimulation between them.

Proof. Let $\mathscr{R}$ be a process-based sp-bisimulation between nets N_1 and N_2. Define $\mathscr{B} := \{\{(\rho_1(\mathfrak{s}), \rho_2(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{N}^\circ\} \mid (\rho_1, \mathcal{N}, \rho_2) \in \mathscr{R}\}$. Then $\mathscr{B}$ is an sp-bisimulation:

- Let $c \leq l \in \mathscr{B}$ and $\pi_1(c) = {}^\bullet t_1$ for $t_1 \in T_1$. Then $l = \{(\rho_1(\mathfrak{s}), \rho_2(\mathfrak{s}) \mid \mathfrak{s} \in \mathcal{N}^\circ\}$ for some $(\rho_1, \mathcal{N}, \rho_2) \in \mathscr{R}$. Extend $\mathcal{N}$ to $\mathcal{N}'$ by adding a fresh transition $\mathfrak{t}$ and fresh places s_i for $s \in S_1$ and $i \in \mathbb{N}$ with $F_1(t_1, s) > i$; let ${}^\bullet\mathfrak{t} = \{\mathfrak{s} \in \mathcal{N}^\circ \mid \rho_1(\mathfrak{s}) \in {}^\bullet t_1\}$ and $\mathfrak{t}^\bullet = \{s_i \mid s \in S_1 \wedge i \in \mathbb{N} \wedge F_1(t_1, s) > i\}$. Furthermore, extend ρ_1 to ρ_1' by $\rho_1'(\mathfrak{t}) := t_1$ and $\rho_1'(s_i) := s$. Then ${}^\bullet\rho_1'(\mathfrak{t}) = {}^\bullet t_1 = \rho_1'({}^\bullet\mathfrak{t})$ and $\rho_1'(\mathfrak{t})^\bullet = t_1{}^\bullet = \rho_1'(\mathfrak{t}^\bullet)$, so $(\mathcal{N}', \rho_1')$ is a process of N_1, extending $(\mathcal{N}, \rho_1)$. Since $\mathscr{R}$ is a process-based sp-bisimulation, N_2 has a process $(\mathcal{N}', \rho_2') \geq (\mathcal{N}, \rho_2)$ such that $(\rho_1', \mathcal{N}', \rho_2') \in \mathscr{R}$. Take $t_2 := \rho_2'(\mathfrak{t})$. Then $\ell_2(t_2) = \ell_{\mathcal{N}}(\mathfrak{t}) = \ell_1(t_1)$ and $c = \{(\rho_1(\mathfrak{s}), \rho_2(\mathfrak{s}) \mid \mathfrak{s} \in {}^\bullet\mathfrak{t}\}$, so $\pi_2(c) = \{\rho_2(\mathfrak{s}) \mid \mathfrak{s} \in {}^\bullet\mathfrak{t}\} = \rho_2({}^\bullet\mathfrak{t}) = \rho_2'({}^\bullet\mathfrak{t}) = {}^\bullet\rho_2'(\mathfrak{t}) = {}^\bullet t_2$. Take $c' := \{(\rho_1'(\mathfrak{s}), \rho_2'(\mathfrak{s})) \mid \mathfrak{s} \in \mathfrak{t}^\bullet\}$. Then $\pi_1(c') = t_1{}^\bullet$, $\pi_2(c') = t_2{}^\bullet$ and $l' := l - c + c' = \{(\rho_1'(\mathfrak{s}), \rho_2'(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{N}^\circ - {}^\bullet\mathfrak{t} + \mathfrak{t}^\bullet\} = \{(\rho_1'(\mathfrak{s}), \rho_2'(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{N}'^\circ\} \in \mathscr{B}$.
- The other clause follows by symmetry.

Since $\mathscr{R}$ contains a triple $(\rho_1, \mathcal{N}, \rho_2)$ with $\mathcal{N}$ a causal net containing no transitions, $\mathscr{B}$ contains a linking $l := \{(\rho_1(\mathfrak{s}), \rho_2(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{N}^\circ$ such that $\pi_i(l) = \rho_i(\mathcal{N}^\circ) = M_i$ for $i = 1, 2$, where M_i is the initial marking of N_i. Since $(\mathcal{N}, \rho_i)$ is a process of N_i, N_i must have the the same type as $\mathcal{N}$, for $i = 1, 2$. It follows that $N_1 \leftrightarrows_{sp} N_2$.

Now let $\mathscr{B}$ be an sp-bisimulation between nets N_1 and N_2. Let $\mathscr{R} := \{(\rho_1, \mathcal{N}, \rho_2) \mid (\mathcal{N}, \rho_i)$ is a finite process of N_i $(i = 1, 2)$ and $\{(\rho_1(\mathfrak{s}), \rho_2(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{N}^\circ\} \in \mathscr{B}\}$. Then $\mathscr{R}$ is a process-based sp-bisimulation.

- $\mathscr{B}$ must contain a linking l with $\pi_i(l) = M_i$ for $i = 1, 2$, where M_i is the initial marking of N_i; let $l = \{(s_1^k, s_2^k) \mid k \in K\}$. Let $\mathcal{N}$ be a causal net with places $\mathfrak{s}^k$ for $k \in K$ and no transitions, and define ρ_i for $i = 1, 2$ by $\rho_i(\mathfrak{s}^k) = s_i^k$ for $k \in K$. Then $(\mathcal{N}, \rho_i)$ is an initial process of N_i $(i = 1, 2)$ and $(\rho_1, \mathcal{N}, \rho_2) \in \mathscr{R}$.
- Suppose $(\rho_1, \mathcal{N}, \rho_2) \in \mathscr{R}$ and $(\mathcal{N}', \rho_1')$ is a finite process of N_1 extending $(\mathcal{N}, \rho_1)$. (The case of a finite process of N_2 extending $(\mathcal{N}, \rho_1)$ will follow by symmetry.) Then $l := \{(\rho_1(\mathfrak{s}), \rho_2(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{N}^\circ\} \in \mathscr{B}$. Without loss of generality, I may assume that $\mathcal{N}'$ extends $\mathcal{N}$ by just one transition, $\mathfrak{t}$. The definition of a causal net ensures that ${}^\bullet\mathfrak{t} \subseteq \mathcal{N}^\circ$, and the definition of a process gives $\rho_1'({}^\bullet\mathfrak{t}) = {}^\bullet t_1$, where $t_1 := \rho_1'(\mathfrak{t})$. Let $c := \{(\rho_1(\mathfrak{s}), \rho_2(\mathfrak{s})) \mid \mathfrak{s} \in {}^\bullet\mathfrak{t}\}$. Then $c \leq l$ and $\pi_1(c) = \rho_1({}^\bullet\mathfrak{t}) = \rho_1'({}^\bullet\mathfrak{t}) = {}^\bullet t_1$. Since $\mathscr{B}$ is an sp-bisimulation, there are a transition t_2 with $\ell(t_2) = \ell(t_1)$ and $\pi_2(c) = {}^\bullet t_2$, and a linking c' such that $\pi_1(c') = t_1{}^\bullet$, $\pi_2(c') = t_2{}^\bullet$ and $l' := l - c + c' \in \mathscr{B}$. The definition of a process gives $\rho_1'(\mathfrak{t}^\bullet) = t_1{}^\bullet$. This makes it possible to extend ρ_2 to ρ_2' so that $\rho_2'(\mathfrak{t}) := t_2$, $\rho_2'(\mathfrak{t}^\bullet) = t_2{}^\bullet$ and $c' = \{(\rho_1'(\mathfrak{s}), \rho_2'(\mathfrak{s})) \mid \mathfrak{s} \in \mathfrak{t}^\bullet\}$. Moreover, $\rho_2'({}^\bullet\mathfrak{t}) = \rho_2({}^\bullet\mathfrak{t}) = \pi_2(c) = {}^\bullet t_2$. Thus $(\mathcal{N}', \rho_2')$ is a finite process of N_2 extending $(\mathcal{N}, \rho_2)$. Furthermore, $\{(\rho_1'(\mathfrak{s}), \rho_2'(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{N}'^\circ\} = \{(\rho_1'(\mathfrak{s}), \rho_2'(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{N}^\circ - {}^\bullet\mathfrak{t} + \mathfrak{t}^\bullet\} = l - c + c' \in \mathscr{B}$. Hence $(\rho_1', \mathcal{N}', \rho_2') \in \mathscr{R}$. □

10 Relating sp-bisimilarity to other semantic equivalences

In this section I place sp-bisimilarity in the spectrum of existing semantic equivalences for nets, as indicated in Figure 1.

10.1 Place bisimilarity

The notion of a place bisimulation, defined in [1], can be reformulated as follows.

Definition 10. A *place bisimulation* is a structure preserving bisimulation of the form $\overline{\mathcal{B}}$ (where $\overline{\mathcal{B}}$ is defined in Section 5). Two nets $N_i = (S_i, T_i, F_i, M_i, A_i, \ell_i)$ ($i=1,2$) are *strongly bisimilar*, notation $N_1 \approx_{pb} N_2$, if $A_1 = A_2$ and there is a linking l in a place bisimulation with $M_1 = \pi_1(l)$ and $M_2 = \pi_2(l)$.

It follows that $\approx_{pb}$ is finer than $\leftrightarrows_{sp}$, in the sense that place bisimilarity of two nets implies their structure preserving bisimilarity.

10.2 Occurrence net equivalence

Definitions of the *unfolding* for various classes of Petri nets into an *occurrence net* appear in [25,35,36,16,7,23,12]—I will not repeat them here. In all these cases, the definition directly implies that if an occurrence net $\mathcal{N}$ results from unfolding a net N then $\mathcal{N}$ is safe and there exists a folding of $\mathcal{N}$ into N (recall Definition 6) satisfying

- if $\mathcal{M}$ is a reachable marking of $\mathcal{N}$, and $t \in T$ is a transition of N with ${}^\bullet t \leq \rho(\mathcal{M})$ then there is a $\mathfrak{t} \in \mathcal{T}$ with $\rho(\mathfrak{t}) = t$.

Proposition 5. If such a folding from $\mathcal{N}$ to N exists, then $\mathcal{N} \leftrightarrows_{sp} N$.

Proof. The set of linkings $\mathscr{B} := \{\{(\mathfrak{s}, \rho(\mathfrak{s})) \mid \mathfrak{s} \in \mathcal{M}\} \mid \mathcal{M}$ a reachable marking of $\mathcal{N}\}$ is an sp-bisimulation between $\mathcal{N}$ and N. Checking this is trivial. □

Two nets N_1 and N_2 are *occurrence net equivalent* [16] if they have isomorphic unfoldings. Since isomorphic nets are strongly bisimilar [27] and hence structure preserving bisimilar, it follows that occurrence net equivalence between nets is finer than structure preserving bisimilarity.

In [1] it is pointed out that the strong bisimilarity of Olderog "is not compatible with unfoldings": they show two nets that have isomorphic unfoldings, yet are not strongly bisimilar. However, when the net N is safe, the sp-bisimulation displayed in the proof of Proposition 5 is in fact a strong bisimulation, showing that each net is strongly bisimilar with its unfolding. This is compatible with the observation of [1] because of the non-transitivity of strong bisimilarity.

10.3 Causal equivalence

Causal equivalence is coarser than structure preserving bisimilarity.

Theorem 4. If $N_1 \leftrightarrows_{sp} N_2$ for nets N_1 and N_2, then $N_1 \equiv_{caus} N_2$.

Proof. By Theorem 3 there exists a process-based sp-bisimulation $\mathscr{R}$ between N_1 and N_2. $\mathscr{R}$ must contain a triple $(\rho_1^0, \mathcal{N}^0, \rho_2^0)$ with $\mathcal{N}^0$ a causal net containing no transitions. So $(\mathcal{N}^0, \rho_1^0)$ and $(\mathcal{N}^0, \rho_2^0)$ are initial processes of N_1 and N_2, respectively. The net $\mathcal{N}^0$ contains isolated places only, as many as the size of the initial markings of N_1 and N_2.

Let $\mathcal{N}$ be a causal net of N_1. I have to prove that $\mathcal{N}$ is also a causal net of N_2. Without loss of generality I may assume that $\mathcal{N}^0$ is a prefix of $\mathcal{N}$, as being a causal net of a given Petri net is invariant under renaming of its places and transitions.

So N_1 has a process $\mathcal{P}_1 = (\mathcal{N}, \rho_1)$. By Proposition 4, $\mathcal{P}_1$ is the limit of a chain $\mathcal{P}_1^0 \leq \mathcal{P}_1^1 \leq \mathcal{P}_1^2 \leq \ldots$ of finite processes of N_1. Moreover, for $\mathcal{P}_1^0$ one can take $(\mathcal{N}^0, \rho_1^0)$. Let $\mathcal{P}_1^i = (\mathcal{N}^i, \rho_1^i)$ for $i \in \mathbb{N}$. By induction on $i \in \mathbb{N}$, it now follows from the properties of a process-based sp-bisimulation that N_2 has processes $\mathcal{P}_2^{i+1} = (\mathcal{N}^{i+1}, \rho_2^{i+1})$, such that $(\mathcal{N}^i, \rho_2^i) \leq (\mathcal{N}^{i+1}, \rho_2^{i+1})$ and $(\rho_1^{i+1}, \mathcal{N}^{i+1}, \rho_2^{i+1}) \in \mathscr{R}$. Using Proposition 3, the limit $\mathcal{P}_2 = (\mathcal{N}, \rho_2)$ of this chain is a process of N_2, contributing the causal net $\mathcal{N}$. □

10.4 History preserving bisimilarity

The notion of *history preserving bisimilarity* was originally proposed in [32] under the name *behavior structure bisimilarity*, studied on event structures in [13], and first defined on Petri nets, under to the individual token interpretation, in [2], under the name *fully concurrent bisimulation* equivalence.

Definition 11. [2] Let $\mathcal{N}_i = (\mathcal{S}_i, \mathcal{T}_i, \mathcal{F}_i, \mathcal{M}_{0i}, \mathcal{A}_i, \ell_i)$ $(i = 1, 2)$ be two causal nets. An *order-isomorphism* between them is a bijection $\beta : \mathcal{T}_1 \to \mathcal{T}_2$ such that $\mathcal{A}_1 = \mathcal{A}_2$, $\ell_2(\beta(t)) = \ell_1(t)$ for all $t \in \mathcal{T}_1$, and $t \; \mathcal{F}_1^+ \; u$ iff $\beta(t) \; \mathcal{F}_2^+ \; \beta(u)$ for all $t, u \in T_1$.

Definition 12. [2] A *fully concurrent bisimulation* between two nets N_1 and N_2 is a set $\mathscr{R}$ of triples $((\rho_1, \mathcal{N}_1), \beta, (\mathcal{N}_2, \rho_2))$ with $(\mathcal{N}_i, \rho_i)$ a finite process of N_i, for $i = 1, 2$, and β an order-isomorphism between $\mathcal{N}_1$ and $\mathcal{N}_2$, such that

- $\mathscr{R}$ contains a triple $((\rho_1, \mathcal{N}_1), \beta, (\mathcal{N}_2, \rho_2))$ with $\mathcal{N}_1$ containing no transitions,
- if $(\mathcal{P}_1, \beta, \mathcal{P}_2) \in \mathscr{R}$ and $\mathcal{P}_i'$ with $i \in \{1, 2\}$ is a fin. proc. of N_i extending $\mathcal{P}_i$, then N_j with $j := 3 - i$ has a process $\mathcal{P}_j' \geq \mathcal{P}_j$ such that $(\mathcal{P}_1', \beta', \mathcal{P}_2') \in \mathscr{R}$ for some $\beta' \supseteq \beta$.

Write $N_1 \approx_h N_2$ or $N_1 \approx_{fcb} N_2$ iff such a bisimulation exists.

It follows immediately from the process-based characterisation of sp-bisimilarity in Section 9 that fully concurrent bisimilarity (or history preserving bisimilarity based on the individual token interpretation of nets) is coarser than sp-bisimilarity.

Theorem 5. If $N_1 \leftrightarrows_{sp} N_2$ for nets N_1 and N_2, then $N_1 \approx_{fcb} N_2$.

Proof. A process-based sp-bisimulation is simply a fully concurrent bisimulation with the extra requirement that β must be the identity relation. □

11 Inevitability for non-reactive systems

A run or execution of a system modelled as Petri net N can be formalised as a path of N (defined in Section 3) or a process of N (defined in Section 8). A path or process representing a complete run of the represented system—one that is not just the first part of a larger run—is sometimes called a *complete* path or process. Once a formal definition of a complete path or process is agreed upon, an action b is *inevitable* in a net N iff each complete path (or each complete process) of N contains a transition labelled b. In case completeness is defined both for paths and processes, the definitions ought to be such that they give rise to the same concept of inevitability.

The definition of which paths or processes count as being complete depends on two factors: (1) whether actions that a net can perform by firing a transition are fully under control of the represented system itself or (also) of the environment in which it will be running, and (2) what type of progress or fairness assumption one postulates to guarantee that actions that are due to occur will actually happen. In order to address (2) first, in this section I deal only with nets in which all activity is fully under control of the represented system. In Section 14 I will generalise the conclusions to reactive systems.

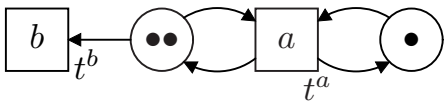
When making no progress or fairness assumptions, a system always has the option not to progress further, and all paths and all processes are complete—in particular initial paths and processes, containing no transitions. Consequently, no action is inevitable in any net, so each semantic equivalence respects inevitability.

When assuming progress, but not justness or fairness, any infinite path or process is complete, and a finite path or process is complete iff it is maximal, in the sense that it has no proper extension. In this setting, interleaving bisimilarity, and hence also each finer equivalence, respects inevitability. The argument is that an interleaving bisimulation induces a relation between the paths of two related nets N_1 and N_2, such that

- each path of N_1 is related to a path of N_2 and vice versa,
- if two paths are related, either both or neither contain a transition labelled b,
- if two paths are related, either both or neither of them are complete.

In the rest of this paper I will assume justness, and hence also progress, but not (weak or strong) fairness, as explained in Section 1.4. In this setting a process is *just* or *complete*[4] iff it is maximal, in the sense that it has no proper extension.

Example 3. The net depicted on the right has a complete process performing the action a infinitely often, but never the action b. It consumes each token that is initially present or stems from any firing of the transition t^a. Hence b is not inevitable. This fits with the intuition that if a transition occurrence is perpetually enabled it will eventually happen—but only when strictly adhering to the individual token interpretation of nets. Under this interpretation, each firing of t^b using a particular token is a different transition occurrence. It is possible to schedule an infinite sequence of as in such a way that none such transition occurrence is perpetually enabled from some point onwards.



[4] The term "complete" is meant to vary with the choice of a progress or fairness assumption; when assuming only justness, it is set to the value "just".

When adhering to the collective token interpretation of nets, the action b could be considered inevitable, as in any execution scheduling as only, transition t^b is perpetually enabled. Since my structure preserving bisimulation fits within the individual token interpretation, here one either should adhere to that interpretation, or restrict attention to safe nets, where there is no difference between both interpretations.

12 History preserving bisimilarity does not respect inevitability

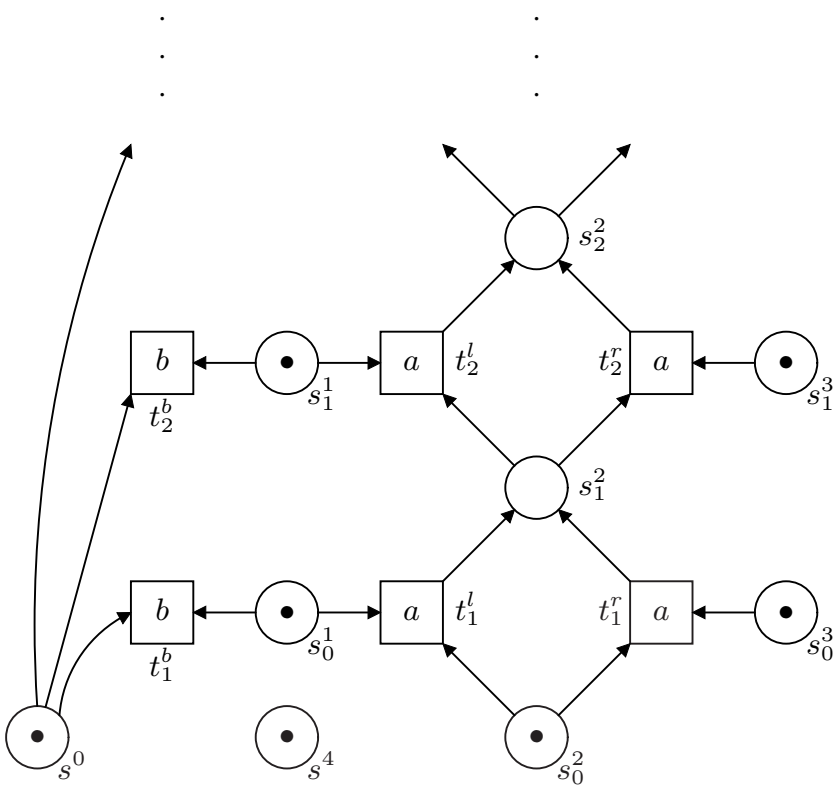


Fig. 5. A net in which the action b is not inevitable

Consider the safe net N_1 depicted in Figure 5, and the net N_2 obtained from N_1 by exchanging for any transition t_i^b ($i{>}0$) the preplace s_{i-1}^1 for s^4. The net N_2 performs in parallel an infinite sequence of a-transitions (where at each step $i{>}0$ there is a choice between t_i^l and t_i^r) and a single b-transition (where there is a choice between t_i^b for $i{>}0$). In N_2 the action b is inevitable. In N_1, on the other hand, b is not inevitable, for the run of N_1 in which t_i^l is chosen over t_i^r for all $i{>}0$ is complete, and cannot be extended which a b-transition. Thus, each semantic equivalence that equates N_1 and N_2 fails to respect inevitability.

Theorem 6. Causal equivalence does not respect inevitability.

Proof. $N_1 \equiv_{caus} N_2$, because both nets have the same causal nets. One of these nets is depicted in Figure 6; the others are obtained by omitting the b-transition, and/or omitting all but a finite prefix of the a-transitions. □

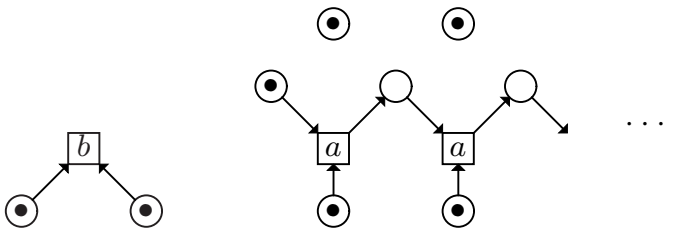


Fig. 6. A causal net of N_1 and N_2

Theorem 7. History preserving bisimilarity does not respect inevitability.

Proof. Recall that N_1 and N_2 differ only in their flow relations, and have the same set of transitions. I need to describe a fully concurrent bisimulation $\mathscr{R}$ between N_1 and N_2. $\mathscr{R}$ consists of a set of triples, each consisting of a process of N_1, a related process of N_2, and an order isomorphism between them. First of all I include all triples $(\mathcal{P}_1, \beta, \mathcal{P}_2)$ where $\mathcal{P}_1$ is an arbitrary process of N_1, $\mathcal{P}_2$ is the unique process of N_2 that induces the same set of transitions as $\mathcal{P}_1$, and β relates transition of $\mathcal{P}_1$ and $\mathcal{P}_2$ when they map to the same transition of N_i (i=1, 2). Secondly, I include all triples $(\mathcal{P}_1, \beta, \mathcal{P}_2)$ where $\mathcal{P}_2$ is an arbitrary process of N_2 inducing both t^b_k and t^l_k for some k>0, and $\mathcal{P}_1$ is any process of N_1 that induces the same transitions as $\mathcal{P}_2$ except that, for some $h \geq k$ the induced transition t^l_h, if present, is replaced by t^r_h, and t^b_k is replaced by t^b_h. (β should be obvious.) It is trivial to check that the resulting relation is a fully concurrent bisimulation indeed. □

13 Structure preserving bisimilarity respects inevitability

Definition 13. A net $\mathcal{N}$ is called a *complete* causal net of a net N if it is the first component of a maximal process $(\mathcal{N}, \rho)$ of N. Two nets N_1 and N_2 are *complete causal net equivalent*, notation $\equiv_{cc}$, if they have the same complete causal nets.

Since the causal nets of a net N are completely determined by the complete causal nets of N, namely as their prefixes, $N_1 \equiv_{cc} N_2$ implies $N_1 \equiv_{caus.} N_2$. It follows immediately from the definition of inevitability that $\equiv_{cc}$ respects inevitability. Thus, to prove that $\leftrightarrows_{sp}$ respects inevitability it suffices to show that $\leftrightarrows_{sp}$ is finer than $\equiv_{cc}$.

Theorem 8. If $N_1 \leftrightarrows_{sp} N_2$ for nets N_1 and N_2, then $N_1 \equiv_{cc} N_2$.

Proof. Suppose $N_1 \leftrightarrows_{sp} N_2$. By Theorem 3 there exists a process-based sp-bisimulation $\mathscr{R}$ between N_1 and N_2. $\mathscr{R}$ must contain a triple $(\rho^0_1, \mathcal{N}^0, \rho^0_2)$ with $\mathcal{N}^0$ a causal net containing no transitions. So $(\mathcal{N}^0, \rho^0_1)$ and $(\mathcal{N}^0, \rho^0_2)$ are initial processes of N_1 and N_2, respectively. The net $\mathcal{N}^0$ contains isolated places only.

Let $\mathcal{N}$ be a complete causal net of N_1. I have to prove that $\mathcal{N}$ is also a complete causal net of N_2. Without loss of generality I may assume that $\mathcal{N}^0$ is a prefix of $\mathcal{N}$, as being a complete causal net of a given Petri net is invariant under renaming of its places.

So N_1 has a complete process $\mathcal{P}_1 = (\mathcal{N}, \rho_1)$. By Proposition 4, $\mathcal{P}_1$ is the limit of a chain $\mathcal{P}^0_1 \leq \mathcal{P}^1_1 \leq \mathcal{P}^2_1 \leq \ldots$ of finite processes of N_1. Moreover, for $\mathcal{P}^0_1$ one

can take $(\mathcal{N}^0, \rho_1^0)$. Let $\mathcal{P}_1^i = (\mathcal{N}^i, \rho_1^i)$ for $i \in \mathbb{N}$. By induction on $i \in \mathbb{N}$, it now follows from the properties of a process-based sp-bisimulation that N_2 has processes $\mathcal{P}_2^{i+1} = (\mathcal{N}^{i+1}, \rho_2^{i+1})$, such that $(\mathcal{N}^i, \rho_2^i) \leq (\mathcal{N}^{i+1}, \rho_2^{i+1})$ and $(\rho_1^{i+1}, \mathcal{N}^{i+1}, \rho_2^{i+1}) \in \mathscr{R}$. Using Proposition 3, the limit $\mathcal{P}_2 = (\mathcal{N}, \rho_2)$ of this chain is a process of N_2. It remains to show that $\mathcal{P}_2$ is complete.

Towards a contradiction, let $\mathcal{P}_{2u} = (\mathcal{N}_u, \rho_{2u})$ be a proper extension of $\mathcal{P}_2$, say with just one transition, u. Then ${}^\bullet u \subseteq \mathcal{N}^\circ$. By the third requirement on occurrence nets of Definition 6, their are only finitely many transitions t with $(t, u) \in \mathcal{F}_{2u}^+$. Hence one of the finite approximations $\mathcal{N}^k$ of $\mathcal{N}$ contains all these transitions. So ${}^\bullet u \subseteq (\mathcal{N}^k)^\circ$. Let, for all $i \geq k$, $\mathcal{P}_{2u}^i = (\mathcal{N}_u^i, \rho_{2u}^i)$ be the finite prefix of $\mathcal{P}_{2u}$ that extends $\mathcal{P}_2^i$ with the single transition u. Then $\mathcal{P}_{2u}^i \leq \mathcal{P}_{2u}^{i+1}$ for all $i \geq k$, and the limit of the chain $\mathcal{P}_{2u}^k \leq \mathcal{P}_{2u}^{k+1} \leq \ldots$ is $\mathcal{P}_{2u}$. By induction on $i \in \mathbb{N}$, it now follows from the properties of a process-based sp-bisimulation that N_1 has processes $\mathcal{P}_{1u}^i = (\mathcal{N}_u^i, \rho_{1u}^i)$ for all $i \geq k$, such that $(\rho_{1u}^i, \mathcal{N}_u^i, \rho_{2u}^i) \in \mathscr{R}$, $(\mathcal{N}^k, \rho_1^k) \leq (\mathcal{N}_u^k, \rho_{1u}^k)$ and $(\mathcal{N}_u^i, \rho_{1u}^i) \leq (\mathcal{N}_u^{i+1}, \rho_{1u}^{i+1})$. Using Proposition 3, the limit $\mathcal{P}_{1u} = (\mathcal{N}_u, \rho_{1u})$ of this chain is a process of N_1. It extends $\mathcal{P}_1$ with the single transition u, contradicting the maximality of $\mathcal{P}_1$. □

14 Inevitability for reactive systems

In the modelling of reactive systems, an action performed by a net is typically a synchronisation between the net itself and its environment. Such an action can take place only when the net is ready to perform it, as well as its environment. In this setting, an adequate formalisation of the concepts of justness and inevitability requires keeping track of the set of actions that from some point onwards are blocked by the environment—e.g. because the environment is not ready to partake in the synchronisation. Such actions are not required to occur eventually, even when they are perpetually enabled by the net itself. Let's speak of a Y*-environment* if Y is this set of actions. In Section 11 I restricted attention to $\emptyset$-environments, in which an action can happen as soon as it is enabled by the net in question. In [15] a path is called Y-just iff, when assuming justness, it models a complete run of the represented system in a Y-environment. The below is a formalisation for this concept for Petri nets under the individual token interpretation.

Definition 14. A process of a net is Y*-just* or Y*-complete* it each of its proper extensions adds a transition with a label in Y.

Note that a just or complete process as defined in Section 11 is a $\emptyset$-just or $\emptyset$-complete process. In applications there often is a subset of actions that are known to be fully controlled by the system under consideration, and not by its environment. Such actions are often called *non-blocking*. A typical example from process algebra [24] is the internal action τ. In such a setting, Y-environments exists only for sets of actions $Y \subseteq \mathscr{C}$, where $\mathscr{C}$ is the set of all non-non-blocking actions.

A process of a net is *complete* if it models a complete run of the represented system in some environment. This is the case iff it is Y-complete for some set $Y \subseteq \mathscr{C}$, which is the case iff it is $\mathscr{C}$-complete.

In [34], non-blocking is a property of transitions rather than actions, and non-blocking transitions are called *hot*. Transitions that are not hot are *cold*, which inspired

my choice of the latter $\mathscr{C}$ above. In this setting, a process $\mathcal{P} = (\mathcal{N}, \rho)$ is complete iff the marking $\rho(\mathcal{N}^\circ)$ enables cold transitions only [34].

Definition 15. A action b is *Y-inevitable* in a net if each Y-complete process contains a transition labelled b. A semantic equivalence $\approx$ *respects Y-inevitability* if whenever $N_1 \approx N_2$ and b is Y-inevitable in N_1, then b is Y-inevitable in N_2. It *respects inevitability* iff it respects Y-inevitability for each $Y \subseteq \mathscr{C}$.

In Section 12 it is shown that $\equiv_{caus}$ and $\approx_h$ do not respect $\emptyset$-inevitability. From this it follows that they do not respect inevitability. In Section 13 it is shown that $\leftrightarrows_{sp}$ does respect $\emptyset$-inevitability. By means of a trivial adaptation the same proof shows that $\leftrightarrows_{sp}$ respects Y-inevitability, for arbitrary Y. All that is needed is to assume that the transition u in that proof has a label $\notin Y$. Thus $\leftrightarrows_{sp}$ respects inevitability.

15 Conclusion

This paper proposes a novel semantic equivalence for current systems represented as Petri nets: *structure preserving bisimilarity*. As a major application—the one that inspired this work—it is used to establish the agreement between the operational Petri net semantics of the process algebra CCSP as proposed by Olderog, and its denotational counterpart. An earlier semantic relation used for this purpose was Olderog's *strong bisimilarity* on safe Petri nets, but that relation failed to be transitive. I hereby conjecture that on the subclass of occurrence nets, strong bisimilarity and structure preserving bisimilarity coincide. If this it true, it follows, together with the observations of Section 6 that strong bisimilarity is included in structure preserving bisimilarity, and of Section 10.2 that each safe net is strongly bisimilar with its unfolding into an occurrence net, that on safe nets structure preserving bisimilarity is the transitive closure of strong bisimilarity.

Section 1.2 proposes nine requirements on a semantic equivalence that is used for purposes like the one above. I have shown that structure preserving bisimilarity meets eight of these requirements and conjecture that it meets the remaining one as well.

- It meets Requirement 1, that it respects branching time, as a consequence of Theorem 5, saying that it is finer than history preserving bisimilarity, which is known to be finer than interleaving bisimilarity.
- It meets Requirement 2, that it fully captures causality and concurrency (and their interplay with branching time),[5] also as a consequence of Theorem 5.
- It meets Requirement 3, that it respects inevitability (under the standard interpretation of Petri nets that assumes justness but not fairness),[5] as shown in Section 13.
- It meets Requirement 4, that it is real-time consistent, as a result of Theorem 5.
- I conjecture that it meets Requirement 5, that it is preserved under action refinement.
- It meets Requirement 6, that it is finer than causal equivalence, by Theorem 4.
- It meets Requirement 7, that it is coarser than $\equiv_{occ}$, as shown in Section 10.2.
- It meets Requirement 8, that it is a congruence for the CCSP operators, by Thm. 2.
- It meets Requirement 9, that it allows to establish agreement between the operational and denotational interpretations of CCSP operators, since it is coarser than Olderog's strong bisimilarity, as shown in Section 6.

[5] when taking the individual token interpretation of nets, or restricting attention to safe ones

Moreover, structure preserving bisimilarity is the first known equivalence that meets these requirements. In fact, it is the first that meets the key Requirements 3, 4, 7 and 9.

Acknowledgement My thanks to Ursula Goltz for proofreading and valuable feedback.

References

1. C. Autant, Z. Belmesk & P. Schnoebelen (1991): *Strong Bisimilarity on Nets Revisited*. In E.H.L. Aarts, J. van Leeuwen & M. Rem, editors: Proc. *PARLE '91*, Eindhoven, The Netherlands, 1991, LNCS 506, Springer, pp. 295–312, doi:10.1007/3-540-54152-7_71.
2. E. Best, R. Devillers, A. Kiehn & L. Pomello (1991): *Concurrent Bisimulations in Petri nets*. *Acta Informatica* 28, pp. 231–264, doi:10.1007/BF01178506.
3. S.D. Brookes, C.A.R. Hoare & A.W. Roscoe (1984): *A theory of communicating sequential processes*. *Journal of the ACM* 31(3), pp. 560–599, doi:10.1145/828.833.
4. L. Castellano, G. De Michelis & L. Pomello (1987): *Concurrency vs interleaving: an instructive example*. *Bulletin of the EATCS* 31, pp. 12–15.
5. P. Degano, R. De Nicola & U. Montanari (1987): *CCS is an (Augmented) Contact Free C/E System*. In M.V. Zilli, editor: *Mathematical Models for the Semantics of Parallelism*, LNCS 280, Springer, pp. 144–165, doi:10.1007/3-540-18419-8_13.
6. E.A. Emerson & E.M. Clarke (1982): *Using Branching Time Temporal Logic to Synthesize Synchronization Skeletons*. *Science of Computer Programming* 2(3), pp. 241–266, doi:10.1016/0167-6423(83)90017-5.
7. J. Engelfriet (1991): *Branching Processes of Petri Nets*. *Acta Informatica* 28(6), pp. 575–591, doi:10.1007/BF01463946.
8. A. Fehnker, R.J. van Glabbeek, P. Höfner, A.K. McIver, M. Portmann & W.L. Tan (2013): *A Process Algebra for Wireless Mesh Networks used for Modelling, Verifying and Analysing AODV*. Technical Report 5513, NICTA, Sydney, Australia. Available at http://arxiv.org/abs/1312.7645.
9. H. Genrich & E. Stankiewicz-Wiechno (1980): *A Dictionary of Some Basic Notions of Net Theory*. In W. Brauer, editor: *Advanced Course: Net Theory and Applications*, LNCS 84, Springer, pp. 519–531, doi:10.1007/3-540-10001-6_39.
10. R.J. van Glabbeek (1990): *The refinement theorem for ST-bisimulation semantics*. In M. Broy & C.B. Jones, editors: Proceedings IFIP TC2 Working Conference on *Programming Concepts and Methods*, Sea of Gallilee, Israel, April 1990, North-Holland, pp. 27–52.
11. R.J. van Glabbeek (2001): *The Linear Time – Branching Time Spectrum I; The Semantics of Concrete, Sequential Processes*. In J.A. Bergstra, A. Ponse & S.A. Smolka, editors: *Handbook of Process Algebra*, chapter 1, Elsevier, pp. 3–99, doi:10.1016/B978-044482830-9/50019-9.
12. R.J. van Glabbeek (2005): *The Individual and Collective Token Interpretations of Petri Nets*. In M. Abadi & L. de Alfaro, editors: Proc. *CONCUR 2005*, San Francisco, USA, August 2005, LNCS 3653, Springer, pp. 323–337, doi:10.1007/11539452_26.
13. R.J. van Glabbeek & U. Goltz (2001): *Refinement of Actions and Equivalence Notions for Concurrent Systems*. *Acta Informatica* 37, pp. 229–327, doi:10.1007/s002360000041.
14. R.J. van Glabbeek & P. Höfner (2015): *CCS: It's not fair!* *Acta Informatica* 52(2-3), pp. 175–205, doi:10.1007/s00236-015-0221-6.
15. R.J. van Glabbeek & P. Höfner (2015): *Progress, Fairness and Justness in Process Algebra*. Available at http://arxiv.org/abs/1501.03268.
16. R.J. van Glabbeek & F.W. Vaandrager (1987): *Petri net models for algebraic theories of concurrency (extended abstract)*. In J.W. de Bakker, A.J. Nijman & P.C. Treleaven, editors: Proc. *PARLE*, LNCS 259, Springer, pp. 224–242, doi:10.1007/3-540-17945-3_13.

17. R.J. van Glabbeek & F.W. Vaandrager (1997): *The Difference Between Splitting in n and $n+1$*. Information and Comput. 136(2), pp. 109–142, doi:10.1006/inco.1997.2634.
18. U. Goltz & A. Mycroft (1984): *On the relationship of CCS and Petri nets*. In J. Paredaens, editor: *Proceedings 11^{th} ICALP*, Antwerpen, LNCS 172, Springer, pp. 196–208, doi:10.1007/3-540-13345-3_18.
19. U. Goltz & W. Reisig (1983): *The Non-Sequential Behaviour of Petri Nets*. Information and Control 57(2-3), pp. 125–147, doi:10.1016/S0019-9958(83)80040-0.
20. C.A.R. Hoare (1985): *Communicating Sequential Processes*. Prentice Hall, Englewood Cliffs.
21. R. Loogen & U. Goltz (1991): *Modelling nondeterministic concurrent processes with event structures*. Fundamenta Informaticae 14(1), pp. 39–74.
22. A.W. Mazurkiewicz, E. Ochmanski & W. Penczek (1989): *Concurrent Systems and Inevitability*. TCS 64(3), pp. 281–304, doi:10.1016/0304-3975(89)90052-2.
23. J. Meseguer, U. Montanari & V. Sassone (1997): *On the semantics of place/transition Petri nets*. Mathematical Structures in Computer Science 7(4), pp. 359–397, doi:10.1017/S0960129597002314.
24. R. Milner (1990): *Operational and algebraic semantics of concurrent processes*. In J. van Leeuwen, editor: *Handbook of Theoretical Computer Science*, chapter 19, Elsevier Science Publishers B.V. (North-Holland), pp. 1201–1242. Alternatively see *Communication and Concurrency*, Prentice-Hall, Englewood Cliffs, 1989, of which an earlier version appeared as *A Calculus of Communicating Systems*, LNCS 92, Springer, 1980, doi:10.1007/3-540-10235-3.
25. M. Nielsen, G.D. Plotkin & G. Winskel (1981): *Petri nets, event structures and domains, part I*. TCS 13(1), pp. 85–108, doi:10.1016/0304-3975(81)90112-2.
26. E.-R. Olderog (1987): *Operational Petri net semantics for CCSP*. In G. Rozenberg, editor: *Advances in Petri Nets 1987*, covers the 7th European Workshop on *Applications and Theory of Petri Nets*, Oxford, UK, June 1986, LNCS 266, Springer, pp. 196–223, doi:10.1007/3-540-18086-9_27.
27. E.-R. Olderog (1991): *Nets, Terms and Formulas: Three Views of Concurrent Processes and their Relationship*. Cambridge Tracts in Theor. Comp. Sc. 23, Cambridge University Press.
28. E.-R. Olderog & C.A.R. Hoare (1986): *Specification-oriented semantics for communicating processes*. Acta Informatica 23, pp. 9–66, doi:10.1007/BF00268075.
29. C.A. Petri (1977): *Non-sequential processes*. Internal Report GMD-ISF-77.05, GMD, St. Augustin.
30. G.D. Plotkin (2004): *A Structural Approach to Operational Semantics*. The Journal of Logic and Algebraic Programming 60–61, pp. 17–139, doi:10.1016/j.jlap.2004.05.001. Originally appeared in 1981.
31. A. Pnueli (1977): *The Temporal Logic of Programs*. In: *Foundations of Computer Science (FOCS '77)*, IEEE, pp. 46–57, doi:10.1109/SFCS.1977.32.
32. A. Rabinovich & B.A. Trakhtenbrot (1988): *Behavior Structures and Nets*. Fundamenta Informaticae 11(4), pp. 357–404.
33. W. Reisig (1985): *Petri nets – an introduction*. EATCS Monographs on Theoretical Computer Science, Volume 4, Springer, doi:10.1007/978-3-642-69968-9.
34. W. Reisig (2013): *Understanding Petri Nets - Modeling Techniques, Analysis Methods, Case Studies*. Springer, doi:10.1007/978-3-642-33278-4.
35. G. Winskel (1984): *A new definition of morphism on Petri nets*. In M. Fontet & K. Mehlhorn, editors: *Proceedings STACS 84*, LNCS 166, Springer, pp. 140–150, doi:10.1007/3-540-12920-0_13.
36. G. Winskel (1987): *Event structures*. In W. Brauer, W. Reisig & G. Rozenberg, editors: *Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986, Part II; Proceedings of an Advanced Course*, Good Honnef, September 1986, LNCS 255, Springer, pp. 325–392, doi:10.1007/3-540-17906-2_31.