On Synchronous and Asynchronous Interaction in Distributed Systems

Rob van Glabbeek
NICTA, Sydney, Australia
University of New South Wales, Sydney, Australia
rvg@cs.stanford.edu

Ursula Goltz
Institute for Programming and Reactive Systems
TU Braunschweig, Braunschweig, Germany
goltz@ips.cs.tu-bs.de

Jens-Wolfhard Schicke*
Institute for Programming and Reactive Systems
TU Braunschweig, Braunschweig, Germany
drahflow@gmx.de

When considering distributed systems, it is a central issue how to deal with interactions between components. In this paper, we investigate the paradigms of synchronous and asynchronous interaction in the context of distributed systems. We investigate to what extent or under which conditions synchronous interaction is a valid concept for specification and implementation of such systems. We choose Petri nets as our system model and consider different notions of distribution by associating locations to elements of nets. First, we investigate the concept of simultaneity which is inherent in the semantics of Petri nets when transitions have multiple input places. We assume that tokens may only be taken instantaneously by transitions on the same location. We exhibit a hierarchy of 'asynchronous' Petri net classes by different assumptions on possible distributions. Alternatively, we assume that the synchronisations specified in a Petri net are crucial system properties. Hence transitions and their preplaces may no longer placed on separate locations. We then answer the question which systems may be implemented in a distributed way without restricting concurrency, assuming that locations are inherently sequential. It turns out that in both settings we find semi-structural properties of Petri nets describing exactly the problematic situations for interactions in distributed systems.

1 Introduction

In this paper, we address interaction patterns in distributed systems. By a distributed system we understand here a system which is executed on spatially distributed locations, which do not share a common clock (for performance reasons for example). We want to investigate to what extent or under which conditions synchronous interaction is a valid concept for specification and implementation of such systems. It is for example a well-known fact that synchronous communication can be simulated by asynchronous communication using suitable protocols. However, the question is whether and under which circumstances these protocols fully retain the original behaviour of a system. What we are interested in here are precise descriptions of what behaviours can possibly be preserved and which cannot.

The topic considered here is by no means a new one. We give a short overview on related approaches in the following.

Already in the 80th, Luc Bougé considered a similar problem in the context of distributed algorithms. In [5] he considers the problem of implementing symmetric leader election in the sublanguages of CSP obtained by allowing different forms of communication, combining input and output guards in guarded

*This paper was partially written during a four month stay of J.-W. Schicke at NICTA, during which he was supported by DAAD (Deutscher Akademischer Austauschdienst) and NICTA.

choice in different ways. He finds that the possibility of implementing leader election depends heavily on the structure of the communication graphs. Truly symmetric schemes are only possible in CSP with arbitrary input and output guards in choices.

Synchronous interaction is a basic concept in many languages for system specification and design, e.g. in statechart-based approaches, in process algebras or the π-calculus. For process algebras and the π-calculus, language hierarchies have been established which exhibit the expressive power of different forms of synchronous and asynchronous interaction. In [4] Frank de Boer and Catuscia Palamidessi consider various dialects of CSP with differing degrees of asynchrony. Similar work is done for the π-calculus in [15] by Catuscia Palamidessi, in [13] by Uwe Nestmann and in [8] by Dianele Gorla. A rich hierarchy of asynchronous π-calculi has been mapped out in these papers. Again mixed-choice, i.e. the ability to combine input and output guards in a single choice, plays a central rôle in the implementation of truly synchronous behaviour.

In [17], Peter Selinger considers labelled transition systems whose visible actions are partitioned into input and output actions. He defines asynchronous implementations of such a system by composing it with in- and output queues, and then characterises the systems that are behaviourally equivalent to their asynchronous implementations. The main difference with our approach is that we focus on asynchrony within a system, whereas Selinger focusses on the asynchronous nature of the communications of a system with the outside world.

Also in hardware design it is an intriguing quest to use interaction mechanisms which do not rely on a global clock, in order to gain performance. Here the simulation of synchrony by asynchrony can be a crucial issue, see for instance [10] and [11].

In contrast to the approaches based on language constructs like the work on CSP or the π-calculus, we choose here a very basic system model for our investigations, namely Petri nets. The main reason for this choice is the detailed way in which a Petri net represents a concurrent system, including the interaction between the components it may consist of. In an interleaving based model of concurrency such as labelled transition systems modulo bisimulation semantics, a system representation as such cannot be said to contain synchronous or asynchronous interaction; at best these are properties of composition operators, or communication primitives, defined in terms of such a model. A Petri net on the other hand displays enough detail of a concurrent system to make the presence of synchronous communication discernible. This makes it possible to study synchronous and asynchronous interaction without digressing to the realm of composition operators.

Also in Petri net theory, the topic which concerns us here has already been tackled. It has been investigated in [9] and [18] whether and how a Petri net can be implemented in a distributed way. We will comment on these and other related papers in the area of Petri net theory in the conclusion.

In a Petri net, a transition interacts with its preplaces by consuming tokens. In Petri net semantics, taking a token is usually considered as an instantaneous action, hence a synchronous interaction between a transition and its preplace. In particular when a transition has several preplaces this becomes a crucial issue. In this paper we investigate what happens if we consider a Petri net as a specification of a system that is to be implemented in a distributed way. For this we introduce locations on which all elements of a Petri net have to be placed upon. The basic assumption is that interaction between remote components takes time. In our framework this means that the removal of a token will be considered instantaneous only if the removing transition and the place where the token is removed from are co-located. Our investigations are now twofold.

In Section 3 of this paper, we consider under which circumstances the synchronous interaction between a transition and its preplace may be mimicked asynchronously, thus allowing to put places and their posttransitions on different locations. Following [6], we model the asynchronous interaction between transitions and their preplaces by inserting silent (unobservable) transitions between them. We investigate the effect of this transformation by comparing the behaviours of nets before and after insertion of the silent transitions using a suitable equivalence notion. We believe that most of our results are independent of the precise choice of this equivalence. However, as explained in Section 5, it has to preserve causality, branching time and divergence to some small extent, and needs to abstract from silent transitions. Therefore we choose one such equivalence, based on its technical convenience in establishing our results. Our choice is *step readiness equivalence*. It is a variant of the *readiness equivalence* of [14], obtained by collecting the set of *steps* of multiple actions possible after a certain sequence of actions, instead of just the set of possible actions. We call a net *asynchronous* if, for a suitable placement of its places and transitions, the above-mentioned transformation replacing synchronous by asynchronous interaction preserves step readiness equivalence. Depending on the allowed placements, we obtain a hierarchy of classes of asynchronous nets: *fully asynchronous* nets, *symmetrically asynchronous* nets and *asymmetrically asynchronous* nets. We give semi-structural properties that characterise precisely when a net falls into one of these classes. This puts the results from [6] in a uniform framework and extends them by introducing a simpler notion of asymmetric asynchrony.

In Sections 4 and 5 we pursue an alternative approach. We assume that the synchronisations specified in a Petri net are crucial system properties. Hence we enforce co-locality between a transition and all its preplaces while at the same time assuming that concurrent activity is not possible at a single location. We call nets fulfilling these requirement *distributed* and investigate which behaviours can be implemented by distributed nets. Again we compare the behaviours up to step readiness equivalence. We call a net *distributable* iff its behaviour can be equivalently produced by a distributed net. We give a behavioural and a semi-structural characterisation of a class of non-distributable nets, thereby exhibiting behaviours which cannot be implemented in a distributed way at all. Finally, we give a lower bound of distributability by providing a concrete distributed implementation for a wide range of nets.

An extended abstract of this paper will appear in the proceedings of the 33rd International Symposium on *Mathematical Foundations of Computer Science* (MFCS 2008), Toruń, Poland, August 2008 (E. Ochmański & J. Tyszkiewicz, eds.), LNCS 5162, Springer, 2008, pp. 16-35.

2 Basic Notions

We consider here 1-safe net systems, i.e. places never carry more than one token, but a transition can fire even if pre- and postset intersect.

Definition 1 Let Act be a set of *visible actions* and $\tau \notin \mathrm{Act}$ be an *invisible action*.
A *labelled net* (over Act) is a tuple $N = (S, T, F, M_0, \ell)$ where
- S is a set (of *places*),
- T is a set (of *transitions*),
- $F \subseteq S \times T \cup T \times S$ (the *flow relation*),
- $M_0 \subseteq S$ (the *initial marking*) and
- $\ell : T \to \mathrm{Act} \mathbin{\dot{\cup}} \{\tau\}$ (the *labelling function*).

Petri nets are depicted by drawing the places as circles, the transitions as boxes containing the respective label, and the flow relation as arrows (*arcs*) between them. When a Petri net represents a concurrent system, a global state of such a system is given as a *marking*, a set of places, the initial state being M_0. A marking is depicted by placing a dot (*token*) in each of its places. The dynamic behaviour of the represented system is defined by describing the possible moves between markings. A marking M may evolve into a marking M' when a nonempty set of transitions G *fires*. In that case, for each arc $(s,t) \in F$ leading to a transition t in G, a token moves along that arc from s to t. Naturally, this can happen only if all these tokens are available in M in the first place. These tokens are consumed by the firing, but also new tokens are created, namely one for every outgoing arc of a transition in G. These end up in the places at the end of those arcs. A problem occurs when as a result of firing G multiple tokens end up in the same place. In that case M' would not be a marking as defined above. In this paper we restrict attention to nets in which this never happens. Such nets are called *1-safe*. Unfortunately, in order to formally define this class of nets, we first need to correctly define the firing rule without assuming 1-safety. Below we do this by forbidding the firing of sets of transitions when this might put multiple tokens in the same place.

Definition 2 Let $N = (S,T,F,M_0,\ell)$ be a labelled net. Let $M_1, M_2 \subseteq S$.

We denote the preset and postset of a net element $x \in S \cup T$ by ${}^\bullet x := \{y \mid (y,x) \in F\}$ and $x^\bullet := \{y \mid (x,y) \in F\}$ respectively. These functions are extended to sets in the usual manner, i.e. ${}^\bullet X := \{y \mid y \in {}^\bullet x,\ x \in X\}$.

A nonempty set of transitions $\emptyset \neq G \subseteq T$, is called a *step from M_1 to M_2*, notation $M_1\ [G\rangle_N\ M_2$, if

- all transitions contained in G are *enabled*, that is
$$\forall t \in G.\ {}^\bullet t \subseteq M_1 \wedge (M_1 \setminus {}^\bullet t) \cap t^\bullet = \emptyset\ ,$$
- all transitions of G are *independent*, that is *not conflicting*:
$$\forall t,u \in G, t \neq u.\ {}^\bullet t \cap {}^\bullet u = \emptyset \wedge t^\bullet \cap u^\bullet = \emptyset\ ,$$
- in M_2 all tokens have been removed from the *preplaces* of G and new tokens have been inserted at the *postplaces* of G:
$$M_2 = (M_1 \setminus {}^\bullet G) \cup G^\bullet\ .$$

To simplify statements about possible behaviours of nets, we use some abbreviations.

Definition 3 Let $N = (S,T,F,M_0,\ell)$ be a labelled net.

We extend the labelling function ℓ to (multi)sets elementwise.

$\longrightarrow_N \subseteq \mathcal{P}(S) \times \mathbb{N}^{\mathrm{Act}} \times \mathcal{P}(S)$ is given by $M_1 \xrightarrow{A}_N M_2 \Leftrightarrow \exists\, G \subseteq T.\ M_1\ [G\rangle_N\ M_2 \wedge A = \ell(G)$

$\xrightarrow{\tau}_N \subseteq \mathcal{P}(S) \times \mathcal{P}(S)$ is defined by $M_1 \xrightarrow{\tau}_N M_2 \Leftrightarrow \exists t \in T.\ \ell(t) = \tau \wedge M_1\ [\{t\}\rangle_N\ M_2$

$\Longrightarrow_N \subseteq \mathcal{P}(S) \times \mathrm{Act}^* \times \mathcal{P}(S)$ is defined by $M_1 \xRightarrow{a_1 a_2 \cdots a_n}_N M_2 \Leftrightarrow$
$$M_1 \xrightarrow{\tau}{}^*_N \xrightarrow{\{a_1\}}_N \xrightarrow{\tau}{}^*_N \xrightarrow{\{a_2\}}_N \xrightarrow{\tau}{}^*_N \cdots \xrightarrow{\tau}{}^*_N \xrightarrow{\{a_n\}}_N \xrightarrow{\tau}{}^*_N M_2$$
where $\xrightarrow{\tau}{}^*_N$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}_N$.

We write $M_1 \xrightarrow{A}_N$ for $\exists M_2.\ M_1 \xrightarrow{A}_N M_2$, $M_1 \not\xrightarrow{A}_N$ for $\nexists M_2.\ M_1 \xrightarrow{A}_N M_2$ and similar for the other two relations. Likewise $M_1[G\rangle_N$ abbreviates $\exists M_2.\ M_1[G\rangle_N M_2$.

A marking M_1 is said to be *reachable* iff there is a $\sigma \in \mathrm{Act}^*$ such that $M_0 \xRightarrow{\sigma}_N M_1$. The set of all reachable markings is denoted by $[M_0\rangle_N$.

We omit the subscript N if clear from context.

As said before, here we only want to consider 1-safe nets. Formally, we restrict ourselves to *contact-free nets*, where in every reachable marking $M_1 \in [M_0\rangle$ for all $t \in T$ with ${}^\bullet t \subseteq M_1$
$$(M_1 \setminus {}^\bullet t) \cap t^\bullet = \emptyset\ .$$

For such nets, in Definition 2 we can just as well consider a transition t to be enabled in M iff ${}^\bullet t \subseteq M$, and two transitions to be independent when ${}^\bullet t \cap {}^\bullet u = \emptyset$.

In this paper we furthermore restrict attention to nets for which ${}^\bullet t \neq \emptyset$ and ${}^\bullet t$ and $t^\bullet$ are finite for all $t \in T$ and $s^\bullet$ is finite for all $s \in S$. We also require the initial marking M_0 to be finite. A consequence of these restrictions is that all reachable markings are finite, and it can never happen that infinitely many independent transitions are enabled. Henceforth, with *net* we mean a labelled net obeying the above restrictions.

In our nets transitions are labelled with *actions* drawn from a set $\mathrm{Act} \mathbin{\dot\cup} \{\tau\}$. This makes it possible to see these nets as models of *reactive systems*, that interact with their environment. A transition t can be thought of as the occurrence of the action $\ell(t)$. If $\ell(t) \in \mathrm{Act}$, this occurrence can be observed and influenced by the environment, but if $\ell(t) = \tau$, t is an *internal* or *silent* transition whose occurrence cannot be observed or influenced by the environment. Two transitions whose occurrences cannot be distinguished by the environment are equipped with the same label. In particular, given that the environment cannot observe the occurrence of internal transitions at all, all of them have the same label, namely τ.

We use the term *plain nets* for nets where ℓ is injective and no transition has the label τ, i.e. essentially unlabelled nets. Similarly, we speak of *plain τ-nets* to describe nets where $\ell(t) = \ell(u) \neq \tau \Rightarrow t = u$, i.e. nets where every observable action is produced by a unique transition. In this paper we focus on plain nets, and give semi-structural characterisations of classes of plain nets only. However, in defining whether a net belongs to one of those classes, we study its implementations, which typically are plain τ-nets. When proving our impossibility result (Theorem 3 in Section 5) we even allow arbitrary nets as implementations.

We use the following variation of readiness semantics [14] to compare the behaviour of nets.

Definition 4 Let $N = (S, T, F, M_0, \ell)$ be a net, $\sigma \in \mathrm{Act}^*$ and $X \subseteq \mathbb{N}^{\mathrm{Act}}$.
$<\sigma, X>$ is a *step ready pair* of N iff

$$\exists M.\ M_0 \stackrel{\sigma}{\Longrightarrow} M \wedge M \stackrel{\tau}{\not\longrightarrow} \wedge X = \{A \in \mathbb{N}^{\mathrm{Act}} \mid M \stackrel{A}{\longrightarrow}\}.$$

We write $\mathscr{R}(N)$ for the set of all step ready pairs of N.
Two nets N and N' are *step readiness equivalent*, $N \approx_{\mathscr{R}} N'$, iff $\mathscr{R}(N) = \mathscr{R}(N')$.

The elements of a set X as above are multisets of actions, but as in all such multisets that will be mentioned in this paper the multiplicity of each action occurrence is at most 1, we use set notation to denote them.

3 Asynchronous Petri Net Classes

In Petri nets, an inherent concept of simultaneity is built in, since when a transition has more than one preplace, it can be crucial that tokens are removed instantaneously. When using a Petri net to model a system which is intended to be implemented in a distributed way, this built-in concept of synchronous interaction may be problematic.

In this paper, a given net is regarded as a *specification* of how a system should behave, and this specification involves complete synchronisation of the firing of a transition and the removal of all tokens from its preplaces. In this section, we propose various definitions of an *asynchronous implementation* of a net N, in which such synchronous interaction is wholly or partially ruled out and replaced by asynchronous interaction. The question to be clarified is whether such an asynchronous implementation faithfully mimics the dynamic behaviour of N. If this is the case, we call the net N *asynchronous* with respect to the chosen interaction pattern.

The above programme, and thus the resulting concept of asynchrony, is parametrised by the answers to three questions:

1. Which synchronous interactions do we want to rule out exactly?
2. How do we replace synchronous by asynchronous interaction?
3. When does one net faithfully mimic the dynamic behaviour of another?

To answer the first question we associate a *location* to each place and each transition in a net. A transition may take a token instantaneously from a preplace (when firing) iff this preplace is co-located with the transition; if the preplace resides on a different location than the transition, we have to assume the collection of the token takes time, and thus the place looses its token *before* the transition fires.

We model the association of locations to the places and transitions in a net $N = (S, T, F, M_0, \ell)$ as a function $D : S \cup T \to \text{Loc}$, with Loc a set of possible locations. We refer to such a function as a *distribution* of N. Since the identity of the locations is irrelevant for our purposes, we can just as well abstract from Loc and represent D by the equivalence relation $\equiv_D$ on $S \cup T$ given by $x \equiv_D y$ iff $D(x) = D(y)$.

In this paper we do not deal with nets that have a distribution built in. We characterise the interaction patterns we are interested in by imposing particular restrictions on the allowed distributions. The implementor of a net can choose any distribution that satisfies the chosen requirements, and we call a net asynchronous for a certain interaction pattern if it has a correct asynchronous implementation based on any distribution satisfying the respective requirements.

The *fully asynchronous* interaction pattern is obtained by requiring that all places and all transitions reside on different locations. This makes it necessary to implement the removal of every token in a time-consuming way. However, this leads to a rather small class of asynchronous nets, that falls short for many applications. We therefore propose two ways to loosen this requirement, thereby building a hierarchy of classes of asynchronous nets. Both require that all places reside on different locations, but a transition may be co-located with one of its preplaces. The *symmetrically asynchronous* interaction pattern allows this only for transitions with a single preplace, whereas in the *asymmetrically asynchronous* interaction pattern any transition may be co-located with one of its preplaces. Since two preplaces can never be co-located, this breaks the symmetry between the preplaces of a transition; an implementor of a net has to choose at most one preplace for every transition, and co-locate the transition with it. The removal of tokens from all other preplaces needs to be implemented in a time-consuming way. Note that all three interaction patterns break the synchronisation of the token removal between the various preplaces.

Definition 5 Let D be a distribution on a net $N = (S, T, F, M_0, \ell)$, and let $\equiv_D$ be the induced equivalence relation on $S \cup T$. We say that D is

- *fully distributed*, $D \in \mathscr{Q}_{\text{FD}}$, when $x \equiv_D y$ for $x, y \in S \cup T$ only if $x = y$,
- *symmetrically distributed*, $D \in \mathscr{Q}_{\text{SD}}$, when

$$\begin{array}{lll} p \equiv_D q & \text{for } p, q \in S & \text{only if } p = q, \\ t \equiv_D p & \text{for } t \in T,\ p \in S & \text{only if } {}^\bullet t = \{p\} \text{ and} \\ t \equiv_D u & \text{for } t, u \in T & \text{only if } t = u \text{ or } \exists p \in S.\ t \equiv_D p \equiv_D u, \end{array}$$

- *asymmetrically distributed*, $D \in \mathscr{Q}_{\text{AD}}$, when

$$\begin{array}{lll} p \equiv_D q & \text{for } p, q \in S & \text{only if } p = q, \\ t \equiv_D p & \text{for } t \in T,\ p \in S & \text{only if } p \in {}^\bullet t \text{ and} \\ t \equiv_D u & \text{for } t, u \in T & \text{only if } t = u \text{ or } \exists p \in S.\ t \equiv_D p \equiv_D u. \end{array}$$

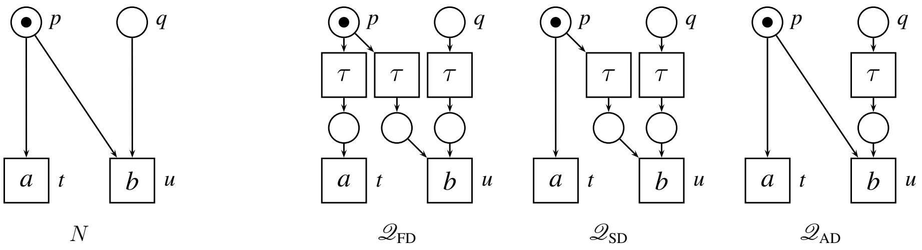

Figure 1: Possible results for $I_D(N)$ given different requirements

The second question raised above was: How do we replace synchronous by asynchronous interaction? In this section we assume that if an arc goes from a place s to a transition t at a different location, a token takes time to move from s to t. Formally, we describe this by inserting silent (unobservable) transitions between transitions and their remote preplaces. This leads to the following notion of an asynchronous implementation of a net with respect to a chosen distribution.

Definition 6 Let $N = (S, T, F, M_0, \ell)$ be a net, and let $\equiv_D$ be an equivalence relation on $S \cup T$.
The *D-based asynchronous implementation* of N is $I_D(N) := (S \cup S^\tau, T \cup T^\tau, F', M_0, \ell')$ with

$$\begin{aligned}
S^\tau &:= \{s_t \mid t \in T,\ s \in {}^\bullet t,\ s \not\equiv_D t\}\ ,\\
T^\tau &:= \{t_s \mid t \in T,\ s \in {}^\bullet t,\ s \not\equiv_D t\}\ ,\\
F' &:= \{(t,s) \mid t \in T,\ s \in t^\bullet\} \cup \{(s,t) \quad \mid t \in T,\ s \in {}^\bullet t,\ s \equiv_D t\}\\
&\qquad \cup \{(s,t_s),(t_s,s_t),(s_t,t) \quad \mid t \in T,\ s \in {}^\bullet t,\ s \not\equiv_D t\}\ ,\\
\ell' \restriction T &= \ell \qquad \text{and} \qquad \ell'(t_s) = \tau \quad \text{for} \quad t_s \in T^\tau.
\end{aligned}$$

Proposition 1 For any (contact-free) net N, and any choice of $\equiv_D$, the net $I_D(N)$ is contact-free, and satisfies the other requirements imposed on nets, listed in Section 2.

Proof In Appendix A. □

The above protocol for replacing synchronous by asynchronous interaction appears to be one of the simplest ones imaginable. More intricate protocols, involving many asynchronous messages between a transition and its preplaces, could be contemplated, but we will not study them here. Our protocol involves just one such message, namely from the preplace to its posttransition. It is illustrated in Figure 1.

The last question above was: When does one net faithfully mimic the dynamic behaviour of another? This asks for a *semantic equivalence* on Petri nets, telling when two nets display the same behaviour. Many such equivalences have been studied in the literature. We believe that most of our results are independent of the precise choice of a semantic equivalence, as long as it preserves causality and branching time to some degree, and abstracts from silent transitions. Therefore we choose one such equivalence, based on its technical convenience in establishing our results, and postpone questions on the effect of varying this equivalence for further research. Our choice is *step readiness equivalence*, as defined in Section 2. Using this equivalence, we define a notion of *behavioural asynchrony* by asking whether the asynchronous implementation of a net preserves its behaviour. This notion is parametrised by the chosen interaction pattern, characterised as a requirement on the allowed distributions.

Definition 7 Let $\mathscr{Q}$ be a requirement on distributions of nets.
A plain net N is *behaviourally $\mathscr{Q}$-asynchronous* iff there exists a distribution D of N meeting the requirement $\mathscr{Q}$ such that $I_D(N) \approx_{\mathscr{R}} N$.

Intuitively, the only behavioural difference between a net N and its asynchronous implementation $I_D(N)$ can occur when in N a place $s \in {}^\bullet u$ is marked, whereas in $I_D(N)$ this token is already on its way from s to its posttransition u. In that case, it may occur that a transition $t \neq u$ with $s \in {}^\bullet t$ is enabled in N, whereas t is not enabled in the described state of $I_D(N)$. We call the situation in N leading to this state of $I_D(N)$ a *distributed conflict*; it is in fact the only circumstance in which $I_D(N)$ fails to faithfully mimic the dynamic behaviour of N.

Definition 8 Let $N = (S, T, F, M_0, \ell)$ be a net and D a distribution of N.
N has a *distributed conflict with respect to D* iff

$$\exists t, u \in T\ \exists p \in {}^\bullet t \cap {}^\bullet u.\ t \neq u \wedge p \not\equiv_D u \wedge \exists M \in [M_0\rangle_N.\ {}^\bullet t \subseteq M\ .$$

We wish to call a net N *(semi)structurally asynchronous* iff the situation outlined above never occurs, so that the asynchronous implementation does not change the behaviour of the net. As for behavioural asynchrony, this notion of asynchrony is parametrised by the set of allowed distributions.

Definition 9 Let $\mathscr{Q}$ be a requirement on distributions of nets.
A net N is *(semi)structurally $\mathscr{Q}$-asynchronous* iff there exists a distribution D of N meeting the requirement $\mathscr{Q}$ such that N has no distributed conflicts with respect to D.

The following theorem shows that distributed conflicts describe exactly the critical situations: For all plain nets the notions of structural and behavioural asynchrony coincide, regardless of the choice if $\mathscr{Q}$.

Theorem 1 Let N be a plain net, and $\mathscr{Q}$ a requirement on distributions of nets.
Then N is behaviourally $\mathscr{Q}$-asynchronous iff it is structurally $\mathscr{Q}$-asynchronous.

Proof In Appendix A. □

Because of this theorem, we call a plain net $\mathscr{Q}$-asynchronous if it is behaviourally and/or structurally $\mathscr{Q}$-asynchronous. In this paper we study this concept for plain nets only. When taking $\mathscr{Q} = \mathscr{Q}_{\text{FD}}$ we speak of *fully asynchronous nets*, when taking $\mathscr{Q} = \mathscr{Q}_{\text{SD}}$ of *symmetrically asynchronous nets*, and when taking $\mathscr{Q} = \mathscr{Q}_{\text{AD}}$ of *asymmetrically asynchronous nets*.

Example 1 The net N of Figure 1 is not fully asynchronous, for its unique D-based asynchronous implementation $I_D(N)$ with $D \in \mathscr{Q}_{\text{FD}}$ (also displayed in Figure 1) is not step readiness equivalent to N. In fact $\langle \varepsilon, \emptyset \rangle \in \mathscr{R}(I_D(N)) \setminus \mathscr{R}(N)$. This inequivalence arises because in $I_D(N)$ the option to do an a-action can be disabled already before any visible action takes place; this is not possible in N.

The only way to avoid a distributed conflict in this net is by taking $t \equiv_D p \equiv_D u$. This is not allowed for any $D \in \mathscr{Q}_{\text{FD}}$ or $D \in \mathscr{Q}_{\text{SD}}$, but it is allowed for $D \in \mathscr{Q}_{\text{AD}}$ (cf. the last net in Figure 1). Hence N is asymmetrically asynchronous, but not symmetrically asynchronous.

Since $\mathscr{Q}_{\text{FD}} \subseteq \mathscr{Q}_{\text{SD}} \subseteq \mathscr{Q}_{\text{AD}}$, any fully asynchronous net is symmetrically asynchronous, and any symmetrically asynchronous net is also asymmetrically asynchronous. Below we give semi-structural characterisations of these three classes of nets. The first two stem from [6], where the class of fully asynchronous nets is called *FA*(B) and the class of symmetrically asynchronous nets is called *SA*(B). The class *AA*(B) in [6] is somewhat larger than our class of asymmetrically asynchronous nets, for it is based on a slightly more involved protocol for replacing synchronous by asynchronous interaction.

Definition 10 A plain net $N = (S, T, F, M_0, \ell)$ has a

– *partially reachable conflict* iff

$$\exists t, u \in T\ \exists p \in {}^\bullet t \cap {}^\bullet u.\ t \neq u \wedge \exists M \in [M_0\rangle_N.\ {}^\bullet t \subseteq M\ ,$$

– *partially reachable* N iff

$$\exists t, u \in T\ \exists p \in {}^\bullet t \cap {}^\bullet u.\ t \neq u \wedge |{}^\bullet u| > 1 \wedge \exists M \in [M_0\rangle_N.\ {}^\bullet t \subseteq M\ ,$$

– *left and right border reachable* M iff

$$\exists t, u, v \in T\ \exists p \in {}^\bullet t \cap {}^\bullet u\ \exists q \in {}^\bullet u \cap {}^\bullet v.\ \begin{array}{l} t \neq u \wedge u \neq v \wedge p \neq q \wedge \\ \exists M_1, M_2 \in [M_0\rangle_N.\ {}^\bullet t \subseteq M_1 \wedge {}^\bullet v \subseteq M_2\ . \end{array}$$

Theorem 2 Let N be a plain net.

– N is fully asynchronous iff it has no partially reachable conflict.
– N is symmetrically asynchronous iff it has no partially reachable N.
– N is asymmetrically asynchronous iff it has no left and right border reachable M.

Proof Straightforward with Theorem 1. □

In the theory of Petri nets, there have been extensive studies on classes of nets with certain structural properties like *free choice nets* [3, 2] and *simple nets* [3], as well as extensions of theses classes. They are closely related to the net classes defined here, but they are defined without taking reachability into account. For a comprehensive overview and discussion of the relations between those purely structurally defined net classes and our net classes see [6]. Restricted to plain nets without dead transitions (meaning that every transition t satisfies the requirement $\exists M \in [M_0\rangle.\ {}^\bullet t \subseteq M$), Theorem 2 says that a net is fully synchronous iff it is conflict-free in the structural sense (no shared preplaces), symmetrically asynchronous iff it is a free choice net and asymmetrically asynchronous iff it is simple.

Our asynchronous net classes are defined for plain nets only. There are two approaches to lifting them to labelled nets. One is to postulate that whether a net is asynchronous or not has nothing to do with its labelling function, so that after replacing this labelling by the identity function one can apply the insights above. This way our structural characterisations (Theorems 1 and 2) apply to labelled nets as well. Another approach would be to apply the notion of behavioural asynchrony of Definition 7 directly to labelled nets. This way more nets will be asynchronous, because in some cases a net happens to be equivalent to its asynchronous implementation in spite of a failure of structural asynchrony. This happens for instance if all transitions in the original net are labelled τ. Unlike the situation for plain nets, the resulting notion of behavioural asynchrony will most likely be strongly dependent on the choice of the semantic equivalence relation between nets.

4 Distributed Systems

The approach of Section 3 makes a difference between a net regarded as a specification, and an asynchronous implementation of the same net. The latter could be thought of as a way to execute the net when a given distribution makes the synchronisations that are inherent in the specification impossible. In this and the following section, on the other hand, we drop the difference between a net and its asynchronous implementation. Instead of adapting our intuition about the firing rule when implementing a net in a distributed way, we insist that all synchronisations specified in the original net remain present as synchronisations in a distributed implementation. Yet, at the same time we stick to the point of view that it is simply not possible for a transition to synchronise its firing with the removal of tokens from preplaces at remote locations. Thus we only allow distributions in which each transition is co-located with all of its preplaces. We call such distributions *effectual*. For effectual distributions D, the implementation transformation I_D is the identity. As a consequence, if effectuality is part of a requirement $\mathscr{Q}$ imposed on distributions, the question whether a net is $\mathscr{Q}$-asynchronous is no longer dependent on whether an asynchronous implementation mimics the behaviour of the given net, but rather on whether the net allows a distribution satisfying $\mathscr{Q}$ at all.

The requirement of effectuality does not combine well will the requirements on distributions proposed in Definition 5. For if $\mathscr{Q}$ is the class of distributions that are effectual and asymmetrically distributed, then only nets without transitions with multiple preplaces would be $\mathscr{Q}$-asynchronous. This rules out most useful applications of Petri nets. The requirement of effectuality by itself, on the other hand, would make every net asynchronous, because we could assign the same location to all places and transitions.

We impose one more fundamental restriction on distributions, namely that when two visible transitions can occur in one step, they cannot be co-located. This is based on the assumption that at a given location visible actions can only occur sequentially, whereas we want to preserve as much concurrency as possible (in order not to loose performance). Recall that in Petri nets simultaneity of transitions cannot be enforced: if two transitions can fire in one step, they can also fire in any order. The standard interpretation of nets postulates that in such a case those transitions are causally independent, and this idea fits well with the idea that they reside at different locations.

Definition 11 Let $N = (S, T, F, M_0, \ell)$ be a net.
The *concurrency relation* $\smile \subseteq T^2$ is given by $t \smile u \Leftrightarrow t \neq u \wedge \exists M \in [M_0\rangle.\ M[\{t, u\}\rangle$.
N is *distributed* iff it has a distribution D such that
- $\forall s \in S,\ t \in T.\ s \in {}^\bullet t \Rightarrow t \equiv_D s$,
- $t \smile u \wedge l(t), l(u) \neq \tau \Rightarrow t \not\equiv_D u$.

It is straightforward to give a semi-structural characterisation of this class of nets:

Observation 1 A net is distributed iff there is no sequence $t_0, \ldots, t_n$ of transitions with $t_0 \smile t_n$ and ${}^\bullet t_{i-1} \cap {}^\bullet t_i \neq \emptyset$ for $i = 1, \ldots, n$.

A structure as in the above characterisation of distributed nets can be considered as a prolonged M containing two independent transitions that can be simultaneously enabled.

It is not hard to find a plain net that is fully asynchronous, yet not distributed. However, restricted to plain nets without dead transitions, the class of asymmetrically asynchronous nets is a strict subclass of the class of distributed nets. Namely, if a net is M-free (where an M is as in Definition 10, but without the reachability condition on the bottom line), then it surely has no sequence as described above.

5 Distributable Systems

In this section, we will investigate the borderline for distributability of systems. It is a well known fact that sometimes a global protocol is necessary when concurrent activities in a system interfere. In particular, this may be necessary for deciding choices in a coherent way. Consider for example the

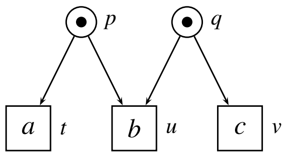

Figure 2: A fully marked M.

simple net in Figure 2. It contains an M-structure, which was already exhibited as a problematic one in Section 3. Transitions t and v are supposed to be concurrently executable (if we do not want to restrict performance of the system), and hence reside on different locations. Thus at least one of them, say t, cannot be co-located with transition u. However, both transitions are in conflict with u.

As we use nets as models of reactive systems, we allow the environment of a net to influence decisions at runtime by blocking one of the possibilities. Equivalently we can say it is the environment that fires transitions, and this can only happen for transitions that are currently enabled in the net. If the net decides between t and u before the actual execution of the chosen transition, the environment might change its mind in between, leading to a state of deadlock. Therefore we work in a branching time semantics, in which the option to perform t stays open until either t or u occurs. Hence the decision to fire u can only be taken at the location of u, namely by firing u, and similarly for t. Assuming that it takes time to propagate any message from one location to another, in no distributed implementation of this net can t and u be simultaneously enabled, because in that case we cannot exclude that both of them happen. Thus, the only possible implementation of the choice between t and u is to alternate the right to fire between t and u, by sending messages between them (cf. Figure 3). But if the environment only sporadically tries to fire t or u it may repeatedly miss the opportunity to do so, leading to an infinite loop of control messages sent back and forth, without either transition ever firing.

In this section we will formalise this reasoning, and show that under a few mild assumptions this type of structures cannot be implemented in a distributed manner at all, i.e. even when we allow the implementation to be completely unrelated to the specification, except for its behaviour. For this, we apply the notion of a distributed net, as introduced in the previous section. Furthermore, we need an equivalence notion in order to specify in which way an implementation as a distributed net is required to preserve the behaviour of the original net. As in Section 3, we choose step readiness equivalence. We call a plain net *distributable* if it is step readiness equivalent to a distributed net. We speak of a *truly synchronous* net if it is not distributable, thus if it may not be transformed into any distributed net with the same behaviour up to step readiness equivalence, that is if no such net exists. We study the concept "distributable" for plain nets only, but in order to get the largest class possible we allow non-plain implementations, where a given transition may be split into multiple transitions carrying the same label.

Definition 12 A plain net N is *truly synchronous* iff there exists no distributed net N' which is step readiness equivalent to N.

We will show that nets like the one of Figure 2 are truly synchronous.

Step readiness equivalence is one of the simplest and least discriminating equivalences imaginable that preserves branching time, causality and divergence to some small extend. Our impossibility result, formalised below as Theorem 3, depends crucially on all three properties, and thus needs to be reconsidered when giving up on any of them. When working in linear time semantics, every net is equivalent to an infinite net that starts with a choice between several τ-transitions, each followed by a conflict-free net

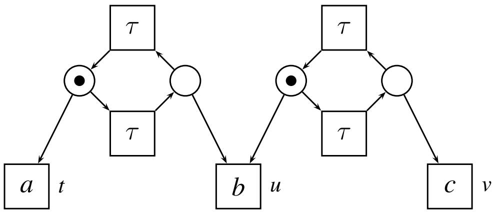

Figure 3: A busy-wait implementation of the net in Figure 2

modelling a single run. This net is N-free, and hence distributed. It can be argued that infinite implementations are not acceptable, but when searching for the theoretical limits to distributed implementability we don't want to rule them out dogmatically. When working in interleaving semantics, any net can be converted into an equivalent distributed net by removing all concurrency between transitions. This can be accomplished by adding a new, initially marked place, with an arc to and from every transition in the net. When fully abstracting from divergence, even when respecting causality and branching time, the net of Figure 2 is equivalent to the distributed net of Figure 3, and in fact it is not hard to see that this type of implementation is possibly for any given net. Yet, the implementation is suspect, as the implemented decision of a choice may fail to terminate. The clause $M \stackrel{\tau}{\not\longrightarrow}$ in Definition 4 is strong enough to rule out this type of implementation, even though our step readiness semantics abstracts from other forms of divergence.

We now characterise the class of nets which we will prove to be truly synchronous.

Definition 13 Let $N = (S, T, F, M_0, \ell)$ be a net.
N has a *fully reachable visible pure* M iff $\exists t, u, v \in T.\ ^\bullet t \cap {}^\bullet u \neq \emptyset \wedge {}^\bullet u \cap {}^\bullet v \neq \emptyset \wedge {}^\bullet t \cap {}^\bullet v = \emptyset \wedge \ell(t), \ell(u), \ell(v) \neq \tau \wedge \exists M \in [M_0\rangle.\ ^\bullet t \cup {}^\bullet u \cup {}^\bullet v \subseteq M$.

Here a *pure* M is an M as in Definition 10 that moreover satisfies ${}^\bullet t \cap {}^\bullet v = \emptyset$, and hence $p \notin {}^\bullet v$, $q \notin {}^\bullet t$ and $t \neq v$. These requirements follow from the conditions above.

Proposition 2 A net with a fully reachable visible pure M is not distributed.

Proof Let $N = (S, T, F, M_0, \ell)$ be a net that has a fully reachable visible pure M, so there exist $t, u, v \in T$ and $p, q \in S$ such that $p \in {}^\bullet t \cap {}^\bullet u \wedge q \in {}^\bullet u \cap {}^\bullet v \wedge {}^\bullet t \cap {}^\bullet v = \emptyset$ and $\exists M \in [M_0\rangle.\ ^\bullet t \cup {}^\bullet u \cup {}^\bullet v \subseteq M$. Then $t \smile v$. Suppose N is distributed by the distribution D. Then $t \equiv_D p \equiv_D u \equiv_D q \equiv_D v$ but $t \smile v$ implies $t \not\equiv_D v$. ↯ □

Now we show that fully reachable visible pure M's that are present in a plain net are preserved under step readiness equivalence.

Lemma 1 Let $N = (S, T, F, M_0, \ell)$ be a plain net.
If N has a fully reachable visible pure M, there exists $<\sigma, X> \in \mathscr{R}(N)$ such that $\exists a, b, c \in$ Act. $a \neq c \wedge \{b\} \in X \wedge \{a, c\} \in X \wedge \{a, b\} \notin X \wedge \{b, c\} \notin X$. (It is implied that $a \neq b \neq c$.)

Proof N has a fully reachable visible pure M, so there are $t, u, v \in T$ and $M \in [M_0\rangle$ such that ${}^\bullet t \cap {}^\bullet u \neq \emptyset$ $\wedge {}^\bullet u \cap {}^\bullet v \neq \emptyset \wedge {}^\bullet t \cap {}^\bullet v = \emptyset \wedge \ell(t), \ell(u), \ell(v) \neq \tau \wedge {}^\bullet t \cup {}^\bullet u \cup {}^\bullet v \subseteq M$. Let $\sigma \in \text{Act}^*$ such that $M_0 \stackrel{\sigma}{\Longrightarrow} M$. Since N is a plain net, $M \stackrel{\tau}{\not\longrightarrow}$ and $\ell(t) \neq \ell(u) \neq \ell(v) \neq \ell(t)$. Hence there exists an $X \subseteq \mathbb{N}^{\text{Act}}$ such that $<\sigma, X> \in \mathscr{R}(N) \wedge \{\ell(u)\} \in X \wedge \{\ell(t), \ell(v)\} \in X \wedge \{\ell(t), \ell(u)\} \notin X \wedge \{\ell(u), \ell(v)\} \notin X$. □

Lemma 2 Let $N = (S, T, F, M_0, \ell)$ be a net.
If there exists $<\sigma, X> \in \mathscr{R}(N)$ such that $\exists a, b, c \in \text{Act}.\ a \neq c \wedge \{b\} \in X \wedge \{a, c\} \in X \wedge \{a, b\} \notin X \wedge \{b, c\} \notin X$, then N has a fully reachable visible pure M.

Proof Let $M \subseteq S$ be the marking which gave rise to the step ready pair $<\sigma, X>$, i.e. $M_0 \overset{\sigma}{\Longrightarrow} M$ and $M \xrightarrow{\{b\}} \wedge M \xrightarrow{\{a,c\}} \wedge M \overset{\{a,b\}}{\not\longrightarrow} \wedge M \overset{\{b,c\}}{\not\longrightarrow}$.

As $a \neq b \neq c \neq a$ there must exist three transitions $t, u, v \in T$ with $\ell(t) = a \wedge \ell(u) = b \wedge \ell(v) = c$ and $M[\{u\}\rangle \wedge M[\{t, v\}\rangle \wedge \neg(M[\{t, u\}\rangle) \wedge \neg(M[\{u, v\}\rangle)$. From $M[\{u\}\rangle \wedge M[\{t, v\}\rangle$ follows ${}^\bullet t \cup {}^\bullet u \cup {}^\bullet v \subseteq M$. From $M[\{t, v\}\rangle$ follows ${}^\bullet t \cap {}^\bullet v = \emptyset$. From $\neg(M[\{t, u\}\rangle)$ then follows ${}^\bullet t \cap {}^\bullet u \neq \emptyset$ and analogously for u and v. Hence N has a fully reachable visible pure M. □

Note that the lemmas above give a behavioural property that for plain nets is equivalent to having a fully reachable visible pure M.

Theorem 3 A plain net with a fully reachable visible pure M is truly synchronous.

Proof Let N be a plain net which has a fully reachable visible pure M. Let N' be a net which is step readiness equivalent to N. By Lemma 1 and Lemma 2, also N' has a fully reachable visible pure M. By Proposition 2, N' is not distributed. Thus N is truly synchronous. □

Theorem 3 gives an upper bound of the class of distributable nets. We conjecture that this upper bound is tight, and a plain net is distributable iff it has no fully reachable visible pure M.

Conjecture 1 A plain net is truly synchronous iff it has a fully reachable visible pure M.

In the following, we give a lower bound of distributability by providing a protocol to implement certain kinds of plain nets distributedly. These implementations do not add additional labelled transitions, but only provide the existing ones with a communication protocol in the form of τ-transitions. Hence these implementations pertain to a notion of distributability in which we restrict implementations to be plain τ-nets. Note that this does not apply to the impossibility result above.

Definition 14 A plain net N is *plain-distributable* iff there exists a distributed plain τ-net N which is step readiness equivalent to N.

Definition 15 Let $N = (S, T, F, M_0, \ell)$ be a net.
We define the *enabled conflict relation* $\# \subseteq T^2$ as

$$t \# u \Leftrightarrow \exists M \in [M_0\rangle.\ M[\{t\}\rangle \wedge M[\{u\}\rangle \wedge \neg(M[\{t, u\}\rangle).$$

We now propose the following protocol for implementing nets. An example depicting it can be found in Figure 5. As locations we take the places in a given net, and the equivalence classes of transitions that are related by the reflexive and transitive closure of the enabled conflict relation. We locate every transition t in its equivalence class, whereas every place gets a private location. Every place s will have an embassy $s^{[t]}$ in every location $[t]$ where one of its posttransitions $t \in s^\bullet$ resides. As soon as s receives a token, it will distribute this information to its posttransitions by placing a token in each of these embassies. The arc from s to t is now replaced by an arc from $s^{[t]}$ to t, so if t could fire in the original net it can also fire in the implementation. So far the construction allows two transitions in different locations that shared the precondition s to fire concurrently, although they were in conflict in the original net. However, if this

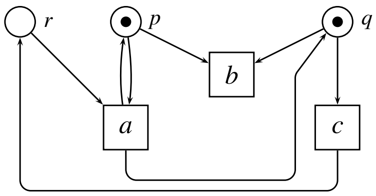

Figure 4: An example net

situation actually occurs, these transitions would have been in an enabled conflict, and thus assigned to the same location. The rest of the construction is a matter of garbage collection. If a transition t fires, for each of its preplaces s, all tokens that are still present in the various embassies of s in locations $[u]$ need to be removed from there. This is done by a special internal transition $t_s^{[u]}$. Once all these transitions (for the various choices of s and $[u]$) have fired, an internal transition t' occurs, which puts tokens in all the postplaces of t.

Definition 16 Let $N = (S, T, F, M_0, \ell)$ be a net.
Let $[t] := \{u \in T \mid t \#^* u\}$. The transition-controlled-choice implementation of N is defined to be the net $N' := (S \cup S^\tau, T \cup T^\tau, F', M_0, \ell')$ with

$$\begin{aligned}
S^\tau &:= \{s^{[t]} \mid s \in S, t \in s^\bullet\} \cup \{\textcircled{t} \mid t \in T\} \cup \\
&\quad\ \{s_t^{[u]}, \overline{s}_t^{[u]} \mid s \in S,\ t, u \in s^\bullet, [u] \neq [t]\} \\
T^\tau &:= \{\boxed{s} \mid s \in S\} \cup \{t' \mid t \in T\} \cup \\
&\quad\ \{t_s^{[u]} \mid s \in S,\ t, u \in s^\bullet, [u] \neq [t]\} \\
F' &:= \{(s, \boxed{s}) \mid s \in S\} \cup \\
&\quad\ \{(\boxed{s}, s^{[t]}), (s^{[t]}, t) \mid s \in S, t \in s^\bullet\} \cup \\
&\quad\ \{(t, \textcircled{t}), (\textcircled{t}, t') \mid t \in T\} \cup \\
&\quad\ \{(t', s) \mid t \in T, s \in t^\bullet\} \cup \\
&\quad\ \{(t, s_t^{[u]}), (s_t^{[u]}, t_s^{[u]}), (t_s^{[u]}, \overline{s}_t^{[u]}), (\overline{s}_t^{[u]}, t'), (s^{[u]}, t_s^{[u]}) \mid s \in S,\ t, u \in s^\bullet,\ [u] \neq [t]\}
\end{aligned}$$

$\ell' \restriction T = \ell$ and $\ell'(T^\tau) = \{\tau\}$.

Theorem 4 A plain net N is plain-distributable iff $\#^* \cap \smile = \emptyset$.

Proof "$\Rightarrow$": When implementing a plain net N by a plain τ-net N' that is step readiness equivalent to N, the $\#$ and $\smile$ relations between the transitions of N also exists between the corresponding visible transitions of N'. This is easiest to see when writing a_N, resp. $a_{N'}$, to denote a transition in N, resp. N', with label a, which must be unique since N is a plain net, resp. N' a plain τ-net. Namely if $a_N \# b_N$, then N has a step ready pair $<\sigma, X>$ with $\{a\}$,$\{b\} \in X$ but $\{a, b\} \notin X$. This must also be a step ready pair of N', and hence $a_{N'} \# b_{N'}$. Likewise, $a_N \smile b_N$ implies $a_{N'} \smile b_{N'}$.

Thus if $\#^* \cap \smile \neq \emptyset$ holds in N, then the same is the case for N', and hence N' is not distributed by Observation 1.

"$\Leftarrow$": If $\#^* \cap \smile = \emptyset$, N can be implemented as specified in Definition 16. In fact, the transition-controlled-choice implementation of any net N yields a net that is step readiness equivalent to N. See Appendix B for a formal proof of this claim. By construction, if N is plain, its transition-controlled-choice implementation is a plain τ-net. Moreover, if $\#^* \cap \smile = \emptyset$ it never happens that concurrent visible transitions are co-located, and hence the implementation will be distributed. $\square$

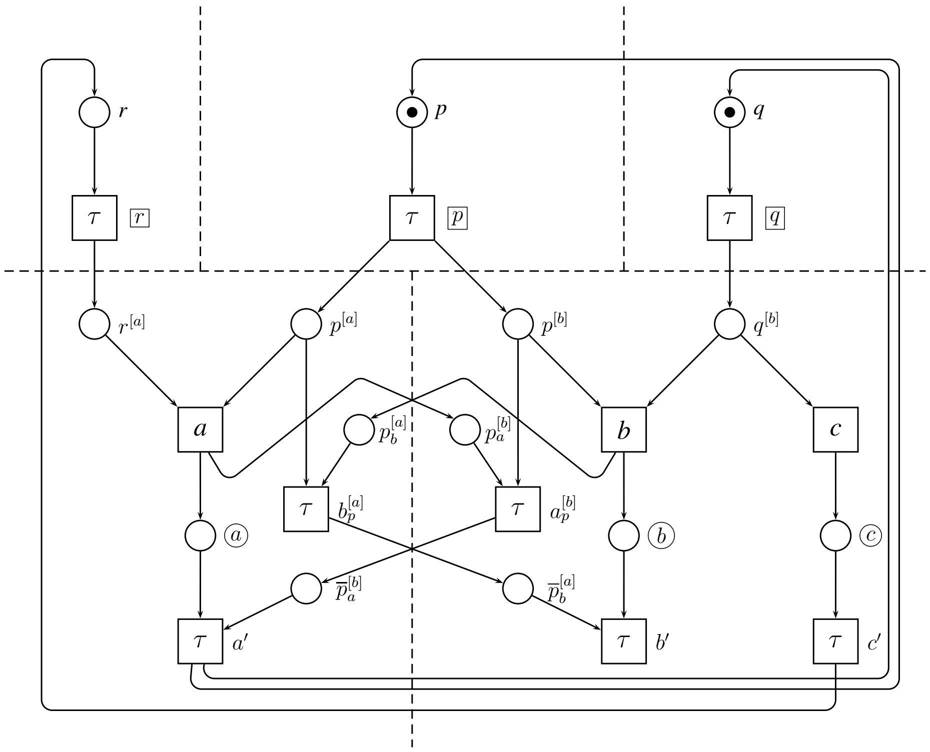

Figure 5: A distributed implementation for the net in Figure 4, partitioning into localities shown by dashed lines

Our definition of distributed nets only enforces concurrent actions to be on different locations if they are visible, and our implementation in Definition 16 produces nets which actually contain concurrent unobservable activity at the same location. If this is undesired it can easily be amended by adding a single marked place to every location and connecting that place to every transition on that location by a self-loop. While this approach will introduce new causality relations, step readiness equivalence will not detect this.

6 Conclusion

In this paper, we have characterised different grades of asynchrony in Petri nets in terms of structural and behavioural properties of nets. Moreover, we have given both an upper and a lower bound of distributability of behaviours. In particular we have shown that some branching-time behaviours cannot be exhibited by a distributed system.

We did not consider connections from transitions to their postplaces as relevant to determine asynchrony and distributability. This is because we only discussed contact-free nets where no synchronisation by postplaces is necessary. In the spirit of Definition 6 we could insert τ-transitions on any or all arcs from transitions to their postplaces, and the resulting net would always be equivalent to the original.

We have already given a short overview on related work in the introduction of this paper. Most closely related to our approach are several lines of work using Petri nets as a model of reactive systems.

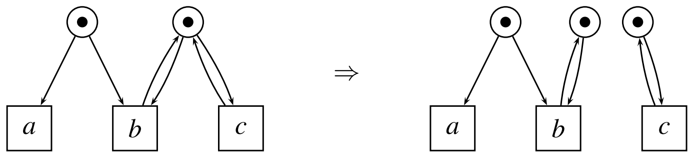

Figure 6: A specification and its Hopkins-implementation which added concurrency.

As mentioned in Section 3, classes of nets with certain structural properties like *free choice nets* [3, 2] and *simple nets* [3], as well as extensions of theses classes, have been extensively studied in Petri net theory, and are closely related to the classes of nets defined here. In [3], Eike Best and Mike Shields introduce various transformations between free choice nets, simple nets and extended variants thereof. They use "essential equivalence" to compare the behaviour of different nets, which they only give informally. This equivalence is insensitive to divergence, which is relied upon in their transformations. It also does not preserve concurrency, which makes it possible to implement *behavioural free choice nets*, that may feature a fully reachable visible M, as free choice nets. They continue to show conditions under which liveness can be guaranteed for many of these classes.

In [1], Wil van der Aalst, Ekkart Kindler and Jörg Desel introduce two extensions to extended simple nets, by excluding self-loops from the requirements imposed on extended simple nets. This however assumes a kind of "atomicity" of self-loops, which we did not allow in this paper. In particular we do not implicitly assume that a transition will not change the state of a place it is connected to by a self-loop, since in case of deadlock, the temporary removal of a token from such a place might not be temporary indeed.

In [16], Wolfgang Reisig introduces a class of systems which communicate using buffers and where the relative speeds of different components are guaranteed to be irrelevant. The resulting nets are simple nets. He then proceeds introducing a decision procedure for the problem whether a marking exists which makes the complete system live.

Dirk Taubner has in [18] given various protocols by which to implement arbitrary Petri nets in the OCCAM programming language. Although this programming language offers synchronous communication he makes no substantial use of that feature in the protocols, thereby effectively providing an asynchronous implementation of Petri nets. He does not indicate a specific equivalence relation, but is effectively using linear-time equivalences to compare implementations to the specification.

The work most similar to our approach we have found is the one by Hopkins, [9]. There he already classified nets by whether they are implementable by a net distributed among different locations. He uses an interleaving equivalence to compare an implementation to the original net, and while allowing a range of implementations, he does require them to inherit some of the structure of the original net. The net classes he describes in his paper are larger than those of Section 3 because he allows more general interaction patterns, but they are incomparable with those of Section 5. One direction of this inequality depends on his choice of interleaving semantics, which allows the implementation in Figure 6. The step readiness equivalence we use does not tolerate the added concurrency and the depicted net is not distributable in our sense. The other direction of the inequality stems from the fact that we allow implementations which do not share structure with the specification but only emulate its behaviour. That way, the net in Figure 7 can be implemented in our approach as depicted.

Still many open questions remain. While our impossibility result holds even when allowing labelled nets as implementations, our characterisation in Theorem 4 only considers unlabelled ones. This begs the

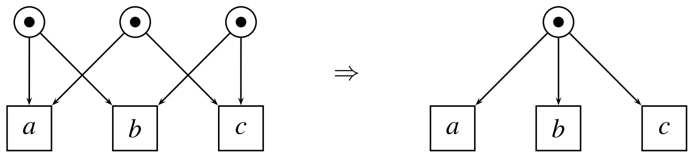

Figure 7: A distributable net which is not considered distributable in [9], and its implementation.

question which class of nets can be implemented using labelled nets. We conjecture that a distributed implementation exists for every net which has no fully reachable visible pure M. We also conjecture that if we allow linear time correct implementations, all nets become distributable, even when only allowing finite implementations of finite nets. We are currently working on both problems.

Just as a distributable net is defined as a net that is behaviourally equivalent to, or implementable by, a distributed net, one could define an *asynchronously implementable* net as one that is implementable by an asynchronous net. This concept is again parametrised by the choice of an interaction pattern. It would be an interesting quest to characterise the various classes of asynchronously implementable plain nets.

Also, extending our work to nets that are not required to be 1-safe will probably generate interesting results, as conflict resolution protocols must keep track of which token they are currently resolving the conflict of.

In regard to practical applicability of our results, it would be very interesting to relate our Petri net based terminology to hardware descriptions in chip design. Especially in modern multi-core architectures performance reasons often prohibit using global clocks while a façade of synchrony must still be upheld in the abstract view of the system.

On a higher level of applications, we expect our results to be useful for language design. To start off, we would like to make a thorough comparison of our results to those on communication patterns in process algebras, versions of the π-calculus and I/O-automata [12]. Using a Petri net semantics of a suitable system description language, we could compare our net classes to the class of nets expressible in the language, especially when restricting the allowed communication patterns in the various ways considered in [4] or in [12]. Furthermore, we are interested in applying our results to graphical formalisms for system design like UML sequence diagrams or activity diagrams, also by applying their Petri net semantics. Our results become relevant when such formalisms are used for the design of distributed systems. Certain choice constructs become problematic then, as they rely on a global mechanism for consistent choice resolution; this could be made explicit in our framework.

References

[1] W.M.P. van der Aalst, E. Kindler & J. Desel (1998): *Beyond asymmetric choice: A note on some extensions.* *Petri Net Newsletter* 55, pp. 3–13.

[2] E. Best (1987): *Structure theory of Petri nets: The free choice hiatus.* In W. Brauer, W. Reisig & G. Rozenberg, editors: *Advances in Petri Nets 1986*, LNCS 254, Springer, pp. 168–206.

[3] E. Best & M.W. Shields (1983): *Some equivalence results for free choice nets and simple nets and on the periodicity of live free choice nets.* In G. Ausiello & M. Protasi, editors: Proceedings 8th Colloquium on *Trees in Algebra and Programming (CAAP '83)*, LNCS 159, Springer, pp. 141–154.

[4] F.S. de Boer & C. Palamidessi (1991): *Embedding as a tool for language comparison: On the CSP hierarchy.* In J.C.M. Baeten & J.F. Groote, editors: Proceedings 2nd International Conference on *Concurrency Theory* (CONCUR'91), Amsterdam, The Netherlands, LNCS 527, Springer, pp. 127–141.

[5] L. Bougé (1988): *On the existence of symmetric algorithms to find leaders in networks of communicating sequential processes.* *Acta Informatica* 25(2), pp. 179–201.

[6] R.J. van Glabbeek, U. Goltz & J.-W. Schicke (2008): *Symmetric and asymmetric asynchronous interaction.* Technical Report 2008-03, TU Braunschweig. Extended abstract in Proceedings 1st *Interaction and Concurrency Experience* (ICE'08) on *Synchronous and Asynchronous Interactions in Concurrent Distributed Systems*, to appear in *Electronic Notes in Theoretical Computer Science*, Elsevier.

[7] R.J. van Glabbeek & W.P. Weijland (1996): *Branching time and abstraction in bisimulation semantics.* *Journal of the ACM* 43(3), pp. 555–600.

[8] D. Gorla (2006): *On the relative expressive power of asynchronous communication primitives.* In L. Aceto & A. Ingólfsdóttir, editors: *Proceedings 9th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '06)*, LNCS 3921, Springer, pp. 47–62.

[9] R.P. Hopkins (1991): *Distributable nets.* In *Advances in Petri Nets 1991*, LNCS 524, Springer, pp. 161–187.

[10] L. Lamport (1978): *Time, clocks, and the ordering of events in a distributed system.* *Communications of the ACM* 21(7), pp. 558–565.

[11] L. Lamport (2003): *Arbitration-free synchronization.* *Distributed Computing* 16(2-3), pp. 219–237.

[12] N. Lynch (1996): *Distributed Algorithms*. Morgan Kaufmann Publishers.

[13] U. Nestmann (2000): *What is a 'good' encoding of guarded choice?* *Information and Computation* 156, pp. 287–319.

[14] E.-R. Olderog & C.A.R. Hoare (1986): *Specification-oriented semantics for communicating processes.* *Acta Informatica* 23, pp. 9–66.

[15] C. Palamidessi (1997): *Comparing the expressive power of the synchronous and the asynchronous pi-calculus.* In Conference Record of the 24th ACM SIGPLAN-SIGACT Symposium on *Principles of Programming Languages (POPL '97)*, ACM Press, pp. 256–265.

[16] W. Reisig (1982): *Deterministic buffer synchronization of sequential processes.* *Acta Informatica* 18, pp. 115–134.

[17] Peter Selinger (1997): *First-order axioms for asynchrony.* In Proceedings 8th International Conference on *Concurrency Theory* (CONCUR'97), Warsaw, Poland, *LNCS* 1243, Springer, pp. 376–390.

[18] Dirk Taubner (1988): *Zur verteilten Implementierung von Petrinetzen.* *Informationstechnik* 30(5), pp. 357–370. Technical report, TUM-I 8805, TU München.

A The Asynchronous Implementation

Given a net N and a distribution D on N, this appendix explores the properties of the D-based asynchronous implementation $I_D(N)$ of N, focussing on the relationship between $I_D(N)$ and N, and culminating in the proofs of Proposition 1 and Theorem 1 of Section 3.

For better readability we will use the abbreviations ${}^\circ x := \{y \mid (y,x) \in F'\}$ and $x^\circ := \{y \mid (x,y) \in F'\}$ instead of ${}^\bullet x$ or $x^\bullet$ when making assertions about the flow relation of an implementation.

The following lemma shows how the D-based asynchronous implementation of a net N simulates the behaviour of N.

Lemma 3 Let $N = (S,T,F,M_0,\ell)$ be a net, $A \subseteq \text{Act}$, $\sigma \in \text{Act}^*$ and $M_1, M_2 \subseteq S$.

1. If $M_1 \xrightarrow{A}_N M_2$ then $M_1 \xrightarrow{\tau}{}^*_{I_D(N)} \xrightarrow{A}_{I_D(N)} M_2$.
2. If $M_1 \overset{\sigma}{\Longrightarrow}_N M_2$ then $M_1 \overset{\sigma}{\Longrightarrow}_{I_D(N)} M_2$.

Proof Assume $M_1\,[G\rangle_N\, M_2$. Then, by construction of $I_D(N)$,

$$M_1\ [\{t_s \mid t \in G,\ s \in {}^\bullet t,\ s \not\equiv_D t\}\rangle_{I_D(N)}\ [\{t \mid t \in G\}\rangle_{I_D(N)}\ M_2.$$

The first part of that execution can be split into a sequence of singleton transitions, all labelled τ.
The second statement follows by a straightforward induction on the length of σ. □

This lemma uses the fact that any marking of N is also a marking on $I_D(N)$. The reverse does not hold, so in order to describe the degree to which the behaviour of $I_D(N)$ is simulated by N we need to explicitly relate markings of $I_D(N)$ to those of N. This is in fact not so hard, as any reachable marking of $I_D(N)$ can be obtained from a reachable marking of N by moving some tokens into the newly introduced buffering places s_t. To establish this formally, we define a function which transforms implementation markings into the related original markings, by shifting these tokens back.

Definition 17 Let $N = (S,T,F,M_0,\ell)$ be a net and let $I_D(N) = (S \cup S^\tau, T \cup T^\tau, F', M_0, \ell')$.
$\tau^\leftarrow : S \cup S^\tau \to S$ is the function defined by

$$\tau^\leftarrow(p) := \begin{cases} s & \text{iff } p = s_t \text{ with } s_t \in S^\tau,\ s \in S,\ t \in T \\ p & \text{otherwise } (p \in S) \end{cases}$$

Where necessary we extend functions to sets elementwise. So for any $M \subseteq S \cup S^\tau$ we have $\tau^\leftarrow(M) = \{\tau^\leftarrow(s) \mid s \in M\} = (M \cap S) \cup \{s \mid s_t \in M\}$. In particular, $\tau^\leftarrow(M) = M$ when $M \subseteq S$.

We now introduce a predicate α on the markings of $I_D(N)$ that holds for a marking iff it can be obtained from a reachable marking of N (which is also a marking of $I_D(N)$) by firing some unobservable transitions. Each of these unobservable transitions moves a token from a place s into a buffering place s_t. Later, we will show that α exactly characterises the reachable markings of $I_D(N)$. Furthermore, as every token can be moved only once, we can also give an upper bound on how many such movements can still take place.

Definition 18 Let $N = (S, T, F, M_0, \ell)$ be a net and $I_D(N) = (S \cup S^\tau, T \cup T^\tau, F', M_0, \ell')$.
The predicate $\alpha \subseteq \mathcal{P}(S \cup S^\tau)$ is given by

$$\alpha(M) \quad :\Leftrightarrow \quad \tau^{\leftarrow}(M) \in [M_0\rangle_N \wedge \forall p, q \in M.\, \tau^{\leftarrow}(p) = \tau^{\leftarrow}(q) \Rightarrow p = q.$$

The function $d : \mathcal{P}(S \cup S^\tau) \to \mathbb{N} \cup \{\infty\}$ is given by $d(M) := |M \cap \{s \mid s \in S,\ \exists t \in s^\bullet.\ s \not\equiv_D t\}|$, where we choose not to distinguish between different degrees of infinity.

Note that $\alpha(M)$ implies $|M| = |\tau^{\leftarrow}(M)|$, and reachable markings of N are always finite (thanks to our definition of a net). Hence $\alpha(M)$ implies $d(M) \in \mathbb{N}$. The following lemma confirms that our informal description of α matches its formal definition.

Lemma 4 Let N and $I_D(N)$ be as above and $M \subseteq S \cup S^\tau$, with M finite.
Then $\forall p, q \in M.\, \tau^{\leftarrow}(p) = \tau^{\leftarrow}(q) \Rightarrow p = q$ iff $\tau^{\leftarrow}(M) \xrightarrow{\tau}{}^{*}_{I_D(N)} M$.

Proof Given that $\tau^{\leftarrow}(M) \subseteq S$, "if" follows directly from the construction of $I_D(N)$.
For "only if", assume $\forall p, q \in M.\, \tau^{\leftarrow}(p) = \tau^{\leftarrow}(q) \Rightarrow p = q$. Then $\tau^{\leftarrow}(M)\ [\{t_s \mid s_t \in M\}\rangle_{I_D(N)}\ M$. $\square$

Now we can describe how any net simulates the behaviour of its fully asynchronous implementation.

Lemma 5 Let N and $I_D(N)$ be as above, $A \subseteq \mathrm{Act}$, $\sigma \in \mathrm{Act}^*$ and $M, M' \subseteq S \cup S^\tau$.

1. $\alpha(M_0)$.
2. If $\alpha(M) \wedge M \xrightarrow{A}_{I_D(N)} M'$ then $\tau^{\leftarrow}(M) \xrightarrow{A}_N \tau^{\leftarrow}(M') \wedge \alpha(M')$.
3. If $\alpha(M) \wedge M \xrightarrow{\tau}_{I_D(N)} M'$ then $d(M) > d(M') \wedge \tau^{\leftarrow}(M) = \tau^{\leftarrow}(M') \wedge \alpha(M')$.
4. If $M_0 \overset{\sigma}{\Longrightarrow}_{I_D(N)} M'$ then $M_0 \overset{\sigma}{\Longrightarrow}_N \tau^{\leftarrow}(M') \wedge \alpha(M')$.

Proof "1": $M_0 \in [M_0\rangle_N$ and $\forall s \in M_0 \subseteq S.\, \tau^{\leftarrow}(s) = s$.

"2": Suppose $\alpha(M)$ and $M\ [G\rangle_{I_D(N)}\ M'$ with $G \subseteq T$. So $\tau^{\leftarrow}(M)$ is a reachable marking of N.

Note that for any $t \in T$ we have that $\tau^{\leftarrow}({}^\circ t) = {}^\bullet t$. Moreover, $\alpha(M)$ implies that

$$X, Y \subseteq M \wedge X \cap Y = \emptyset \quad \Rightarrow \quad \tau^{\leftarrow}(X) \cap \tau^{\leftarrow}(Y) = \emptyset \tag{1}$$

and hence

$$Y \subseteq M \quad \Rightarrow \quad \tau^{\leftarrow}(M \setminus Y) = \tau^{\leftarrow}(M) \setminus \tau^{\leftarrow}(Y)\ . \tag{2}$$

Let $t \in G$. Since t is enabled in M, we have ${}^\circ t \subseteq M$ and hence ${}^\bullet t = \tau^{\leftarrow}({}^\circ t) \subseteq \tau^{\leftarrow}(M)$. Given that N is contact-free and $\tau^{\leftarrow}(M) \in [M_0\rangle_N$, it follows that t is enabled in $\tau^{\leftarrow}(M)$.

Now let $t, u \in G$ with $t \neq u$. Then ${}^\circ t \cup {}^\circ u \subseteq M$ and ${}^\circ t \cap {}^\circ u = \emptyset$, so ${}^\bullet t \cap {}^\bullet u = \tau^{\leftarrow}({}^\circ t) \cap \tau^{\leftarrow}({}^\circ u) = \emptyset$, using (1). Given that ${}^\bullet t \cup {}^\bullet u \subseteq \tau^{\leftarrow}(M)$ and N is contact-free, it follows that also $t^\bullet \cap u^\bullet = \emptyset$ and hence t and u are independent.

Since $M' = (M \setminus {}^\circ G) \cup G^\circ$ we have $\tau^{\leftarrow}(M') = (\tau^{\leftarrow}(M) \setminus \tau^{\leftarrow}({}^\circ G)) \cup \tau^{\leftarrow}(G^\circ) = (\tau^{\leftarrow}(M) \setminus {}^\bullet G) \cup G^\bullet$ and hence $\tau^{\leftarrow}(M)\ [G\rangle_N\ \tau^{\leftarrow}(M')$.

Next we establish $\alpha(M')$. To this end, we may assume that G is a singleton set, for G must be finite—this follows since all (independent) transitions in G are enabled from the reachable marking $\tau^{\leftarrow}(M)$ of N, and N satisfies the finiteness restrictions imposed on nets in Section 2—and when $M[\{t_0, t_1, \ldots, t_n\}\rangle M'$ for some $n \geq 0$ then there are $M_1, M_2, \ldots, M_n$ with $M\ [\{t_0\}\rangle\ M_1[\{t_1\}\rangle\ M_2 \cdots M_n\ [\{t_n\}\rangle\ M'$, allowing us to obtain the general case by induction. So let $G = \{t\}$ with $t \in T$.

Above we have shown that $\tau^{\leftarrow}(M') \in [M_0\rangle_N$. We still need to prove that $\tau^{\leftarrow}(p) = \tau^{\leftarrow}(q) \Rightarrow p = q$ for all $p, q \in M'$. Assume the contrary, i.e. there are $p, q \in M'$ with $\tau^{\leftarrow}(p) = \tau^{\leftarrow}(q)$ but $p \neq q$. Since $\alpha(M)$, at least one of p and q—say p—must not be present in M. Thus $p \in t^\circ = t^\bullet \subseteq S$. As $\tau^{\leftarrow}(q) = \tau^{\leftarrow}(p) = p$ and $q \neq p$, it must be that $q \in S^\tau$. Hence $q \notin t^\circ$, so $q \in M$, and $p = \tau^{\leftarrow}(q) \in \tau^{\leftarrow}(M)$. As shown above, t is enabled in $\tau^{\leftarrow}(M)$. By the contact-freeness of N, $(\tau^{\leftarrow}(M) \setminus {}^\bullet t) \cap t^\bullet = \emptyset$, so $p \in {}^\bullet t$. Using that $p \notin M$, we find that $p \notin {}^\circ t \subseteq M$, so $p \not\equiv_D t$ and $p_t \in {}^\circ t \subseteq M$. As by construction ${}^\circ t \cap t^\circ = \emptyset$, we have $p_t \notin M'$, so $q \neq p_t$. Yet $\tau^{\leftarrow}(q) = p = \tau^{\leftarrow}(p_t)$, contradicting $\alpha(M)$.

"3": Let $t_s \in T^\tau$ such that $M\,[\{t_s\}\rangle_{I_D(N)}\,M'$. Then, by construction of $I_D(N)$, ${}^\circ t_s = \{s\} \wedge t_s{}^\circ = \{s_t\}$. Hence $M' = M \setminus \{s\} \cup \{s_t\}$ and $d(M') = d(M) - 1 \wedge \tau^{\leftarrow}(M') = \tau^{\leftarrow}(M)$. Moreover, $\alpha(M') \Leftrightarrow \alpha(M)$.

"4": Using 1–3, this follows by a straightforward induction on the number of transitions in the derivation $M_0 \stackrel{\sigma}{\Longrightarrow}_{I_D(N)} M'$. □

It follows that α exactly characterises the reachable markings of $I_D(N)$:

Lemma 6 Let N and $I_D(N)$ be as before and $M \subseteq S \cup S^\tau$.
Then $M \in [M_0\rangle_{I_D(N)}$ iff $\alpha(M)$.

Proof "Only if" follows from Lemma 5.4, and "if" follows by Lemmas 3 and 4. □

Using this we now prove Proposition 1 from Section 3:

Proposition 1 For any (contact-free) net $N = (S, T, F, M_0, \ell)$, and any choice of $\equiv_D$, the net $I_D(N)$ is contact-free, and satisfies the other requirements imposed on nets, listed in Section 2.

Proof Let $M \in [M_0\rangle_{I_D(N)}$. Then $\alpha(M)$, and hence $\tau^{\leftarrow}(M) \in [M_0\rangle_N$.

Consider any $t \in T$ with ${}^\circ t \subseteq M$. Assume $(M \setminus {}^\circ t) \cap t^\circ \neq \emptyset$. Since $t^\circ = t^\bullet \subseteq S$ let $p \in S$ be such that $p \in M \cap t^\circ$ and $p \notin {}^\circ t$. As N is contact-free we have $(\tau^{\leftarrow}(M) \setminus {}^\bullet t) \cap t^\bullet = \emptyset$, so since $p \in \tau^{\leftarrow}(M) \cap t^\bullet$ it must be that $p \in {}^\bullet t$. Hence $p_t \in {}^\circ t \subseteq M$ and we have $p \neq p_t$ yet $\tau^{\leftarrow}(p) = p = \tau^{\leftarrow}(p_t)$, violating $\alpha(M)$.

Now consider any $t_p \in T^\tau$ with ${}^\circ t_p \subseteq M$. As ${}^\circ t_p = \{p\}$ and $t_p{}^\circ = \{p_t\}$ we have that $(M \setminus {}^\circ t_p) \cap t_p{}^\circ \neq \emptyset$ only if $p \in M \wedge p_t \in M$. However, $\tau^{\leftarrow}(p) = p = \tau^{\leftarrow}(p_t)$ which would violate $\alpha(M)$.

This established the contact-freeness of $I_D(N)$. By construction, M_0 is finite, ${}^\circ t \neq \emptyset$ and ${}^\circ t$ and t° are finite for all $t \in T \cup T^\tau$, and s° is finite for all $s \in S \cup S^\tau$. □

The following lemma is a crucial step in the proof of Theorem 1.

Lemma 7 Let $N = (S, T, F, M_0, \ell)$ be a net without a distributed conflict w.r.t. a distribution D.
Let $M_1 \in [M_0\rangle_N$ and $M_1 \stackrel{\tau}{\longrightarrow}_{I_D(N)} M_2 \stackrel{\tau}{\longrightarrow}_{I_D(N)} \cdots \stackrel{\tau}{\longrightarrow}_{I_D(N)} M_n \stackrel{\tau}{\not\longrightarrow}_{I_D(N)}$ for some $n \geq 1$.
Then, $M_1 \stackrel{A}{\longrightarrow}_N$ iff $M_n \stackrel{A}{\longrightarrow}_{I_D(N)}$ for all $A \subseteq \text{Act}$.

Proof Suppose ${}^\bullet t \subseteq M_1$ but ${}^\circ t \nsubseteq M_n$ for some $t \in T$. For $p \in {}^\bullet t$ write $\hat{p}_t := p_t$ if $p \not\equiv_D t$ and $\hat{p}_t = p$ otherwise. Then ${}^\circ t = \{\hat{p}_t \mid p \in {}^\bullet t\}$. Pick $p \in {}^\bullet t$ such that $\hat{p}_t \notin M_n$. As $M_n \stackrel{\tau}{\not\longrightarrow}_{I_D(N)}$ we also have $p \notin M_n$. Let $1 \leq i < n$ be the last index such that $p \in M_i$ or $\hat{p}_t \in M_i$. Then $M_i\,[\{u_p\}\rangle_{I_D(N)}\,M_{i+1}$ for some $u \in T$ with $u \neq t$, $p \in {}^\bullet u$ and $p \not\equiv_D u$. But this would constitute a distributed conflict w.r.t. D. ↯

It follows that $M_1\,[t\rangle_N$ implies $M_n\,[t\rangle_{I_D(N)}$ for all $t \in T$. Moreover, it follows immediately from the construction of $I_D(N)$ that if two transitions $t, u \in T$ are independent in N, then they are also independent in $I_D(N)$. Hence $M_1\,[G\rangle_N$ implies $M_n\,[G\rangle_{I_D(N)}$ for all $G \subseteq T$. Thus $M_1 \stackrel{A}{\longrightarrow}_N$ implies $M_n \stackrel{A}{\longrightarrow}_{I_D(N)}$.

For the reverse direction, observe that $\alpha(M_1)$ and $\tau^{\leftarrow}(M_1) = M_1$ because $M_1 \in [M_0\rangle_N$. Hence $\alpha(M_n)$ and $\tau^{\leftarrow}(M_n) = M_1$ by Lemma 5.3 and $M_n \stackrel{A}{\longrightarrow}_{I_D(N)}$ implies $M_1 \stackrel{A}{\longrightarrow}_N$ for all A by Lemma 5.2. □

Theorem 1 Let $N = (S,T,F,M_0,\ell)$ be a plain net, and $\mathscr{Q}$ a requirement on distributions of nets. Then N is behaviourally $\mathscr{Q}$-asynchronous iff it is structurally $\mathscr{Q}$-asynchronous.

Proof "Only if": Suppose N fails to be structurally $\mathscr{Q}$-asynchronous. Let D be a distribution on N meeting the requirement $\mathscr{Q}$. Then N has a distributed conflict with respect to D, i.e.

$$\exists t,u \in T\ \exists p \in {}^\bullet t \cap {}^\bullet u.\ t \neq u \wedge p \not\equiv_D u \wedge \exists M \in [M_0\rangle_N.\ {}^\bullet t \subseteq M\ .$$

We need to show that $I_D(N) \not\approx_{\mathscr{R}} N$.

Let $M \in [M_0\rangle_N$ be such that ${}^\bullet t \subseteq M$ and let $\sigma \in \text{Act}^*$ be such that $M_0 \overset{\sigma}{\Longrightarrow}_N M$. Then N has a step ready pair $<\sigma, X>$ with $\{\ell(t)\} \in X$. As plain nets are deterministic, M is the only marking of N with the property that $M_0 \overset{\sigma}{\Longrightarrow}_N M$. Hence N has exactly once step ready pair of the form $<\sigma, X>$, and it satisfies $\{\ell(t)\} \in X$.

Lemma 3 yields $M_0 \overset{\sigma}{\Longrightarrow}_{I_D(N)} M$. Let $M_1 := (M\setminus\{p\})\cup\{p_u\}$. Then $M\ [u_p\rangle_{I_D(N)}\ M_1$ by Definition 6, so $M \overset{\tau}{\longrightarrow} M_1$. By Lemma 5.3, we have $M_1 \overset{\tau}{\longrightarrow}_{I_D(N)} M_2 \overset{\tau}{\longrightarrow}_{I_D(N)} \cdots \overset{\tau}{\longrightarrow}_{I_D(N)} M_n \overset{\tau}{\not\longrightarrow}_{I_D(N)}$ for some $n \leq d(M) \in \mathbb{N}$. As $v^\circ \subseteq S^\tau$ for all $v \in T^\tau$, we have $p \notin M_i$ for $i = 1,2,\ldots,n$. Moreover, in case $p \not\equiv t$ we have $p_t \in v^\circ$ only if $p \in {}^\circ v$; hence also $p_t \notin M_i$ for $i = 1,2,\ldots,n$. It follows that ${}^\circ t \not\subseteq M_n$. Thus $I_D(N)$ has a step ready pair $<\sigma, X>$ with $\{\ell(t)\} \notin X$. We find that $\mathscr{R}(I_D(N)) \neq \mathscr{R}(N)$.

"If": Suppose N is structurally $\mathscr{Q}$-asynchronous, i.e. there is a distribution D on N meeting the requirement $\mathscr{Q}$, such that N has no distributed conflicts with respect to D. We show that $\mathscr{R}(I_D(N)) = \mathscr{R}(N)$.

''$\supseteq$'': Let $<\sigma, X> \in \mathscr{R}(N)$. Then there is a marking M of N such that $M_0 \overset{\sigma}{\Longrightarrow}_N M$, $M \overset{A}{\longrightarrow}_N$ for all $A \in X$ and $M \overset{A}{\not\longrightarrow}_N$ for all $A \notin X$. Lemma 3 yields $M_0 \overset{\sigma}{\Longrightarrow}_{I_D(N)} M$. By Lemma 5.3, we have $M \overset{\tau}{\longrightarrow}_{I_D(N)} M_1 \overset{\tau}{\longrightarrow}_{I_D(N)} M_2 \overset{\tau}{\longrightarrow}_{I_D(N)} \cdots \overset{\tau}{\longrightarrow}_{I_D(N)} M_n \overset{\tau}{\not\longrightarrow}_{I_D(N)}$ for some $0 \leq n \leq d(M) \in \mathbb{N}$. Now Lemma 7 yields $<\sigma, X> \in \mathscr{R}(I_D(N))$.

"$\subseteq$": Let $<\sigma, X> \in \mathscr{R}(I_D(N))$. Then there is a marking M of $I_D(N)$ such that $M_0 \overset{\sigma}{\Longrightarrow}_{I_D(N)} M$, $M \overset{\tau}{\not\longrightarrow}_{I_D(N)}$, and $M \overset{A}{\longrightarrow}_{I_D(N)}$ iff $A \in X$. Lemma 5.4 yields $M_0 \overset{\sigma}{\Longrightarrow}_N \tau^{\leftarrow}(M) \wedge \alpha(M)$ and Lemma 4 gives $\tau^{\leftarrow}(M) \overset{\tau}{\longrightarrow}{}^*_{I_D(N)} M$. Now Lemma 7 yields $<\sigma, X> \in \mathscr{R}(N)$. □

B The Transition-Controlled-Choice Implementation

In this appendix we show that the transition-controlled-choice implementation of any net N is step readiness equivalent to N. To this end we use the following result.

Lemma 8 Let $N = (S,T,F,M_0,\ell)$ and $N' = (S',T',F',M'_0,\ell')$ be two nets, and $\ell'(t) \neq \tau$ for $t \in T'$. Suppose there is a function $\tau^{\Leftarrow} : \mathcal{P}(S) \to \mathcal{P}(S')$ from the markings of N to the markings of N', a *distance* function $d : \mathcal{P}(S) \to \mathbb{N} \cup \{\infty\}$ and a predicate $\beta \subseteq \mathcal{P}(S)$ such that

$$\beta(M_0) \wedge \tau^{\Leftarrow}(M_0) = M'_0 \tag{1}$$
$$\beta(M_1) \wedge M_1 \overset{\tau}{\longrightarrow}_N M_2 \ \Rightarrow\ \beta(M_2) \wedge \tau^{\Leftarrow}(M_2) = \tau^{\Leftarrow}(M_1) \wedge d(M_1) > d(M_2) \tag{2}$$
$$\beta(M_1) \wedge M_1 \overset{A}{\longrightarrow}_N M_2 \ \Rightarrow\ \beta(M_2) \wedge \tau^{\Leftarrow}(M_1) \overset{A}{\longrightarrow}_{N'} \tau^{\Leftarrow}(M_2) \tag{3}$$
$$\beta(M_1) \wedge d(M_1) > 0 \ \Rightarrow\ M_1 \overset{\tau}{\longrightarrow}_N \tag{4}$$
$$\beta(M_1) \wedge d(M_1) = 0 \wedge \tau^{\Leftarrow}(M_1) \overset{A}{\longrightarrow}_{N'} M'_2 \ \Rightarrow\ \exists M_2.\ M_1 \overset{A}{\longrightarrow}_N M_2 \wedge M'_2 = \tau^{\Leftarrow}(M_2)\ . \tag{5}$$

Then $N \approx_{\mathscr{R}} N'$.

Proof "$\mathscr{R}(N) \subseteq \mathscr{R}(N')$": Conditions (1–5) allow any step ready pair $<\sigma, X>$ of N to be mimicked step for step by N'. To be precise, if $<\sigma, X> \in \mathscr{R}(N)$, then there is a marking M_1 with $M_0 \overset{\sigma}{\Longrightarrow}_N M_1$, $M_1 \overset{\tau}{\not\longrightarrow}_N$, $M_1 \overset{A}{\longrightarrow}_N$ for any $A \in X$ and $M_1 \overset{A}{\not\longrightarrow}_N$ for any $A \notin X$. As for all reachable markings M_1 of N, we have $\beta(M_1)$. Now (1–3) imply $M'_0 \overset{\sigma}{\Longrightarrow}_{N'} \tau^{\Leftarrow}(M_1)$. Furthermore, (3) implies $\tau^{\Leftarrow}(M_1) \overset{A}{\longrightarrow}_{N'}$ for any $A \in X$, (4) implies $d(M_1) = 0$, and hence (5) implies $\tau^{\Leftarrow}(M_1) \overset{A}{\not\longrightarrow}_{N'}$ for any $A \notin X$.

"$\mathscr{R}(N') \subseteq \mathscr{R}(N)$": From conditions (2–5) we infer:

$$\beta(M_1) \;\Rightarrow\; \exists M_2.\ M_1 \overset{\tau}{\longrightarrow}{}^{*}_{N} M_2 \wedge M_2 \overset{\tau}{\not\longrightarrow} \wedge \beta(M_2) \wedge \tau^{\Leftarrow}(M_2) = \tau^{\Leftarrow}(M_1) \tag{6}$$

$$\beta(M_1) \wedge \tau^{\Leftarrow}(M_1) \overset{A}{\longrightarrow}_{N'} M'_2 \;\Rightarrow\; \exists M_2.\ M_1 \overset{\tau}{\longrightarrow}{}^{*}_{N} \overset{A}{\longrightarrow}_N M_2 \wedge \beta(M_2) \wedge M'_2 = \tau^{\Leftarrow}(M_2) \tag{7}$$

The first statement follows by repeated application of (2); the second by repeated application of (4) and (2), then (5) and (3). Conditions (1) and (7) imply that every reachable marking of N' is of the form $\tau^{\Leftarrow}(M)$ with M a reachable marking of N. Moreover, (1), (6) and (7) yield, for $\sigma \in \text{Act}^*$,

$$M'_0 \overset{\sigma}{\Longrightarrow}_{N'} M' \;\Rightarrow\; \exists M.\ M_0 \overset{\sigma}{\Longrightarrow}_N M \wedge M \overset{\tau}{\not\longrightarrow} \wedge \beta(M) \wedge M' = \tau^{\Leftarrow}(M)\,.$$

In combination with (3–5) this implies that any ready pair $<\sigma, X>$ of N' is also a ready pair of N. □

In fact, conditions (1–5) are strong enough to show that N and N' are semantically equivalence in various other ways as well; in particular $\tau^{\Leftarrow}$ constitutes a *branching bisimulation* between N and N', as defined in [7]. In order to apply Lemma 8, we will take N to be the transition-controlled-choice implementation of a given net N' that features no transitions labelled τ.

Definition 19

Let $N' = (S, T, F', M_0, \ell')$ be a net with $\ell'(t) \neq \tau$ for $t \in T'$, and $N = (S \cup S^\tau, T \cup T^\tau, F, M_0, \ell)$ its transition-based-choice implementation.

The function $\tau^{\Leftarrow} : \mathcal{P}(S \cup S^\tau) \to \mathcal{P}(S)$ is defined by

$$\tau^{\Leftarrow}(M) := (M \cap S) \cup \{s \mid s \in S,\ \{s^{[t]} \mid t \in s^\bullet\} \subseteq M\} \cup \{s \mid s \in t^\bullet \wedge (t) \in M\}\,.$$

The function $\tau^{\Longleftarrow} : \mathcal{P}(S \cup S^\tau) \to \mathcal{P}(S)$ is defined by

$$\tau^{\Longleftarrow}(M) := (M \cap S) \cup \{s \mid s \in S, \{s^{[t]} \mid t \in s^\bullet\} \cap M \neq \emptyset\} \cup \{s \mid s \in {}^\bullet t \wedge (t) \in M\}\,.$$

The function $d : \mathcal{P}(S \cup S^\tau) \to \mathbb{N} \cup \{\infty\}$ is defined by

$$d(M) := |M \cap S| + \sum_{(t) \in M} (1 + |t^\bullet|) + \sum_{s^{[u]}_t \in M} 1\,.$$

The predicate $\beta \subseteq \mathcal{P}(S \cup S^\tau)$ is defined by

$$\beta(M) :\Leftrightarrow \tau^{\Longleftarrow}(M) \in [M_0\rangle_{N'} \wedge \tag{β_1}$$

$$\left(s^{[t]} \in M \Rightarrow s \notin M\right) \wedge \tag{β_2}$$

$$\left(s^{[u]} \in M \wedge s^{[t]} \notin M \Rightarrow \exists v \in s^\bullet.\ s^{[u]}_v \in M\right) \wedge \tag{β_3}$$

$$\left((t), (u) \in M \wedge t \neq u \Rightarrow {}^\bullet t \cap {}^\bullet u = \emptyset\right) \wedge \tag{β_4}$$

$$\left(s^{[u]}_t \in M \Rightarrow \overline{s}^{[u]}_t \notin M \wedge s^{[u]}, (t) \in M\right) \wedge \tag{β_5}$$

$$\left(\overline{s}^{[u]}_t \in M \Rightarrow (t) \in M\right) \wedge \tag{β_6}$$

$$\left((t) \in M \Rightarrow \forall s \in {}^\bullet t,\ u \in s^\bullet.\ s, s^{[t]} \notin M \wedge \left([u] \neq [t] \Rightarrow s^{[u]}_t \in M \vee \overline{s}^{[u]}_t \in M\right)\right). \tag{β_7}$$

Some conjuncts in the definition of $\beta(M)$ are universally quantified over (some of) s, t and u; we write

- $\beta_i^{s,t,u}(M)$ to say that marking M satisfies the instance of β_i for the specific values s, t and u,
- $\beta_i^{s}(M)$ for $\forall t,u \in s^\bullet.\ \beta_i^{s,t,u}(M)$,
- and $\beta_i(M)$ for $\forall s \in S.\ \beta_i^{s}(M)$,

so that $\beta(M)$ iff $\beta_1(M) \wedge \beta_2(M) \wedge \beta_3(M) \wedge \beta_4(M) \wedge \beta_5(M) \wedge \beta_6(M) \wedge \beta_7(M)$.

Lemma 9 Let N', N, $\tau^{\Leftarrow}$, $\tau^{\Longleftarrow}$, d, and β be as in Definition 19.
Then N is a net as defined in Section 2 and the clauses (1)–(5) of Lemma 8 hold.

Proof Again, we use $^\circ x$ and x° instead of $^\bullet x$ and $x^\bullet$ when making assertions about the flow relation of N (the implementation). Given that $^\bullet t \neq \emptyset$ and $^\bullet t$ and $t^\bullet$ are finite for all $t \in T$ and $s^\bullet$ is finite for all $s \in S$, by construction we have $^\circ t \neq \emptyset$ and $^\circ t$ and t° are finite for all $t \in T \cup T^\tau$ and s° is finite for all $s \in S \cup S^\tau$. As N has the same initial marking as N', it must be finite. In order to show that N is contact-free, we must show that for each reachable marking $M \in [M_0\rangle_N$ the following four properties are satisfied:

(i) If $s \in M$ then $s^{[t]} \notin M$ for all $t \in s^\bullet$.
(ii) If $s_t^{[u]}, s^{[u]} \in M$ then $\overline{s}_t^{[u]} \notin M$.
(iii) If $s^{[t]} \in M$ for all $s \in {}^\bullet t$ then $\textcircled{t} \notin M$ and $s_t^{[u]} \notin M$ for all $s \in {}^\bullet t$ and $u \in s^\bullet$ with $[u] \neq [t]$.
(iv) If $\textcircled{t} \in M$ and $\overline{s}_t^{[u]} \in M$ for all $s \in {}^\bullet t$ and $u \in s^\bullet$ with $[u] \neq [t]$, then $M \cap t^\bullet = \emptyset$.

We proceed to show that all four properties are implied by $\beta(M)$. This entails that the contact-freeness of N will follow immediately from the validity of clauses (1)–(3) of Lemma 8.

Property (i) follows immediately from $\beta_2(M)$ and (ii) from $\beta_5(M)$. The claim $\textcircled{t} \notin M$ of property (iii) follows from $\beta_7(M)$, and using this the claim $s_t^{[u]} \notin M$ from $\beta_5(M)$. For (iv), assume, towards a contradiction, that $\textcircled{t} \in M$, yet $s \in M \cap t^\bullet$. Then $^\bullet t \subseteq \tau^{\Longleftarrow}(M)$. Now $\beta_1(M)$ and the contact-freeness of N' gives $(\tau^{\Longleftarrow}(M) \setminus {}^\bullet t) \cap t^\bullet = \emptyset$. As $s \in M \cap t^\bullet \subseteq \tau^{\Longleftarrow}(M) \cap t^\bullet$ we obtain $s \in {}^\bullet t$, contradicting $\beta_7(M)$.

It remains to show the validity of clauses (1)–(5). Clause (1) follows directly from the definitions.

Clause (2): Assume $\beta(M_1)$. As remarked in Section 2, reachable markings of N' are finite, so by $\beta_1(M_1)$ $M_1 \cap S$ is finite and M_1 contains only finitely many places of the form $\textcircled{t}$ (using $\beta_4(M_1)$ and that $^\bullet t \neq \emptyset$ for $t \in T$). Since for a given t, using that $^\bullet t$ and $s^\bullet$ are finite, there are only finitely many places $s_t^{[u]}$ in N, it follows by $\beta_5(M_1)$ that M_1 contains only finite many places of the form $s_t^{[u]}$. From this we conclude that $d(M_1)$ is finite. We proceed by a case distinction over all transitions labelled τ.

Assume $M_1\ [\boxed{s}\rangle_N\ M_2$. Then $M_2 = (M_1 \setminus \{s\}) \cup \{s^{[t]} \mid t \in s^\bullet\}$ and $\tau^{\Leftarrow}(M_2) = \tau^{\Leftarrow}(M_1)$ as well as $\tau^{\Longleftarrow}(M_2) = \tau^{\Longleftarrow}(M_1)$. Moreover, $d(M_2) = d(M_1) - 1$ as $s \in M_1 \cap S$ but $s \notin M_2$ and the $s^{[t]}$ don't contribute to d. It remains to check that $\beta(M_2)$. We will do that for each of the six conjuncts separately. The validity of β_1 is clearly preserved, in the sense that $\beta_1(M_1)$ implies $\beta_1(M_2)$. The same holds for β_4 and β_6, as places of the form $\textcircled{t}$ and $\overline{s}_t^{[u]}$ do not figure as pre- or postplaces of the transition $\boxed{s}$. Requirement $\beta_2^s(M_2)$ simply holds, as $s \notin M_2$, whereas for $s' \neq s$ requirement $\beta_2^{s'}(M_2)$ is preserved. In the same way we obtain $\beta_3(M_2)$, $\beta_5(M_2)$ and $\beta_7(M_2)$.

Assume $M_1\ [t_s^{[u]}\rangle_N\ M_2$. Then $M_2 = (M_1 \setminus \{s_t^{[u]}, s^{[u]}\}) \cup \{\overline{s}_t^{[u]}\}$. From $s_t^{[u]} \in M_1$ we obtain $\textcircled{t} \in M_1$ by $\beta_5^{s,t,u}(M_1)$ and $s^{[t]} \notin M_1$ by $\beta_7(M_1)$. Hence the removal of any $s^{[u]}$ does not affect $\tau^{\Leftarrow}$, and we have $\tau^{\Leftarrow}(M_2) = \tau^{\Leftarrow}(M_1)$. As the only change in summands contributing to d is the removal of $s_t^{[u]}$, we have $d(M_2) = d(M_1) - 1$. Since $\textcircled{t} \in M_1$, the removal of $s^{[u]}$ does not affect $\tau^{\Longleftarrow}$ either, and we have $\tau^{\Longleftarrow}(M_2) = \tau^{\Longleftarrow}(M_1)$. Hence β_1 is preserved. Requirement $\beta_2^{s,u}(M_2)$ holds (since $s^{[u]} \notin M_2$)

and ${\beta_2}^{s',t'}$ for $s' \neq s$ or $t' \neq u$ is preserved. Likewise, ${\beta_3}^{s,t',u}(M_2)$ holds (since $s^{[u]} \notin M_2$) and ${\beta_3}^{s',t',u'}$ with $s' \neq s$ or $u' \neq u$ is preserved. Requirement ${\beta_5}^{s,t,u}(M_2)$ holds (because $s_t^{[u]} \notin M_2$), and ${\beta_5}^{s',t',u'}$ with $s' \neq s$ or $u' \neq u$ is preserved. As for ${\beta_5}^{s,t',u}$ with $t' \neq t$, by $\beta_4(M_1)$ we have $t' \notin M_1$ and hence by ${\beta_5}^{s,t',u}(M_1)$ it must be that $s_{t'}^{[u]} \notin M_1$, and thus $s_{t'}^{[u]} \notin M_2$. This yields ${\beta_5}^{s,t',u}(M_2)$. Since $t \in M_1$ we have $t \in M_2$ and hence ${\beta_6}^{s,t,u}(M_2)$ holds. All other instances of β_6 are preserved. Requirements β_4 and β_7 are preserved as well.

Assume $M_1\ [t'\rangle_N\ M_2$. Then $M_2 = (M_1 \setminus \{t, \overline{s}_t^{[u]} \mid s \in {}^\bullet t,\ u \in s^\bullet,\ [u] \neq [t]\}) \cup \{s \mid s \in t^\bullet\}$ and $\tau^{\Leftarrow}(M_2) = \tau^{\Leftarrow}(M_1)$. Again $d(M_2) = d(M_1) - 1$ as the single t contributed $1 + |t^\bullet|$ whereas all the newly produced places s together contribute $|t^\bullet|$. As $t \in M_1$ we have ${}^\bullet t \in \tau^{\Longleftarrow}(M_1)$. Moreover, for $s \in {}^\bullet t$ and $u, v \in s^\bullet$, $[u] \neq [t]$, $v \neq t$ we have $t, \overline{s}_t^{[u]} \in M_1$, so $s, s^{[t]}, v \notin M_1$ by $\beta_7(M_1)$ and $\beta_4(M_1)$ and $s_t^{[u]}, s_v^{[u]}, s^{[u]} \notin M_1$ by $\beta_5(M_1)$ and $\beta_3(M_1)$. Hence t is the only place in M_1 that contributes $s \in {}^\bullet t$ to $\tau^{\Longleftarrow}(M_1)$. Therefore $\tau^{\Longleftarrow}(M_2) = (\tau^{\Longleftarrow}(M_1) \setminus {}^\bullet t) \cup t^\bullet$. Hence $\tau^{\Longleftarrow}(M_1)\ [\{t\}\rangle_{N'}\ \tau^{\Longleftarrow}(M_2)$, so β_1 is preserved. Requirements β_3, β_4, β_5 and β_6 are easily seen to be preserved as well. Since N' is contact-free, we have $(\tau^{\Longleftarrow}(M_1) \setminus {}^\bullet t) \cap t^\bullet = \emptyset$, using $\beta_1(M_1)$. So for $s \in t^\bullet$ we have either $s \notin \tau^{\Longleftarrow}(M_1)$ or $s \in {}^\bullet t$. Either possibility implies $s^{[u]} \notin M_1$ for $u \in s^\bullet$, and $v \notin M_1$ for $v \in s^\bullet$, $v \neq t$. Hence $s^{[u]}, u \notin M_2$ for $u \in s^\bullet$. Using this, also β_2 and β_7 turn out to be preserved.

Clause (3): Assume $\beta(M_1) \wedge M_1\ [G\rangle_N\ M_2$ with $\ell(t) \neq \tau$ for all $t \in G$. Then

$$M_2 = (M_1 \setminus {}^\circ G) \cup G^\circ = M_1 \setminus \{s^{[t]} \mid s \in {}^\bullet G\} \cup \{t, s_t^{[u]} \mid t \in G,\ s \in {}^\bullet t,\ u \in s^\bullet,\ [u] \neq [t]\}.$$

For all $t \in G$ and $s \in {}^\bullet t$ we have $s^{[t]} \in M_1$ and hence $s \in \tau^{\Longleftarrow}(M_1)$. Thus $\tau^{\Longleftarrow}(M_1)[t\rangle_{N'}$.

Claim 1: Let $t \in G$, $s \in {}^\bullet t$ and $u, v \in s^\bullet$. Then $v \notin M_1$ and $s^{[u]} \in M_1$.

Proof: Assume, towards a contradiction, that $v \in M_1$. Then ${}^\bullet v \subseteq \tau^{\Longleftarrow}(M_1)$ and thus $\tau^{\Longleftarrow}(M_1)[v\rangle_{N'}$. As $s \in {}^\bullet t \cap {}^\bullet v$ we have $\neg\tau^{\Longleftarrow}(M_1)[t, v\rangle_{N'}$, so $\beta_1(M_1)$ and Definition 15 yield $t \# v$, and hence $[t] = [v]$. Nevertheless, $\beta_7(M_1)$ gives $s^{[v]} \notin M_1$, whereas $s^{[t]} \in M_1$. $\lightning$

Next assume that $s^{[u]} \notin M_1$. Then ${\beta_3}^{s,u,t}(M_1)$ yields $\exists v \in s^\bullet.\ s_v^{[t]} \in M_1$, and ${\beta_5}^{s,v,t}(M_1)$ gives $v \in M_1$. $\lightning$

Claim 2: Let $t_1, t_2 \in G$ with $t_1 \neq t_2$. Then ${}^\bullet t_1 \cap {}^\bullet t_1 = \emptyset$.

Proof: Assume, towards a contradiction, that $s \in {}^\bullet t_1 \cap {}^\bullet t_2$. Then $\tau^{\Longleftarrow}(M_1)[t_1\rangle_{N'}$ and $\tau^{\Longleftarrow}(M_1)[t_2\rangle_{N'}$, but $\neg\tau^{\Longleftarrow}(M_1)[t_1, t_2\rangle_{N'}$, so $\beta_1(M_1)$ and Definition 15 yield $t_1 \# t_2$, and hence $[t_1] = [t_2]$. But this implies $s^{[t_1]} = s^{[t_2]} \in {}^\circ t_1 \cap {}^\circ t_2$, contradicting $M_1\ [G\rangle_N$. $\lightning$

Claim 3: Let $t \in G$, $s \in {}^\bullet t$ and $v \in {}^\bullet s$. Then $s, v \notin M_1$.

Proof: Since $s^{[t]} \in M_1$ we have $s \notin M_1$ by $\beta_2(M_1)$. Assume, towards a contradiction, that $v \in M_1$. Then ${}^\bullet v \subseteq \tau^{\Longleftarrow}(M_1) \in [M_0\rangle_{N'}$, using $\beta_1(M_1)$. As N' is contact-free, we have $(\tau^{\Longleftarrow}(M_1) \setminus {}^\bullet v) \cap v^\bullet = \emptyset$. So since $s \in \tau^{\Longleftarrow}(M_1) \cap v^\bullet$ it must be that $s \in {}^\bullet v$. But then $v \notin M_1$ by Claim 1. $\lightning$

Claim 1 implies that ${}^\bullet G \subseteq \tau^{\Leftarrow}(M_1)$, and Claim 2 yields $\tau^{\Leftarrow}(M_1)\ [G\rangle_{N'}\ M_2'$ for some M_2'. By Claim 3 we have $\tau^{\Leftarrow}(M_1 \setminus {}^\circ G) = \tau^{\Leftarrow}(M_1) \setminus {}^\bullet G$ and thus

$$\tau^{\Leftarrow}(M_2) = \tau^{\Leftarrow}((M_1 \setminus {}^\circ G) \cup G^\circ) = (\tau^{\Leftarrow}(M_1) \setminus {}^\bullet G) \cup G^\bullet = M_2'.$$

It remains to check that $\beta(M_2)$. First of all, $\tau^{\Longleftarrow}(M_2) = \tau^{\Longleftarrow}(M_1)$ and hence β_1 is preserved. It is easy to see that β_2, β_6 and β_7 are preserved. Requirement ${\beta_3}^s$ for $s \notin {}^\bullet G$ is also preserved, whereas ${\beta_3}^s(M_2)$ for $s \in {}^\bullet t$, $t \in G$ holds with $v := t$. Requirement β_4 may fail to be preserved only if $\exists t_1, t_2 \in G$ with $t_1 \neq t_2$ and ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$ or if $\exists t \in G$ and $v \in M_1$ with ${}^\bullet t \cap {}^\bullet v \neq \emptyset$. These cases are ruled out by Claims 2 and 1. Requirement ${\beta_5}^s$ with $s \notin {}^\bullet G$ is preserved. Since there is no $v \in M_1$ with ${}^\bullet G \cap {}^\bullet v \neq \emptyset$, by $\beta_5(M_1)$ and $\beta_6(M_1)$ there are no $s_v^{[u]}, \overline{s}_v^{[u]} \in M_1$ with $s \in {}^\bullet G$. Moreover, for all $t \in G$, $s \in {}^\bullet t$ and $u \in s^\bullet$ with $[u] \neq [t]$ we have $s^{[u]} \in M_1$ and hence $s^{[u]} \in M_2$. Thus we obtain $\beta_5^s(M_2)$ for $s \in {}^\bullet G$.

Clause (4): By a case distinction on the three summands of $d(M_1)$.

Assume $\exists s \in M_1 \cap S$. Then $M_1[\boxed{s}\rangle_{N'}$.

Assume $\exists s_t^{[u]} \in M_1$. Then by $\beta_5(M_1)$ also $s^{[u]} \in M_1$ and hence $M_1[t_s^{[u]}\rangle_{N'}$.

Assume $\exists t \in M_1$ but $\neg\exists s_t^{[u]} \in M_1$. Then by $\beta_7(M_1)$ also $\exists \overline{s}_t^{[u]} \in M_1$ for all $s \in {}^\bullet t$ and $u \in s^\bullet$ with $[u] \neq [t]$. Thus $M_1[t'\rangle_{N'}$.

Clause (5): $d(M_1) = 0$ implies $M_1 \cap S = \emptyset$ and M_1 does not contain places of the form t or $s_t^{[u]}$. By $\beta_6(M_1)$ it doesn't contain places of the form $\overline{s}_t^{[u]}$ either. Hence all places in M_1 have the form $s^{[t]}$ for $s \in S$ and $t \in s^\bullet$. Moreover, by $\beta_3(M_1)$, for any $s \in S$ either M_1 contains all places $s^{[t]}$ with $t \in s^\bullet$ or none. Thus $M_1 = \{s^{[t]} \mid s \in \tau^{\Leftarrow}(M_1),\ t \in s^\bullet\}$. Using this, when $\tau^{\Leftarrow}(M_1)\ [G\rangle_{N'}\ M_2'$ for $G \subseteq T$, there is a unique M_2 such that $M_1\ [G\rangle_N\ M_2$. It remains to show that $\tau^{\Leftarrow}(M_2) = M_2'$.

First of all, note that $M_2 \cap S = \emptyset$. Secondly, we have

$$\{s \mid s \in S,\ \{s^{[t]} \mid t \in s^\bullet\} \subseteq M_2\} = \{s \mid s \in \tau^{\Leftarrow}(M_1),\ s \notin {}^\bullet G\} = \tau^{\Leftarrow}(M_1) \setminus {}^\bullet G.$$

Finally, $\{s \mid s \in t^\bullet \wedge t \in M_2\} = \{s \mid s \in t^\bullet \wedge t \in G\} = G^\bullet$.

Thus, applying Definitions 19 and 2, $\tau^{\Leftarrow}(M_2) = (\tau^{\Leftarrow}(M_1) \setminus {}^\bullet G) \cup G^\bullet = M_2'$. □

Definition 20 For N a net and i and action, let N/i be the net obtained by renaming all occurrences of i into τ.

Proposition 2 If $N \approx_{\mathscr{R}} N'$ then $N/i \approx_{\mathscr{R}} N'/i$.

Proof $<\sigma, X>$ is a step ready pair of N/i iff N has a step ready pair $<\rho, X>$, where the sequence σ can be obtained from ρ by deleting all i's, and $\{i\} \notin X$. □

Theorem 2 Any net is step readiness equivalent to its transition-controlled-choice implementation.

Proof Let $N_\tau' = (S, T, F', M_0, \ell_\tau')$ be a net and $N_\tau = (S \cup S^\tau, T \cup T^\tau, F, M_0, \ell_\tau)$ its transition-controlled-choice implementation. Obtain N' from N_τ' and N from N_τ by changing all τ-labels of transitions in T—but not those in T^τ—into i. Thus $N = (S \cup S^\tau, T \cup T^\tau, F, M_0, \ell)$ where ℓ satisfies $\ell(t) = \tau$ if $t \in T^\tau$; $\ell(t) = i$ if $t \in T$ and $\ell_\tau(t) = \tau$; and $\ell(t) = \ell_\tau(t)$ otherwise. Then N is still the transition-controlled-choice implementation of N', and moreover N' has no τ-labels. Furthermore, $N'/i = N_\tau'$ and $N/i = N_\tau$. Lemmas 8 and 9 yield $N \approx_{\mathscr{R}} N'$. So by Proposition 2 we obtain $N/i \approx_{\mathscr{R}} N'/i$, which is $N_\tau \approx_{\mathscr{R}} N_\tau'$. □