Technische Universität Berlin

Forschungsberichte der Fakultät IV – Elektrotechnik und Informatik

An Algebraic Approach to Timed Petri Nets with Applications to Communication Networks

Extended Version

Pascal Lingnau
Karsten Gabriel
Tony Modica

Technische Universität Berlin

Bericht-Nr. 2012 – 02
ISSN 1436-9915

An Algebraic Approach to Timed Petri Nets with Applications to Communication Networks

Extended Version

Pascal Lingnau
Karsten Gabriel
Tony Modica

Technische Universität Berlin, Germany

{pasling,kgabriel,modica}@cs.tu-berlin.de

Bericht-Nr. 2012/02
ISSN 1436-9915

Contents

An Algebraic Approach to Timed Petri Nets with Applications to Communication Networks

Extended Version

Pascal Lingnau Karsten Gabriel Tony Modica

Abstract

In this report, we define a formalism for a time-extension to algebraic place/transition (P/T) nets. This allows time durations to be assigned to the transitions of a P/T net, representing delays present in the systems that are being modelled, which in turn influence (restrict) the firing behaviour of the nets. This is especially useful when modelling time-dependent systems.

The new contribution of this approach is the definition of categories for the timed net classes of timed P/T nets, timed P/T systems and timed P/T states. Moreover, we define functorial relations between these categories as well as functorial relations to categories of untimed P/T nets and systems.

The first main result is the formalisation of morphisms for all three net classes that preserve firing behaviour. The second main result is the equivalence of the categories of timed P/T systems and states, establishing a relation between structurally identical nets with a time offset. As a third main result we formalise structuring techniques for timed P/T nets and show that timed P/T nets fit in the framework of $\mathcal{M}$-adhesive categories.

1 Introduction

Petri nets are a formalism widely used for modelling and analysing systems and processes. First introduced by Carl Adam Petri in [Pet62], the notion of Petri nets (and P/T nets in particular) has been refined and extended over the time [Rei85, Rei91, MM90]. Different approaches, as well as extensions and enhancements exist, including algebraic high-level (AHL-)Nets [EHP$^+$02, Ehr04], P/T nets with individual tokens (PTI nets), [MGE$^+$10], and coloured Petri nets (CPNs) [Jen97, JKW07, JK09], among others.

Algebraic high-level nets are based on a combination of P/T nets and algebraic specifications, using data types and values, as well as terms and conditions defined by the specification which influence the firing behaviour. PTI nets are P/T nets with individual tokens, while coloured Petri nets use ML-data types and -expressions to control the firing of transitions and include data with each token.

One aspect often needed when modelling systems of any kind is time-based analysis, especially for real-time or in general time-critical systems. These include (but are not limited to) embedded systems monitoring and controlling industrial appliances and real-time communication over networks.

P/T nets do not inherently provide a way to model the passing of time or to restrict the firing behaviour with regards to passing time. In order to be able to model time-dependent systems using P/T nets, the notion of P/T nets has to be modified to respect durations of events in the system, effectively making transitions “take time”.

The modelling of time-critical systems has always been an important topic when it comes to the planning and development of (especially) real-time software and hardware systems and systems in general that are under some kind of time constraint. Being able to analyse a model with respect to temporal aspects as well as reliability (i.e. universal reachability of certain systems states, thus ruling out the possibility of deadlocks) is crucial when dealing with these kinds of projects.

There have been several approaches to including a notion of time in P/T nets in the past, such as time Petri nets [BD91] or deterministic timed Petri nets [BH07], often using designated time durations for transitions, and in some cases for places. The resulting models can then be analysed with regards to the time values and also different aspects like reachability and boundedness.

Currently the most common P/T net variant using a time notion are timed coloured Petri nets, which are similar to AHL nets, but use ML-data types and -expressions to control the firing behaviour of the underlying P/T net of a CPN model. The timed CPN extension allows the definition of time durations for transitions and modifies the firing behaviour accordingly.

The aforementioned P/T net variants however do not include ways to establish relations between different nets, therefore it is not possible to apply rule-based graph transformation and structuring techniques or specify processes for a given timed net, which is possible with algebraic P/T nets.

1.1 Aims

The main goal of this paper is the algebraic definition of timed P/T (or TPT) Nets to incorporate the previously mentioned advantages of algebraic P/T nets, including net structure and firing behaviour. The definition is based on algebraic P/T nets, enhancing them in order to include time durations as well as tokens with timestamps, while staying as close to the regular P/T net structure and firing behaviour as possible. Two case studies serve as examples on how the formalism can be used to model time-dependent systems and include certain conditions regarding their behaviour in the timed models.

We also define categories for different classes of timed nets. In category theory, a category is comprised of a class of objects and a set of morphisms between these objects that fulfill two basic properties, namely the associativity of morphisms and the existence of identity morphisms for each object. This very unrestrictive definition allows the objects and morphisms to be entities of an arbitrary kind, and thus category theory is a way to describe mathematical constructs in an abstract way. This in turn allows to extend properties that are proven to be true for categories in general to any mathematical structure as long as it can be defined as a category.

Defining categories of timed nets enables us to inherit certain properties, for example allowing the definition of structuring techniques (like union and fusion, which are already defined for algebraic P/T nets and are based on the categorical pushout construction), while morphisms allow us to specify processes for timed P/T nets. A special class of categories, $\mathcal{M}$-adhesive categories (with a class $\mathcal{M}$ of monomorphisms, fulfilling certain properties) allows the formalisation of rule-based transformation using the double pushout (DPO) approach.

The categories we define in this paper include those of timed P/T nets as well as timed P/T-systems and states, which are comprised of a timed P/T-system and a global clock value. Afterwards, we define functors between these categories, as well as functors that map timed P/T nets onto standard P/T nets. We also show that the categories of timed P/T systems and timed P/T states are equivalent, in the way that for every timed P/T state, modelling a system's state with absolute time, there is a corresponding timed P/T system with the same expressiveness, where the time is only modelled relatively. Vice versa, for a timed P/T system modelling relative time, we obtain a corresponding timed P/T state by adding a concrete clock value.

Moreover, we establish definitions for the structuring techniques union, fusion and restriction of timed P/T nets, analogous to the corresponding constructions for P/T nets. Union allows the construction of nets by gluing two nets together at so-called gluing points, while fusion glues together components of one net.

Finally, we outline how to extend the timed P/T net definition to timed AHL-nets, and briefly compare timed AHL-nets with timed CPNs. In the following, we provide an overview of the different sections.

1.2 Structure of the Paper

In Section 2, we take a look at different approaches to implementing a notion of time in P/T nets or their variants. We provide a short overview on their main features and differences, as well as some of the effects that the design decisions have with respect to the models.

In Section 3, we introduce two case studies, including a model of a computer network as well as an example representing a factory production line. These examples then serve as motivation for the definition of timed P/T nets and are also used as running examples to illustrate these definitions. We first show the models as regular P/T nets, which are extended to timed P/T nets in the following sections in order to fulfil special requirements we impose on the models.

In Section 4, we provide a formal overview of P/T nets and P/T systems, including the firing behaviour, P/T-net morphisms, and the structuring techniques union and fusion, as well as the categories of P/T nets and P/T systems.

In Section 5, we formally define the notion of timed P/T nets together with a firing behaviour based on timed markings, selections and states. Afterwards, we apply the newly defined timed P/T net approach to the case studies from Section 3, extending the original models and then simulating them as an example for the application of timed P/T nets.

In Section 6, we define the categories of timed P/T nets, timed P/T systems and timed P/T states, analogously to the categories of P/T nets and P/T systems. Moreover, we show that the morphisms of all three net classes preserve firing behaviour. We also define functorial relations between the timed P/T systems and states, as well as skeleton functors which translate timed P/T nets to P/T nets and timed P/T systems to P/T systems, respectively. We then show that these skeleton functors preserve firing behaviour. We also show that the categories of timed P/T systems and timed P/T states are equivalent, showing that they are essentially the same.

In Section 7, we define structuring techniques union, fusion and restriction for timed P/T nets analogously to those for algebraic P/T nets. Moreover, we show that the category of timed P/T nets fits into the abstract categorical framework of $\mathcal{M}$-adhesive categories which means that our approach is suitable for rule-based transformation of timed P/T nets in the sense of graph transformation.

The conclusion in Section 8 provides an overview on the main subjects of the paper, as well as the main results. We also give a short outline of how the notion of timed P/T nets can be extended to timed AHL-nets, and conduct a brief comparison of the timed P/T models shown in this paper with their timed CPN counterparts.

2 Related Work

In the past, there have already been a number of different approaches on how to introduce a notion of time to various flavours of Petri nets. While some of these are largely different from one another in the way they are integrated into the respective formalisation, they all share the common purpose of implementing a way to describe, design, and analyse models of time-dependent systems or processes.

In this section, we take a look at a selection of works in this area and roughly compare the methods used in the respective approaches.

We also briefly discuss the approach of P/T nets with individual tokens (PTI Nets), as the idea of timed selections in our approach of timed P/T nets is based on the concept of selections in the PTI approach.

2.1 Coloured Petri Nets

Coloured Petri nets (CPNs) were first introduced by Kurt Jensen in [Jen97] and described in detail by Jensen and Kristensen in [JK09]. In CPNs, a type ("colourset") is assigned to each place, allowing only tokens with values ("colours") of that specific type (or colourset) on each place. Expressions for edge inscriptions and transition conditions are denoted in ML-Syntax. The data types used are ML data types.

There is also a timed CPN extension, which assigns a duration to a transition (or single edge) and so called timestamps for each token, indicating the earliest point in time when a token can be used for a transition. A transition that fires adds the duration of the transition to the created tokens' timestamps, so in general, they can not be used immediately, but rather after the time the transition takes has passed. In a timed CPN model, not every place has to be timed (i.e. the tokens on this place do not possess timestamps), and the set of places of a timed CPN can contain both timed and untimed places.

There is also a tool provided for modelling timed CPNs, called "cpntools", which provides ways to design and analyse coloured Petri nets, including state-space analysis and model checking techniques (as described in [JKW07]), allowing for in-depth analysis and verification of net behaviour.

The timed CPN firing behaviour requires that a transition fires at the earliest point in time at which it is activated. This is a limitation that is in place to obtain a definite firing behaviour (although in the case of a conflict, one of the activated transitions has to be chosen at random). Therefore, during simulation, the global clock is monotonically increasing, as there is no possibility of a transition being activated at a point in time that has already passed.

2.2 Other Tools

Other notable tools for modeling variants of timed Petri nets include ROMEO, TINA and ORIS, which all employ different methods of analysis, but generally employ state-space analysis as well as model checking using different types of tree logic (LTL,CTL,TCTL).

A comparison of these tools can be found in [GLMR05].

2.3 Time Petri Nets

Time Petri nets (TPNs), introduced by P. Merlin in 1974 and described by Berthomieu and Diaz in [BD91], assign two labels to each transition, denoting the time that has to pass before that transition can fire after being enabled (EFT, earliest firing time), and the maximum time the transition can be enabled until it has to fire (LFT, latest firing time).

This firing behaviour is significantly different from that of timed coloured Petri nets, allowing for much more refined models with more control over the behaviour of the models.

Berthomieu and Diaz also describe means to analyse Time Petri Nets, using a state-space approach, while proving that the reachability and boundedness problems for TPNs are undecidable ([BD91]).

2.4 Deterministic Timed Petri Nets

Deterministic timed Petri nets, introduced by B. Hruz and M.C. Zhou in [BH07], pursue a rather unique approach for firing behaviour, actually introducing a delay between removal of tokens upon firing of a transition and the creation of tokens on the output place. In addition, each place has a designated delay, denoting the time before a created token can be consumed from that place.

Deterministic timed Petri nets are based on timed marked graphs, with marked graphs being a subclass of Petri nets, where each place of a marked graph has exactly one input edge and one output edge, as described in [CCCS92].

2.5 PTI Nets

Petri Nets with individual tokens, first introduced in [MGE$^+$10, MGH11], describe a formalism for Petri nets with tokens that are distinguishable from one another. Moving away from the collective token approach, where the tokens in a marking are simply denoted by a sum with no way to select a certain token from that sum, PTI nets use a set of specific tokens that are mapped (via a function) onto the respective places they are located on.

Since in the indivual token approach the tokens are unique, there has to be an indication of which tokens are consumed when firing a transition. This is done by choosing a selection of tokens (which is contained in the current marking), under which the respective transition is activated.

Note that in this paper we do not pursue an individual token approach in our definition of timed P/T nets even though we use certain aspects from the PTI formalism.

The algebraic approach presented in this paper allows us to formalise relations between timed nets via morphisms, allowing e. g. to specify a process of a timed P/T net, apply structuring techniques such as union and fusion to timed P/T nets and also define categories of different timed P/T net classes. Furthermore, we aim for a more liberal approach to activation and firing behaviour, being as unrestrictive as possible.

3 Case Studies

This section contains two case studies serving as motivation for the definition of timed P/T nets and as running examples to illustrate the definitions. The first case study is a model of a computer network, similar to a token ring network, while the second example is a model of a part of a production process. We introduce the two case studies as P/T nets first, then outline the required extension to model them as timed P/T nets.

3.1 Network Infrastructure

The following example models a (computer) network with several nodes (clients), which are connected to routers/switches. The client computers send data to the routers, which forward these packets among each other until the data can be sent directly to the target client. Figure 1 shows a sketch of this network.

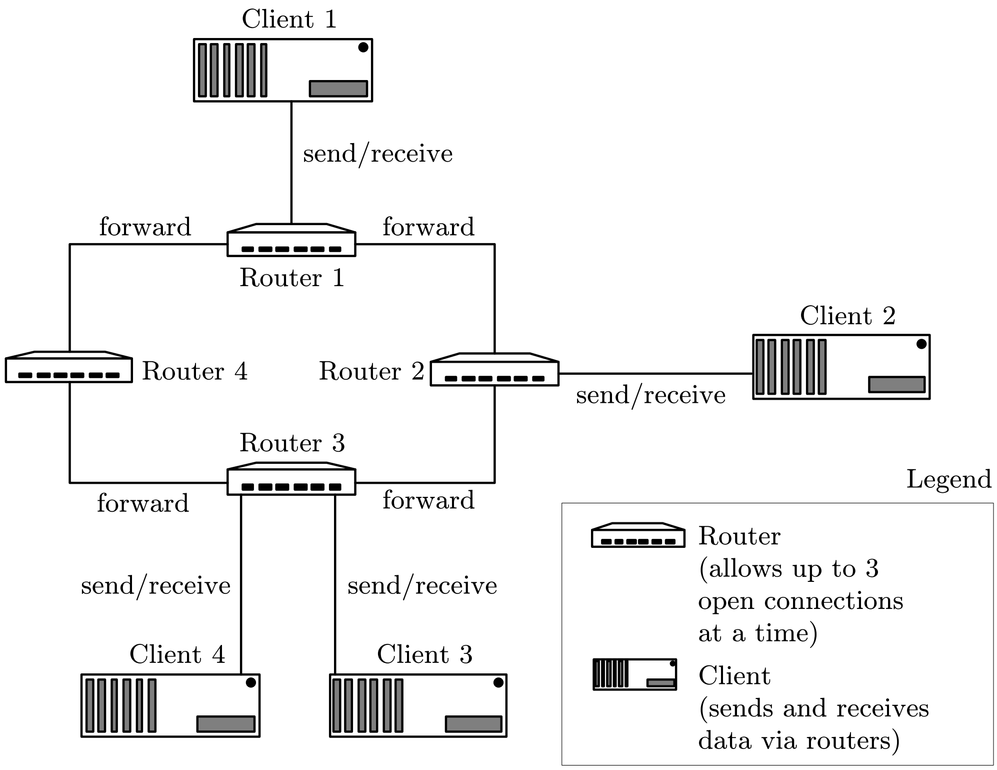


Figure 1: Network infrastructure

For the representation as a P/T net (Figure 2), we use one place for each client and router, which are connected via transitions, representing the sending and receiving of data from/to the clients, as well as the forwarding of packets between routers. The transitions designated rcv1 through rcv4 are used for transferring data from a switch to a client, while send1 through send4 model the transfer of data from a client to a switch. The switches are interconnected via forwarding transitions (fwd1 through fwd4), used to forward data between them.[1] Note that forwarding is only possible in one direction (clockwise in the

[1]Usually like in the network model, there is an intuitive interpretation of a transition's firing behaviour that it passes a token from an incoming arrow to an outgoing arrow. Thus, we may say that a particular token is passed through the net, although formally there is no implicit relation between consumed and produced tokens in a firing step.

figure). Additionally, each router has a “ready place”, which holds tokens that are consumed and immediately produced again whenever a forward occurs. The number of tokens on one of these places represents the maximum number of concurrent connections the router is able to maintain. Due to the instant production of the tokens upon firing of a forwarding transition, the number of tokens on the ready places has no effect in a regular P/T net, even if many transactions occur at “the same time”.

Therefore, we need a way to express the duration such a transaction takes, which is possible in timed P/T nets. This then allows the assigned durations to have a (restricting) effect on the firing sequences possible in the simulation of the net, which yields a behaviour that is more faithful to that of a real-world network.

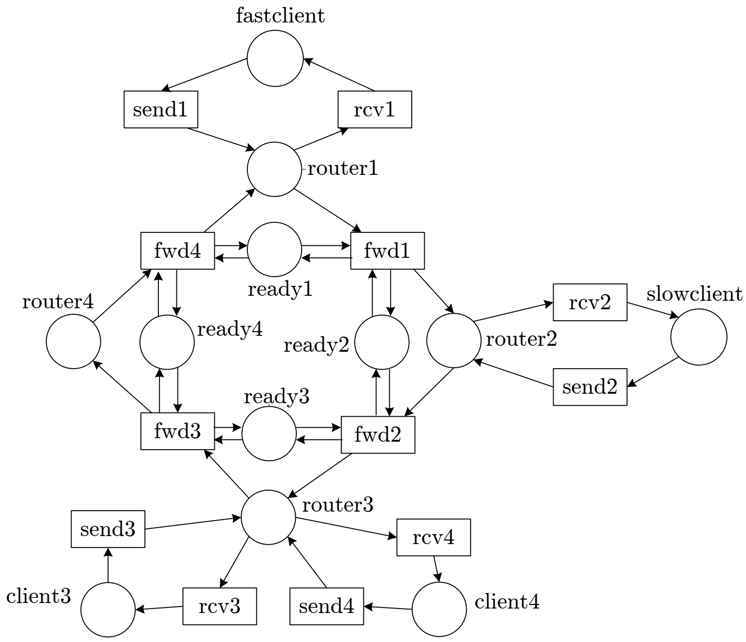


Figure 2: Network infrastructure a P/T net

Figure 3 shows a possible state of the network’s infrastructure as a P/T system (i.e. a net with an initial marking), where each ready-place contains one token, and additionally the fastclient place contains one token, while the slowclient place contains two tokens.

To illustrate the goal we want to achieve with timed P/T nets, Figure 4 shows the same network with time durations assigned to the output edges of the transitions. Note that these are not to be mistaken for the number of tokens created, but are still referring to single tokens that should be created with that specific delay. If multiple tokens were to be created by one output edge, this edge’s inscription would be a sum of time values, each addend corresponding to one token. A similar notation is used for timed nets in Section 5.

3.2 Production Line

Consider the simple production line example shown in Figure 5. The tokens on the Worker place represent workers who manufacture a product, which in turn is represented by tokens placed on the Product place. The manufacturing process is represented by the produce transition, which requires some utility, represented by the token on the Utility place. The

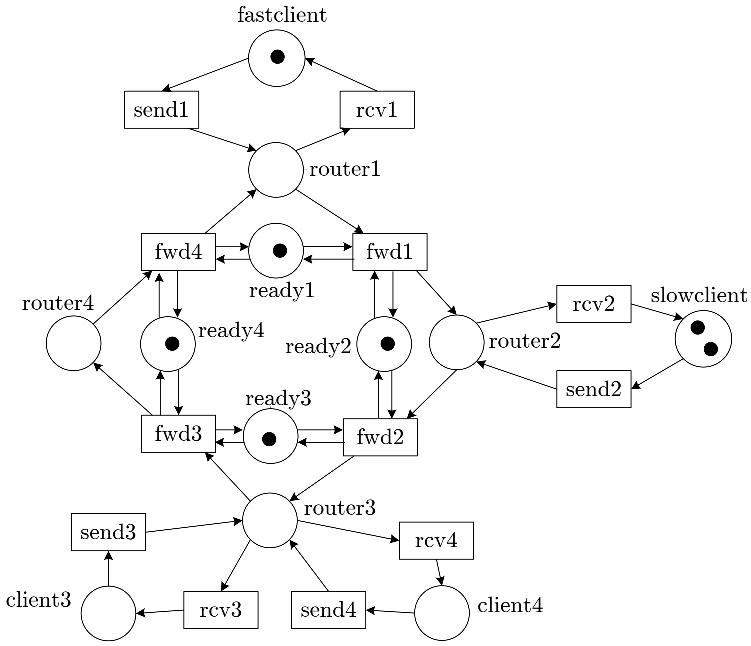


Figure 3: State of network infrastructure as P/T system

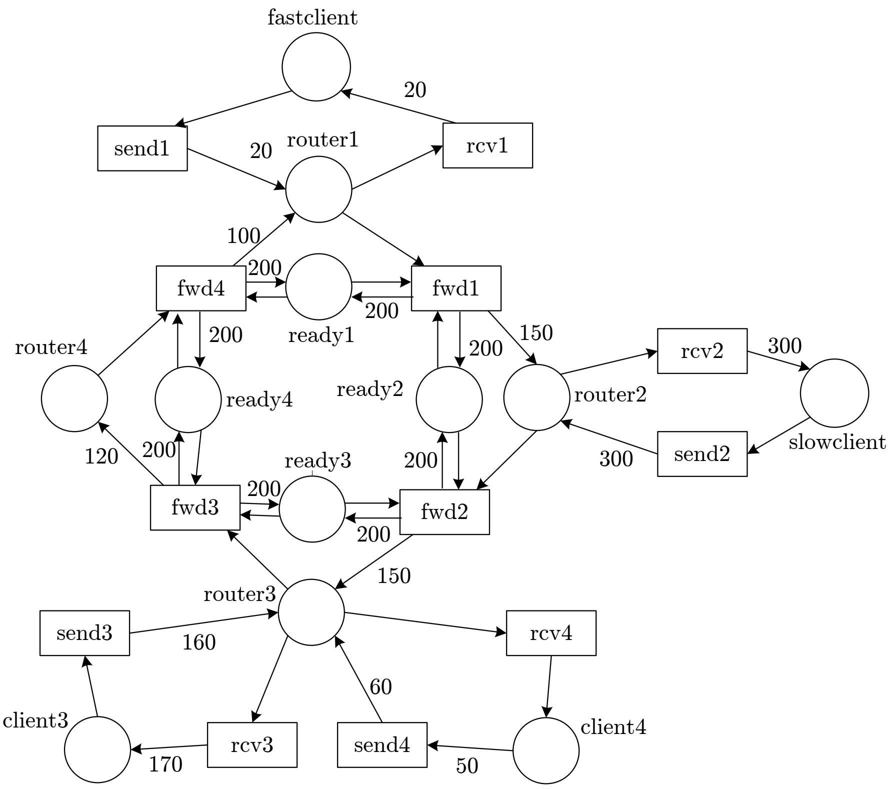


Figure 4: Network infrastructure as P/T net with durations on arcs

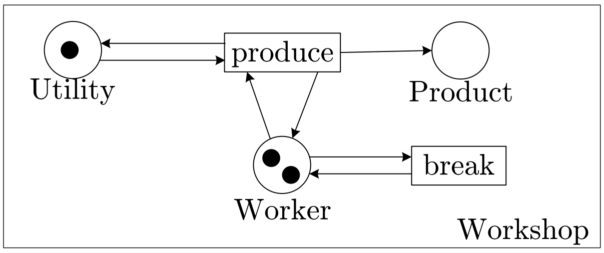


Figure 5: Workshop model as P/T Net

possibility of a worker taking a break is given by the break transition.

Note that there are two workers while the Utility place only holds one token. In the non-timed P/T net, however, the newly created utility token (from the produce transition) can be used without restrictions, so the availability of only one utility does not have an immediate effect on the firing behaviour. Similarly, the break transition does not have an actual effect, since it only removes one token from the Worker place and recreates a token on the same place.

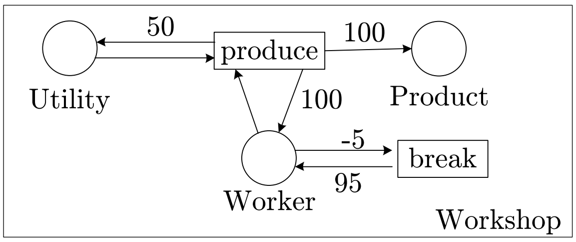


Figure 6: Workshop as timed P/T Net

Again, to illustrate how a timed P/T net of the production line could be modelled, Figure 6 shows the same net with time delays for the edges. Time values at the input edges change the earliest point in time at which the input tokens can be used. In this case, a value of -5 at the input edge of the *break* transition actually delays the time at which the worker tokens can be used by 5 time units. We discuss the details in Section 5, after formally defining the notion of timed P/T nets.

4 P/T Nets and Systems

This section contains an overview of algebraic P/T-nets in the sense of [MM90, ER97]. We review the formal definition of P/T nets, including net structure and firing behaviour, and also provide a short overview on the category of P/T nets. Afterwards, we review the notion of P/T systems and their category. We also briefly demonstrate the structuring techniques union and fusion.

Since we use the monoid notation for P/T nets, note that an element $s \in X^{\oplus}$ is a formal sum $s = \sum_{i=1}^{n} \lambda_i x_i$ with $\lambda_i \in \mathbb{N}, x_i \in X$ which implies that in s, there are λ_i occurrences of x_i. As for the addition, for another sum $s' = \sum_{i=1}^{n} \lambda'_i x_i$, we have $s \oplus s' = \sum_{i=1}^{n} (\lambda_i + \lambda'_i) x_i$.

4.1 P/T Nets

In the following segment, we provide a short overview of algebraic P/T nets.

P/T nets are based on sets of *places* and *transitions* that are connected by *arcs*. Arcs can only directly connect a place to a transition (input edges) or a transition to a place (output edges). The sum of places connected to a transition via an input arc is called the predomain of that transition, analogously the places connected via an output arc are called the post domain. When displaying a P/T net graphically, the usual conventions for graphical representations are circles for places, rectangles for transitions and arrows for arcs. Resources are represented by *tokens* (visualized by black dots) that are located on places, indicating the availability of resources to a transition (which represents some kind of action). A specific distribution of tokens on places is called a *marking*.

Arc inscriptions on the input arcs indicate how many tokens are needed on that arc's connected place in order to be able to "fire" the transition. Each input arc of the transition must have its token number requirement satisfied for it in order to fire.

Upon firing, the respective number of tokens inscribed on the input arcs of the firing transition are removed from the input places, then tokens are placed on the output places, with the number of tokens created on each output place matching the respective output arc's inscription.

Continuous exercising of this so called "token game" simulates the modelled system and allows for different types of analysis.

The following is a formal definition of P/T nets.

Definition 4.1 (P/T Net)**.** A *P/T Net* $N = (P, T, pre, post)$ consists of

- a set P of places,
- a set T of transitions, and
- functions $pre, post : T \to P^{\oplus}$ describing the pre- and post domain of each transition

A P/T net can be depicted as $T \underset{post}{\overset{pre}{\rightrightarrows}} P^{\oplus}$.

Next, we define a marking, which is essentially the assignment of a number of tokens to each place of a P/T net.

Definition 4.2 (Marking)**.** Given a P/T net $N = (P, T, pre, post)$. Then, a *marking* M is an element $M \in P^{\oplus}$.

Example 4.3 (Marking of P/T Nets). Figure 7 shows a P/T net $N_1 = (P, T, pre, post)$ with $P = \{p_1, p_2, p_3, p_4\}$ and $T = \{t_1, t_2\}$. As for the pre- and post domains of $t1$, we have $pre(t1) = 3p_1$, $post(t_1) = p2 \oplus 3p_3$ and for $t2$, $pre(t_2) = p_1$, $post(t_2) = p_4$.

The marking shown in the example is $M = 3p_1$, since there are three tokens on the place $p1$.

Another example is given by the network infrastructure P/T net in Figure 2, with places $P = \{fastclient, slowclient, client3, client4, router1, router2, router3, router4,$ $ready1, ready2, ready3, ready4\}$ and transitions $T = \{send1, rcv1, send2, rcv2, send3, rcv3, send4, rcv4, fwd1, fwd2, fwd3, fwd4\}$. A marking of this net is shown in Figure 3, with marking $M = fastclient \oplus ready1 \oplus ready2 \oplus ready3 \oplus ready4 \oplus 2slowclient$.

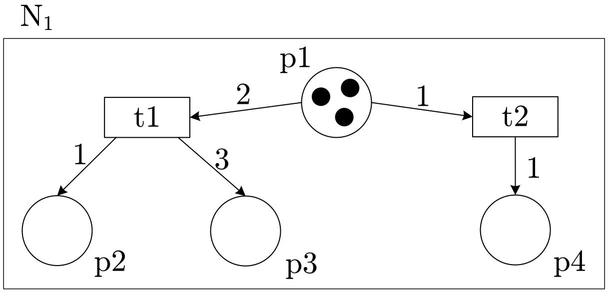


Figure 7: P/T net N1 and marking M

Next, we define the *firing behaviour* of a P/T net, which decides when a transition is activated, i.e. the conditions that have to be fulfilled so that a transition can fire. Upon firing, the input tokens (according to the predomain of the firing transition) are removed, and new tokens are placed on the output places of the transition.

Definition 4.4 (Activation, Firing Behaviour)**.** Let $N = (P, T, pre, post)$ be a P/T net and $M \in P^{\oplus}$ a marking of N.

- A transition $t \in T$ is *activated* under M, if $pre(t) \leq M$.
- A transition t that is activated under marking M can *fire*, written $M \xrightarrow{t} M'$, respectively $M\ [t\rangle\ M'$, leading to the follower marking M' with

$$M' = M \ominus pre(t) \oplus post(t).$$

Example 4.5 (Firing Step). In Figure 7, transition t_1 is activated, since $pre(t_1) = 2p_1$, $M = 3p_1$, and so $pre(t_1) \leq M$. Therefore, t_1 can fire, resulting in the follower marking M' shown in Figure 8. M' is calculated as

$$M' = M \ominus pre(t_1) \oplus post(t_1) = (3p_1) \ominus (2p_1) \oplus (p_2 \oplus 3p_3) = p_1 \oplus p_2 \oplus 3p_3.$$

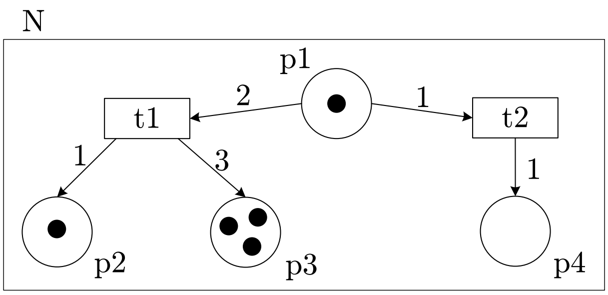


Figure 8: P/T net N after firing of t_1

4.2 Category of P/T Nets

Here, we define morphisms between P/T nets, and subsequently the category **PTNets**. The basics of category theory are covered in Appendix A.

First, we define the notion of P/T net morphisms, which are mappings from one P/T net onto another, defined componentwise on the sets of places and transitions, such that the pre- and post domains of all transitions are preserved.

Definition 4.6 (P/T Net Morphism)**.** Given P/T nets $N_i = (P_i, T_i, pre_i, post_i)$ with $i = 1, 2$. Then, a *P/T net morphism* $f : N_1 \to N_2 = (f_P, f_T)$ is a pair of mappings $f_P : P_1 \to P_2, f_T : T_1 \to T_2$, such that the following diagram commutates componentwise for pre and post:

$$\begin{array}{ccc} T_1 & \overset{pre_1}{\underset{post_1}{\rightrightarrows}} & P_1^{\oplus} \\ {\scriptstyle f_T}\downarrow & & \downarrow{\scriptstyle f_P^{\oplus}} \\ T_2 & \overset{pre_2}{\underset{post_2}{\rightrightarrows}} & P_2^{\oplus} \end{array}$$

Example 4.7 (P/T Net Morphism). Figure 9 shows P/T nets N_1 and N_2 with P/T-net morphism $f : N_1 \to N_2$ with $f_P(p_1) = (p_1), f_P(p_2) = f_P(p_3) = p_{23}$ and $f_T(t_1) = (t_1)$.
The morphism condition is fulfilled, because $post_2(f_T(t_1)) = post_2(t_1) = 2p_{23}$, $f_P^{\oplus}(post_1(t_1)) = f_P^{\oplus}(p_2 \oplus p_3) = p_{23} \oplus p_{23} = 2p_{23}$, and thus, $post_2(f_T(t_1)) = f_P^{\oplus}(post_1(t_1))$. Analogously, we have $pre_2(f_T(t_1)) = f_P^{\oplus}(pre_1(t_1))$.

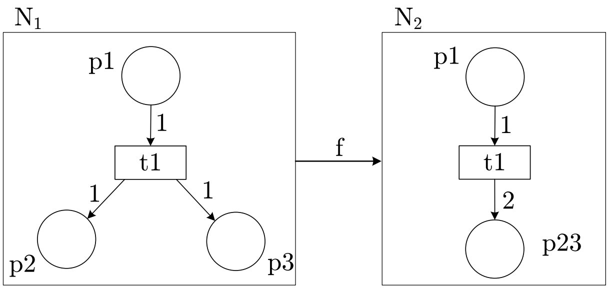


Figure 9: P/T-net morphism

Fact 4.8 (P/T Met Morphisms Preserve Firing Behaviour)**.** Given P/T nets $N_i = (P_i, T_i, pre_i, post_i)$ with $i = 1, 2$, a marking $M \in P_1^{\oplus}$ of N_1 and a P/T net morphism $f : N_1 \to N_2$. Let $t \in T_1$ be a transition in N_1 which is activated under M. The for every firing step $M \xrightarrow{t} M'$ in N_1 there is a corresponding firing step $f_P^{\oplus}(M) \xrightarrow{f_T(t)} f_P^{\oplus}(M')$ in N_2.

Definition 4.9 (Category **PTNets** of P/T Nets)**.** The class of all P/T nets along with P/T net morphisms constitute the category **PTNets**.

The identities and composition are defined componentwise as identities and composition, respectively, of places and transitions in the category **Sets** of sets and functions.

4.3 Category of P/T Systems

A P/T system is a tuple containing a P/T net along with an (initial) marking.

Definition 4.10 (P/T System)**.** A *P/T system* or marked P/T net

$$S = (N, M)$$

is a P/T net $N = (P, T, pre, post)$ with (initial) marking $M \in P^{\oplus}$.

Definition 4.11 (P/T System Morphism)**.** Given P/T systems $S_i = (N_i, M_i)$ with $N = (P_i, T_i, pre_i, post_i)$ and $M_i \in P_i^{\oplus}$ for $i = 1, 2$. Then, a *P/T-system morphism* is a P/T-net morphism $f = (f_P, f_T)$ that fulfils the following condition:

$$\forall p \in P_1 : M_1(p) \leq M_2(f_P(p))$$

The P/T-system morphism f is *marking-strict*, if the following condition is fulfilled:

$$\forall p \in P_1 : M_1(p) = M_2(f_P(p))$$

Example 4.12 (P/T System Morphism). Figure 10 shows a P/T-system morphism $f : (N_1, M_1) \to (N_2, M_2)$ with markings M_1 of N_1 and M_2 of N_2. The P/T-net morphism condition is fulfilled as shown in the previous example.

As for the P/T-system morphism condition, we have $M_1 = p_1 \oplus p_2$ and $M_2 = 2p_1 \oplus 2p_{23}$. So, since $M_1(p_1) = 1 \leq 2 = M_2(f_P(p_1))$ and $M_1(p_2) = 1 \leq 2 = M_2(f_P(p_2))$, we have $M_1(p) \leq M_2(f_P(p))$ for all $p \in P_1$. Therefore, f is a (non-marking-strict) P/T-system morphism.

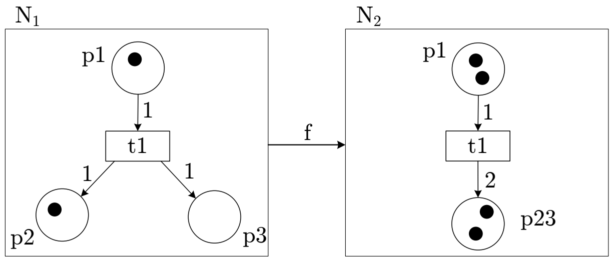


Figure 10: P/T-system morphism

Definition 4.13 (Category **PTSys** of P/T Systems)**.** The class of all P/T systems, along with P/T-system morphisms, constitute the category **PTSys**. The composition of two P/T system morphisms is defined as the composition of the corresponding P/T-net morphisms.

4.4 Structuring Techniques

In this subsection, we review two structuring techniques for P/T nets: union and fusion. The union of two nets N_1 and N_2 over an interface N_0 results in a new net N_3, containing N_1 and N_2, which are "glued" together at their common components in N_0. union and fusion are based on the categorical notions of pushout and coequaliser, which are covered in Appendix A.

Definition 4.14 (Union of P/T Nets)**.** Given P/T nets $N_i = (P_i, T_i, pre_i, post_i)$ for $i = 0, 1, 2$ with P/T net morphisms $f_1 : N_0 \to N_1$ and $f_2 : N_1 \to N_2$.

$$\begin{array}{ccc} N_0 & \xrightarrow{f_1} & N_1 \\ {\scriptstyle f_2}\downarrow & (1) & \downarrow{\scriptstyle g_1} \\ N_2 & \xrightarrow[g_2]{} & N_3 \end{array}$$

Then, the union object $N_3 = (P_3, T_3, pre_3, post_3)$ is constructed componentwise as pushouts in **Sets** for the sets of places (P_3) and transitions (T_3). pre_3 and $post_3$ are induced by the pushout construction.

Example 4.15 (Union of P/T Nets). Figure 11 shows the union of P/T nets N_1, N_2 with N_0 as the interface. The morphism f_1 maps the places and transitions according to their labels, while the mapping of f_2 is non-injective, since $f_{2P}(p1) = f_{2P}(p2) = p1, 2$.

For g_1 and g_2, we have:

- $g_{1P}(p1) = g_{1P}(p2) = p1, 2$, $g_{1P}(p4) = p4$, $g_{1T}(t1) = t1$, $g_{1T}(t5) = t5$,
- $g_{2P}(p1, 2) = p1, 2$, $g_{2P}(p3) = p3$, $g_{2T}(t1) = t1$, $g_{2T}(t3) = t3$.

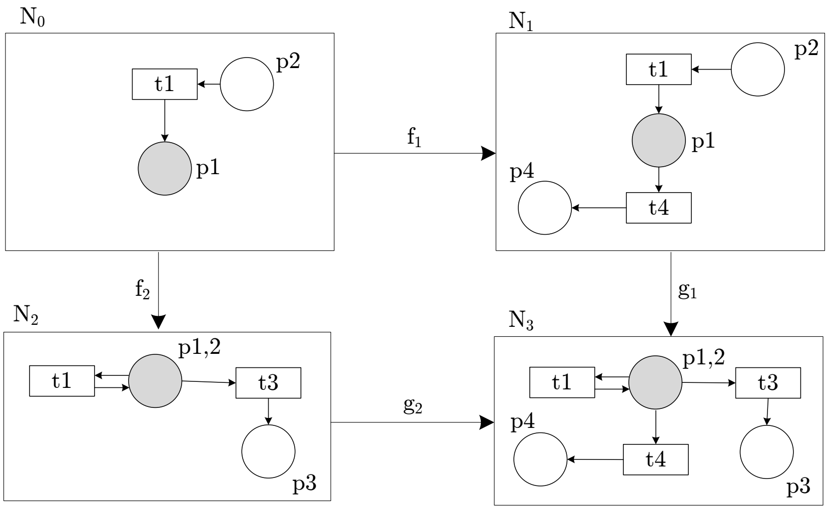


Figure 11: Union of P/T Nets

Remark 4.16 (Union is Pushout). Given a union object N_3 of P/T nets N_1, N_2 with N_0 as the interface. Then, N_3 is the pushout of N_1, N_2 with N_0 as the interface.

Definition 4.17 (Fusion of P/T Nets)**.** Given P/T nets $N_1 = (P_i, T_i, pre_i, post_i)$ for $i = 1, 2$ with P/T net morphisms $f, g : N_1 \to N_2$.

$$N_1 \overset{f}{\underset{g}{\rightrightarrows}} N_2 \xrightarrow{c} N_3$$

Then, the fusion object $N_3 = (P_3, T_3, pre_3, post_3)$ with morphism $c : N_2 \to N_3$ is constructed componentwise as coequalisers in **Sets** for the sets of places (P_4) and transitions (T_4). pre_3 and $post_3$ are induced by the coequaliser construction.

Example 4.18 (Fusion of P/T Nets). Figure 12 shows the fusion of P/T nets N_1, N_2 with $f_T(p_1) = p_1$, $g_T(p_1) = p'_1$, $f_T(p_2) = p_2$, $g_T(p_2) = p'_2$. In the resulting net N_3, the place p_1 is the identification of the places p_1, p'_1 from N_2.

Remark 4.19 (Fusion is Coequaliser). Given a fusion (N_3, c) of P/T nets $f, g : N_1 \to N_2$ then (N_3, c) is the coequaliser of $f, g : N_1 \to N_2$.

4.5 Processes of P/T Nets

The concept of processes in P/T nets is essential to model not only sequential, but especially concurrent firing behaviour. A process of a P/T net is given by an occurrence net K together with a P/T net morphism $p : K \to N$.

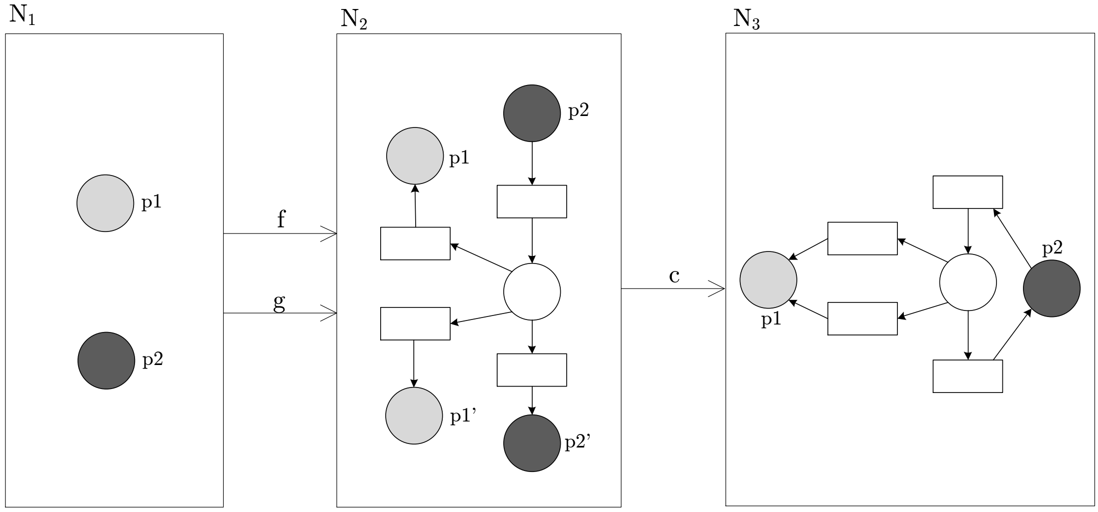


Figure 12: Fusion of P/T Nets

Definition 4.20 (Occurrence Net)**.** An *occurrence net* K is a P/T net $K = (P, T, pre, post)$ such that for all $t \in T$ with $pre(t) = \sum_{i=1}^{n} p_i$ and notation $\bullet t = \{p_1, \ldots, p_n\}$ for the pre domain and similarly $t\bullet$ for the post domain, we have:

1. *(Unarity)* $\bullet t$ and $t\bullet$ are sets rather than multisets for all $t \in T$, i. e. for $\bullet t$ the places $p_1, \ldots, p_n$ are pairwise distinct,
2. *(No Forward Conflicts)* $\bullet t \cap \bullet t' = \emptyset$ for all $t, t' \in T, t \neq t'$,
3. *(No Backward Conflicts)* $t \bullet \cap t' \bullet = \emptyset$ for all $t, t' \in T, t \neq t'$, and
4. *(Strict Partial Order)* the causal relation $<_K \subseteq (P \uplus T) \times (T \uplus P)$ defined by the transitive closure of

$$\{(p, t) \in P \times T \mid p \in \bullet t\} \cup \{(t, p) \in T \times P \mid p \in t\bullet\}$$

   is a finitary strict partial order, i. e. the causal relation is irreflexive and for each element in the relation the set of its predecessors is finite.

Definition 4.21 (P/T Process)**.** A *P/T process* of a P/T net N is a P/T morphism $p : K \to N$ where K is an occurrence net.

Example 4.22 (P/T Process). Figure 13 shows a P/T net version *Workshop-Net* of our production line example from Section 3. The P/T morphism p : *Workshop-Proc* $\to$ *Workshop-Net*, mapping every place and every transition to the place respectively transition with the same name but without number, is a P/T process of the net *Workshop-Net*, because *Workshop-Proc* is an occurrence net.

The net *Workshop-Proc* is unary, because all arcs in the net have a weight of 1. The net has no backward or forward conflict, because all places are at most in the pre and post domain, respectively, of one single transition. Moreover, the causal relation is a finitary strict partial order, since the net is finite and does not contain any cycles.

The process models a scenario with two workers and one utility. The first worker starts the production and then takes a break before continuing to work on a second product. The second worker uses the free utility at some point between the two productions of the first workers which can be before, after and/or while the first worker has its break.

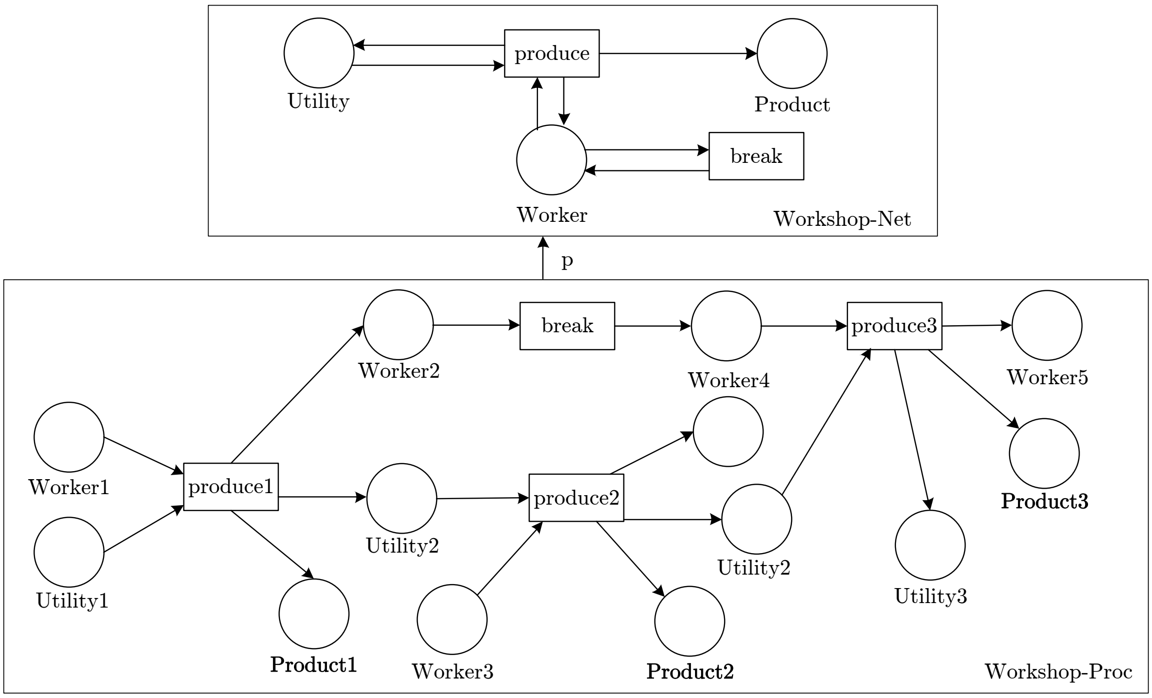


Figure 13: Production line process

5 Timed P/T Nets

In this section, we provide a formal definition of timed P/T nets, largely based on the monoidal definition of P/T nets. Beforehand, we outline the requirements towards a timed P/T net framework, examining different options for certain aspects and explain the reasoning behind the decisions made in the formalisation.

As for the formal definitions, we first define the net structure and timed markings, selection and states. Based on this, we define the firing behaviour of timed P/T nets including activation, firing steps and firing sequences.

Finally, we apply the definitions to the case studies presented in Section 3, showing the simulation of the timed P/T nets representing the network and production line models.

5.1 Requirements

In this segment, we establish a series of requirements for a formalism for timed P/T nets. This is a comprehensive list of features one would expect from such a construct, along with different possible ways of designing each aspect.

Timed P/T nets are intended to be used to model and analyse time-dependent processes, or in general systems that need to be able to react or finish their execution inside a specific time constraint (so called real-time systems). Another use would be analysis and optimisation of a specific system or process with regard to the time that has passed. Therefore, we need to be able to keep track of the time while simulating a model, as well as to have available a way to assign a duration to each action (represented by transitions in the case of P/T nets), which in turn needs to have an effect on the firing behaviour of the timed P/T net, so that the passing of time actually has an effect on how the net behaves.

First, we discuss model time. This refers to the way time values are represented in the model, and how a global clock can be implemented, in order to be able to tell when events can occur or occur during the simulation. Next, a way of representing the duration of actions (or in this case, transitions) needs to be found. Finally, we need to determine a way to ensure that the duration of a transition actually has an effect on the firing behaviour of the net, i. e. only allowing transitions to fire after the duration of another transition has passed if it is directly dependent on tokens created by that transition.

5.1.1 Model Time

Before starting to remodel transitions and the firing behaviour, we need to decide on a data type to represent time durations and instants of time in general. Basically, this is a decision between having a discrete or continuous time model.

Discrete time means having a finite number of time steps between any two points in time. There would not be a way to insert a time step in between two directly consecutive points in time. The benefit of this approach would be simplicity in both modelling and simulating nets.

However, if one would like to refine an action (transition) that takes one time unit, all the durations in the model would have to be upscaled in order to allow for a more detailed model.

Continuous time, in contrast, has an infinite number of time values between any two values. This allows for later refinement of nets. In any case, it is less restrictive than discrete time and ultimately allows the modeller to choose the level of detail they wish to apply to their model.

Therefore, for the timed P/T net formalism, we use the set of real numbers $\mathbb{R}$ as the data type for time values. Since the natural numbers are included in the set of real numbers, the modeller is still free to only use those if they wish. For simpler models, the set of natural numbers is sufficient and also the most intuitive way for representing the time values.

The timed P/T net formalism employs a global clock, which is a time value representing the current model time. This clock is the basis for the decision whether or not a transition can use the tokens on its input place, or if firing is only possible at a later global clock value.

5.1.2 Time Duration

A timed P/T net formalism needs to include a way to express the duration that actions in the modelled system take. Since actions in a P/T net are represented by transitions, each transition gets assigned durations for each incoming and outgoing edge.

While a single duration for each transition might be sufficient for many (simpler) applications, there may be cases in which some results of an action might be available before another one is ready. As an example, consider a production facility, where one side-product of a production step is ready at an earlier point in time than the actual final/main product.

Since a transition can consume and produce more than one token per connected place, each output edge gets assigned a sum of time values, each sum element representing one token's time it takes until it is available after the transition has fired.

While time durations for outgoing edges might be easily understood as the time it takes for that token to be "created" by the transition, time durations on input edges are more complex and not as intuitively understandable. As we show later in this section, durations on the input edges allow earlier (or demand later) consumption of a token than would otherwise be allowed by the current global clock value.

5.1.3 Marking

We define a marking of a timed P/T net of one place, just like the edge inscriptions, as a sum of timestamps located on that particular place. And, consequently, a marking of the whole net is a sum of pairs of places and time values.

5.1.4 Firing Behaviour

The definitions of time durations in a timed P/T net take effect in the firing behaviour. Perhaps the simplest approach would be just adding up all transition durations without changing the firing behaviour. This, however, does not allow a detailed analysis (and simulation) of processes and is therefore insufficient.

A more complex, but also intuitive, way is to delay token creation corresponding to the time durations assigned to the transitions. However, this would mean keeping track of every token that needs to be created, for example in the form of local clocks for each token currently "in creation", which would make firing behaviour and therefore simulation of a net very complex.

The approach we introduce here is based on that used in coloured Petri nets (see [JK09]), which assigns a so called timestamp (time value) to each token upon creation (by firing a transition). This timestamp represents the earliest point in time at which this token can be consumed by a transition, so that it will usually be assigned a later time stamp value than the current model time, the time difference being this transition's edge's duration.

Timestamps have to be included in the definition of the activation of transitions, checking whether the current time has advanced enough in order to consume all the required input tokens. This method also covers the approach where token creation is delayed, and is generally a more feasible approach regarding the definition of the firing behaviour.

5.1.5 Net Structure

Timed P/T Nets extend the notion of P/T nets by introducing time durations for edge inscriptions, as well as a global clock. Tokens are being represented as their respective time stamps, which indicate at what time a particular token can be used again in order to fire a transition. A marking of a place is represented as a sum of time values, where one value indicates when that token can be consumed by a transition.

In the following, we define timed P/T nets based on the definition of regular P/T nets, with one set each for places and transitions, as well as functions pre and post, which map a sum over the Cartesian product of places and time values to a transition, defining the edges of the net with their respective durations.

Each tuple of a time value and place denotes one token that is created or consumed, specifying the place it is created on or taken from, as well as the time offset until it becomes available (after production) or the amount of time a token can be removed early from an input place (as seen later when defining the firing behaviour).

Apart from this, the definition is similar to that of standard P/T nets.

Definition 5.1 (Timed P/T Net)**.** A *timed P/T net* or TPT net $TN = (P, T, pre, post)$ consists of

- a set P of places,
- a set T of transitions, and
- functions $pre, post : T \to (P \times \mathbb{R})^{\oplus}$

Remark 5.2 (Note on Graphical Representation). Graphical representations of timed P/T nets are similar to those of untimed P/T nets, with some alterations. Places and transitions are still depicted by circles and rectangles respectively and connected by arrows which represent the edges.

Edge inscriptions are now sums ($\oplus$) of time values, which means that the number of tokens produced by an outgoing edge of a transition is now the number of addends in that edge's inscription. So for example an edge inscribed with a single zero (which is not meaningful for untimed P/T nets) means that upon firing of the connected transition, one token with no time offset is produced on the target place. Analogously to classic P/T nets, an edge with no inscription means that there is a single token created or consumed with no time delay, i.e. an empty edge is equivalent to an edge with a single zero.

Tokens, instead of being information-devoid objects, now carry a timestamp (which, as discussed earlier, is the earliest point in time at which the token can be used), so the tokens are now represented by numbers instead of black dots inside the places. This is not to be confused with actual data represented by tokens for example in AHL-nets, but rather an additional and independent type of information.

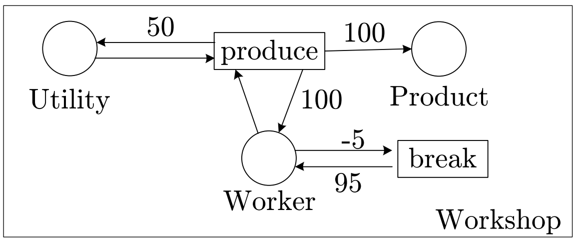


Figure 14: Workshop as timed P/T net

Example 5.3 (Timed P/T Net). Figure 14 shows the timed P/T net $Workshop = (P, T, pre, post)$, taken from the production line case study with

- $P = \{Utility, Product, Worker\}$,
- $T = \{produce, break\}$,
- $pre(produce) = (Utility, 0) \oplus (Worker, 0)$
- $pre(break) = (Worker, -5)$,

- $post(produce) = (Utility, 50) \oplus (Product, 100) \oplus (Worker, 100)$,
- $post(break) = (Worker, 95)$.

5.2 Firing Behaviour

Now we define the firing behaviour of timed P/T nets. For this purpose we introduce timed markings and selections of these markings, which are then used to define under which conditions a transition is activated.

5.2.1 Timed Marking, Selection and State

For TPT nets, we define a timed marking which represents the distribution of tokens on the places with their respective timestamps. Analogously to markings in P/T nets, we define markings as an element of the commutative free monoid $(P \times \mathbb{R})^{\oplus}$.

Definition 5.4 (Timed Marking)**.** A *timed marking* of a TPT net $TN = (P, T, pre, post)$ is an element $M \in (P \times \mathbb{R})^{\oplus}$.

Remark 5.5 (Representation of Timed Marking)*.*

1. An untimed marking M can be written either in the form $\sum_{i=1}^{n} p_i$ or in the form $\sum_{i=1}^{n} \lambda_i p_i$. In the first form for a place p, an index i with $p_i = p$ represents one token on place p. In the second (shorter) form we usually have $n = |P|$, and $\lambda_i p_i$ means that there are $\lambda_i \in \mathbb{N}$ tokens on place p_i.

   Analogously, a timed marking M can be written either in the form $\sum_{i=1}^{n}(p_i, r_i)$ or in the form $\sum_{i=1}^{n}\sum_{j=1}^{n_i}(p_i, r_i^j)$. In the first form for a place p and index i with $p_i = p$ we have a token with time-value r_i on place p. In the second form we have a sum of time-values $\sum_{j=1}^{n_i}(p_i, r_i^j)$ on place $p_i \in P$.

2. Based on the short form for untimed markings, an untimed marking $M = \sum_{i=1}^{n} \lambda_i p_i$ can also be represented as a function $M : P \to \mathbb{N}$ with $M(p_i) = \lambda_i$. Analogously, a timed marking $M = \sum_{i=1}^{n}\sum_{j=1}^{n_i}(p_i, r_i^j)$ can be represented as a function $M : P \to \mathbb{R}^{\oplus}$ with $M(p_i) = \sum_{j=1}^{n_i} r_i^j$.

Example 5.6 (Timed Marking)*.* In Figure 15, the marking of the timed P/T net $Workshop$ is

$$M = (Utility, 60) \oplus (Worker, 25) \oplus (Worker, 110) \oplus (Product, 110).$$

This means that there is one token with timestamp 60 on the place $Utility$, two tokens on the $Worker$ place with the timestamps 25 and 110, respectively. The $Product$ place contains one token with the timestamp 110.

We define a timed state as a tuple containing a timed net with a timed marking and the current global clock value.

Definition 5.7 (Timed State)**.** A *timed state* TS is a 3-tuple

$$TS = (TN, M, \tau)$$

with timed P/T net TN, a marking M of TN and a global clock value $\tau \in \mathbb{R}$.

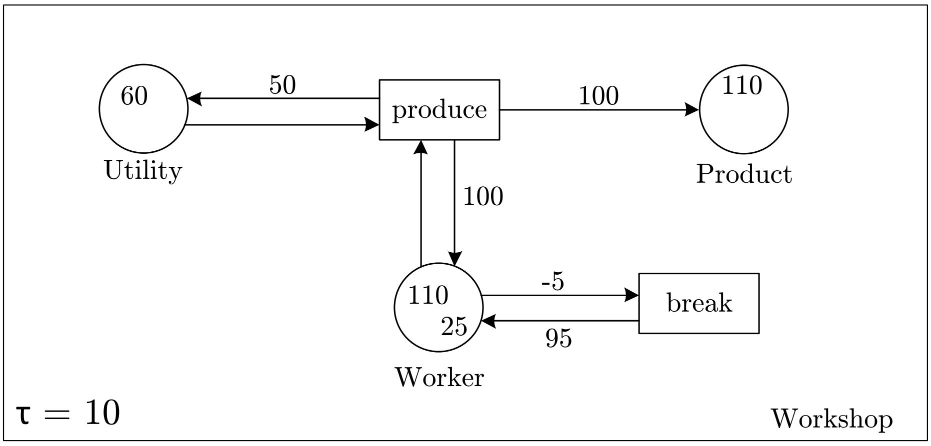


Figure 15: Workshop net with timed marking and global clock value

Note that if the net that is being referred to is apparent from the context, we will sometimes omit the net and call (M, τ) a timed state.

Example 5.8 (Timed State). We consider the the current timed state (TN, M, τ) as shown in Figure 15 with

$$M = (Utility, 60) \oplus (Product, 110) \oplus (Worker, 26) \oplus (Worker, 110) \text{ and } \tau = 60.$$

Since timed states contain a clock value, we need to define a way to change this clock value to retrieve a new timed state, a so-called time step. This is needed because firing a transition does *not* advance the global clock, thus allowing actions to overlap in time. Timesteps allow us to change the time to the desired clock value at which the next firing step is to take place.

Definition 5.9 (Timestep)**.** Given a timed state (TN, M, τ) with timed P/T net TN, a marking M of TN and a clock value $\tau \in \mathbb{R}$ as well as an arbitrary time difference $\Delta\tau \in \mathbb{R}$. Then, there is a *time step* resulting in the timed state $(TN, M, \tau + \Delta\tau)$, written

$$(TN, M, \tau) \xrightarrow{\Delta\tau} (TN, M, \tau + \Delta\tau).$$

Example 5.10 (Timestep). Given a timed state $(TN, M, 25)$ with timed P/T net TN, a marking M of TN and the global clock value of 25. In order to advance the clock by 15 time units, we apply the following time step: $(TN, M, 25) \xrightarrow{15} (TN, M, 25 + 15) = (TN, M, 40)$.

Since we only need to consider a marking's tokens in the immediate environment of the predomain of a transition in order to check if it is activated, we define a selection of tokens which is contained in that marking.

Remark 5.11 (Selections). We use an approach similar to selections in Petri nets with individual tokens [MGE+10] (see Section 2.5), where token selections are used for choosing which tokens are used for firing a transition. We do, however, not use selections for the follower markings, but instead provide a definition that is closer to the follower marking definition in the firing behaviour of algebraic P/T nets.

Definition 5.12 (Timed Selection)**.** Given a timed marking $M \in (P \times \mathbb{R})^{\oplus}$ of a TPT net $TN = (P, T, pre, post)$, a *timed selection of* M is a marking $S \leq M$.
We call $\pi_P^{\oplus}(S)$ the *location of* S, where $\pi_P^{\oplus}(\sum_{i=1}^{n}(p_i, r_i)) = \sum_{i=1}^{n} p_i$ is the projection that "forgets" the time-values.

Example 5.13 (Timed Selection). In Figure 15, a valid selection w. r. t. M is for instance $S = (Worker, 25) \oplus (Utility, 60)$, which has a location of $\pi_P^{\oplus}(S) = Worker \oplus Utility$.

Next, we define the firing behaviour of timed P/T nets. We begin by defining the conditions for the activation of a transition. In classic P/T nets, a transition is activated under a marking if there are enough tokens on the input places of that transition. In timed P/T nets, we also need to take into account the timestamps of the involved tokens and whether they are exceeded by the global clock in the net's current state. The time values at the input edges of the transitions are added to the global clock, which enables us to actually remove a token from a place and use it in a transition early (with the edge's time value indicating how much earlier the tokens can be used for the transition).

We define the time-sorted list of tokens for a specific place. This is a function that returns the tokens on a place for a given marking, represented as a list sorted by timestamps (ascending).

Definition 5.14 (Time-Sorted List)**.** Given a marking $M \in (P \times \mathbb{R})^{\oplus}$ of a TPT net $TN = (P, T, pre, post)$. Then for each place $p \in P$ the *time-sorted list w.r.t.* p is defined as

$$M[p] = [r_1, \ldots, r_n] \in \mathbb{R}^*$$

such that

$$M(p) = \sum_{i=1}^{n} r_i \text{ (see Remark 5.5) and for } 1 \leq i < j \leq n : r_i \leq r_j.$$

Example 5.15 (Time-Sorted List). In Figure 15, the time-sorted list of $Worker$ is $M[Worker] = [25, 110]$.

In order to check whether the global clock is "late" enough in order for a transition to fire, we need to be able to compare the timestamps in a marking to those of another marking. For this purpose, we define the notion of time-delays in the following sense:

Definition 5.16 (Time-Delay)**.** Given two timed markings $M_1, M_2 \in (P \times \mathbb{R})^{\oplus}$ of a TPT net $TN = (P, T, pre, post)$. We define the following two types of delays:

- M_1 is a *location-strict delay* of M_2, written $M_1 \overset{\leftarrow}{=} M_2$, if
  1. they have the same location, i. e. $\pi_P^{\oplus}(M_1) = \pi_P^{\oplus}(M_2)$, and
  2. for all $p \in P$: $M_1[p] \geq M_2[p]$.
- M_1 is a *delay* of M_2, written $M_1 \overset{\leftarrow}{\leq} M_2$, if there exists a marking $M_2' \leq M_2$, such that $M_1 \overset{\leftarrow}{=} M_2'$.

Note that we also use the notation $M_2 \overset{\rightarrow}{=} M_1$, which is equivalent to $M_1 \overset{\leftarrow}{=} M_2$. We call the timestamps in M_1 *later* than those in M_2.

Remark 5.17 (Time-Delay). When comparing two time-sorted lists, it does not make a difference whether standard or lexicographical ordering is used, as the lengths of the compared lists are always identical.

The intention behind the symbols chosen for delays is as follows: The bottom comparator indicates which marking is larger (or that they are of equal location) w. r. t. the number of tokens, while the arrow above points to the marking with the higher timestamps, i. e. in the direction of the timestamps which are *later*.

Moreover, note that if M_1 is a location-strict delay of M_2 then it is also a delay of M_2, and there exists only the subsum $M_2' = M_2$ with $M_1 \stackrel{\leftarrow}{=} M_2'$.

Example 5.18 (Time-Delay). Consider the marking $M_1 = (p_1, 3) \oplus (p_2, 2) \oplus (p_2, 5)$ of the timed P/T net TN shown in Figure 16. The marking $M_2 = (p_1, 4) \oplus (p_2, 3) \oplus (p_2, 8)$ is a location-strict delay of M_1, since the location of both markings is the same, and each timestamp in M_1 has a higher or equal timestamp in M_2.

The marking $M_3 = (p_1, 4) \oplus (p_2, 2)$ is a (non-location-strict) delay of M_1, since there is $M_1' = (p_1, 3) \oplus (p_2, 2) \leq M_1$ with $M_3 \stackrel{\leftarrow}{=} M_1'$.

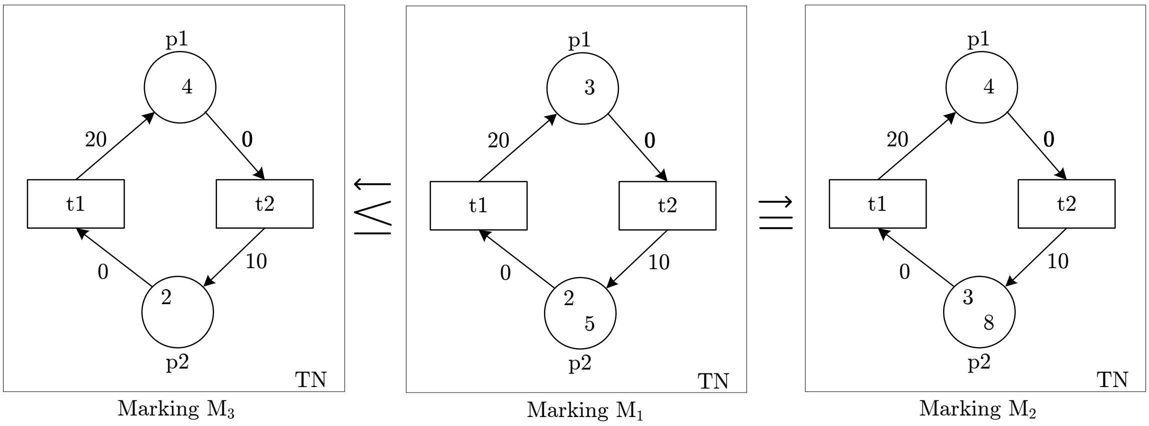


Figure 16: Timed P/T net TN with markings

Definition 5.19 (Maximal Timed Selection)**.** Given a timed marking M and a selection $S \leq M$. We call S a *maximal timed selection of* M, if for all selections $S' \leq M$ with the same location as S, the selection S is a delay of S', i. e. $S \stackrel{\leftarrow}{=} S'$.

Next, we define a way to add a time value to a whole marking (or selection), thus increasing the value of each timestamp. This is needed to take into account the value of the global clock when later checking for activation of a transition.

Definition 5.20 (Timestamp Addition)**.** Given a marking $M = \sum_{i=0}^{n}(p_i, \tau_i)$ of a TPT net $TN = (P, T, pre, post)$. We can then increase the timestamp of each of the marking's tokens by a given value τ, written $M^{+\tau}$, and defined by $M^{+\tau} = \sum_{i=0}^{n}(p_i, \tau_i + \tau)$.

Example 5.21 (Timestamp Addition). Consider a marking $M = (p_1, 2) \oplus (p_2, 3)$.
By adding a value of 5, we obtain the marking $M^{+5} = (p_1, 2{+}5) \oplus (p_2, 3{+}5) = (p_1, 7) \oplus (p_2, 8)$

Definition 5.22 (Time-Extended Function $\square_{\times\mathbb{R}}$)**.** Given a function $f_P : P_1 \to P_2$. Then we define the *timed extension* $f_{P\times\mathbb{R}}$ of f_P as

$$f_{P\times\mathbb{R}} = (f_P \times id_{\mathbb{R}}) : (P_1 \times \mathbb{R}) \to (P_2 \times \mathbb{R}).$$

Remark 5.23 (Time-Extended Function $\square_{\times\mathbb{R}}$). The time-extension $\square_{\times\mathbb{R}}$ along with $\square^{\oplus}$ applied to a function $f_P : P_1 \to P_2$ between places results in a function $f^{\oplus}_{P\times\mathbb{R}} : (P_1 \times \mathbb{R})^{\oplus} \to (P_2 \times \mathbb{R})^{\oplus}$ between markings over P_1 and P_2.

Fact 5.24 (Linearity of Timestamp Addition)**.** Given a timed P/T-net $TN = (P, T, pre, post)$, timed markings $M_1, M_2 \in (P \times \mathbb{R})^{\oplus}$ of TN, a time value $\tau \in \mathbb{R}$ and a function $f_P : P \to P'$. Then we have

1. $M_1 \overset{\rightarrow}{=} M_2 \Leftrightarrow M_1^{+\tau} \overset{\rightarrow}{=} M_2^{+\tau}$, and
2. $f^{\oplus}_{P\times\mathbb{R}}(M^{+\tau}) = f^{\oplus}_{P\times\mathbb{R}}(M)^{+\tau}$.

Proof.

1. Let $M_1 \overset{\rightarrow}{=} M_2$ and let us assume that $M_1^{+\tau} \overset{\rightarrow}{\neq} M_2^{+\tau}$. Then there is $p \in P$ such that $M_1^{+\tau}[p] \not\leq M_2^{+\tau}[p]$, which means that for $M_1[p] = r_1 \dots r_n$ and $M_2[p] = s_1 \dots s_n$ there exists $i \in \{0, \dots, n\}$ such that $r_i + \tau > s_i + \tau$. But this means that $r_i > s_i$ and hence $M_1 \overset{\rightarrow}{\neq} M_2$ which is a contradiction.
   The argumentation in the other direction works completely analogously.

2. Let $M = \sum_{i=0}^{n} (p_i, r_i)$. Then we have

$$\begin{aligned}
f^{\oplus}_{P\times\mathbb{R}}(M^{+\tau}) &= f^{\oplus}_{P\times\mathbb{R}}((\sum_{i=0}^{n} (p_i, r_i))^{+\tau}) = f^{\oplus}_{P\times\mathbb{R}}(\sum_{i=0}^{n} (p_i, r_i + \tau)) \\
&= \sum_{i=0}^{n} f_{P\times\mathbb{R}}(p_i, r_i + \tau) = \sum_{i=0}^{n} (f_P(p_i), r_i + \tau) \\
&= (\sum_{i=0}^{n} (f_P(p_i), r_i))^{+\tau} = (\sum_{i=0}^{n} f_{P\times\mathbb{R}}(p_i, r_i))^{+\tau} \\
&= (f^{\oplus}_{P\times\mathbb{R}}(\sum_{i=0}^{n} (p_i, r_i)))^{+\tau} = (f^{\oplus}_{P\times\mathbb{R}}(M))^{+\tau}
\end{aligned}$$

□

In the following, we define the projection of a selection (of a specific marking) onto a different marking, retaining the amount of tokens of the selection, however with different timestamps.

Definition 5.25 (Projection of Selections)**.** Given a timed P/T-net $TN = (P, T, pre, post)$ and timed markings

$$M_1 = \sum_{i=1}^{n} \sum_{j=1}^{n_i} (p_i, r^i_j) \text{ and } M_2 = \sum_{i=0}^{n} \sum_{j=1}^{n_i} (p_i, s^i_j)$$

of TN with $M_1 \overset{\rightarrow}{=} M_2$ and for all $1 \leq i \leq n, p_i \in P$:

$$M_1[p_i] = [r^i_1 \dots r^i_{n_i}] \text{ and } M_2[p_i] = [s^i_1 \dots s^i_{n_i}].$$

Let S_2 be a selection of M_2. Then the *projection of* S_2 *to* M_1, written $S_2 \downarrow M_1$, is defined by

$$S_2 \downarrow M_1 = \sum_{(p_i,s_j^i)\leq S_2} (p_i, r_j^i)$$

.

Remark 5.26 (Projection of Selection). Note that the relation $\stackrel{\rightarrow}{=}$ is reflexive which means that for a marking M we have $M \stackrel{\rightarrow}{=} M$. Thus, for a selection $S \leq M$ we can obtain $S = S \downarrow M$ as the projection of itself to M.

Fact 5.27 (Projections are Selections)**.** Given a timed P/T-net $TN = (P, T, pre, post)$ and timed markings M_1 and M_2 of TN with $M_1 \stackrel{\rightarrow}{=} M_2$. Let S_2 be a selection of M_2. Then the projection $S_1 = S_2 \downarrow M_1$ is a selection of M_1 with $S_1 \stackrel{\rightarrow}{=} S_2$.

Proof. Due to the fact that $M_1 \stackrel{\rightarrow}{=} M_2$, the markings M_1 and M_2 have the same location. So we have markings

$$M_1 = \sum_{i=1}^{n}\sum_{j=1}^{n_i} (p_i, r_j^i) \text{ and } M_2 = \sum_{i=0}^{n}\sum_{j=1}^{n_i} (p_i, s_j^i)$$

of TN where for all $1 \leq i \leq n, p_i \in P$:

$$M_1[p_i] = [r_1^i \dots r_{n_i}^i] \text{ and } M_2[p_i] = [s_1^i \dots s_{n_i}^i].$$

Now, $(p_i, s_j^i) \leq S_2$ means $(p_i, s_j^i) \leq M_2$ and, thus, $(p_i, r_j^i) \leq M_1$. Hence, $S_2 \downarrow M_1$ as defined above is a selection of M_1.
Moreover, $M_1 \stackrel{\rightarrow}{=} M_2$ implies that $r_j^i \leq s_j^i$ for all $(p_i, s_j^i) \leq S_2$ which means that $S_1[p_i] \leq S_2[p_i]$ for all $p_i \in P$, i.e. $S_2 \downarrow M_1 \stackrel{\rightarrow}{=} S_2$. □

Example 5.28 (Projection of Selections). Figure 17 shows two markings M_1 and M_2 with $M_1 \stackrel{\rightarrow}{=} M_2$, with selection $S_2 \leq M_2$ shown as a subsum of M_2. Furthermore, $S_2 \downarrow M_1$ is shown as a subsum of M_1. As it can be seen in the illustration, M_1 and M_2 have the same location, as do $S_2 \downarrow M_1$ and S_2, and $S_2 \downarrow M_1 \stackrel{\rightarrow}{=} S_2$ holds.

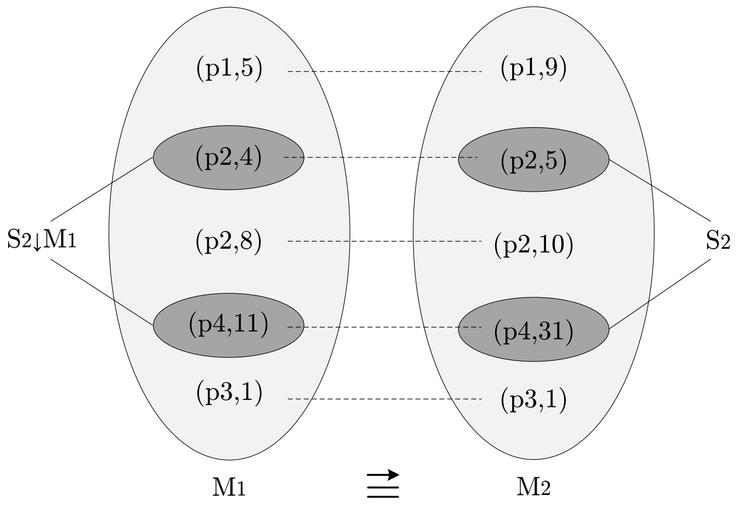


Figure 17: Projection of Selections

5.2.2 Activation and Firing

Finally, we define the activation of a transition. Note that the input edges of transitions are also involved when checking for activation, in particular a positive inscription on an input edge enables the transition to consume tokens before the global clock actually 'reaches' the timestamp of these tokens. Likewise, a negative inscription would delay the input tokens even more. An example for this is given below.

Definition 5.29 (Activation)**.** Given a timed P/T-net $TN = (P, T, pre, post)$, with state (TN, M, τ) and a selection $S \leq M$. Then $t \in T$ *is activated under* (S, τ) if $pre(t)^{+\tau}$ is a location-strict delay of S, i. e. $pre(t)^{+\tau} \stackrel{\leftarrow}{=} S$.

Example 5.30 (Activation). In Figure 15, we have $pre(produce)^{+60} = (Utility, 0 + 60) \oplus (Worker, 0 + 60)$, which is a location-strict delay of the selection $S = (Utility, 60) \oplus (Worker, 25)$. Therefore, *produce* is activated under S at global time $\tau = 60$.

Firing steps in timed P/T nets are defined very similar to those of algebraic P/T nets, however the value of the global clock gets added to the newly created tokens in order to incorporate the global time value. The resulting tokens then have timestamps with an offset from the global clock value, given by the inscribed time values of the output edges of the firing transition.

This corresponds for example to resources in a production line that are available only from a certain point in time (indicated by the timestamp), meaning that a part of the line depending on that resource (a transition) has to wait until it becomes available.

In general, the time values inscribed on the edges can be seen as representing the duration of a transition (at the output edges), as well as indicating the possibility of removing a token early (at the input edges, like mentioned above). Returning to the example of a production line, the duration at an output edge denotes the time a production process (the transition) takes until the resulting product is finished and available for the next step in the line.

Of course, negative time values are also permitted, thus allowing models that are not limited to simple (positive) durations.

Definition 5.31 (Firing Step)**.** Given a timed P/T-net $TN = (P, T, pre, post)$ with state (TN, M, τ) of TN with a global clock value τ and $t \in T$ activated under (S, τ) with $S \leq M$. Then we say that there is a *firing step*

$$M \xrightarrow{(t,S,\tau)} M',$$

where the follower marking M' is given by

$$M' = M \ominus S \oplus post(t)^{+\tau}.$$

Example 5.32 (Firing Step). In Figure 15, with the selection $S = (Utility, 60) \oplus (Worker, 25)$, the follower marking after firing of *produce* at time $\tau = 60$ is

$$\begin{aligned} M' &= M \ominus S \oplus post(produce)^{+\tau} \\ &= ((Utility, 60) \oplus (Worker, 25) \oplus (Worker, 110) \oplus (Product, 110)) \\ &\ominus ((Utility, 60) \oplus (Worker, 25)) \\ &\oplus (Worker, 100 + \tau) \oplus (Utility, 50 + \tau) \oplus (Product, 100 + \tau) \\ &= (Utility, 110) \oplus (Worker, 110) \oplus (Worker, 160) \oplus (Product, 110) \\ &\quad \oplus (Product, 160) \end{aligned}$$

Then, we can concatenate multiple firing steps to a firing sequence. Note that a firing step only results in a marking, not a particular state. The model time is advanced using timesteps (explicitly or implicitly), thus advancing the time to the next desired value at which a firing step can occur.

Definition 5.33 (Firing Sequence)**.** Given a timed P/T state (TN, M_0, τ_0) with timed P/T net $TN = (P, T, pre, post)$, marking M_0 of TN, global clock value τ_0, and $t_i \in T$ activated under (TN, S_i, τ_i) for $i \in \{0, \dots, n-1\}$ and $S_i \leq M_i$.

Then,

$$\begin{aligned} Seq = (TN, M_0, \tau_0) &\xrightarrow{(t_0,S_0)} (TN, M_1, \tau_0) \xrightarrow{\Delta\tau_0} (TN, M_1, \tau_1) \xrightarrow{(t_1,S_1)} \dots (TN, M_{n-1}, \tau_{n-2}) \\ &\xrightarrow{\Delta\tau_{n-2}} (TN, M_{n-1}, \tau_{n-1}) \xrightarrow{(t_{n-1},S_{n-1})} (TN, M_n, \tau_{n-1}) \end{aligned}$$

is a *firing sequence* in the net TN, if for all $i \in \{0, \dots, n-1\}$: $M_i \xrightarrow{(t_i,S_i,\tau_i)} M_{i+1}$ is a firing step.

The following is a shorter variant of the same firing sequence, omitting the timesteps, with the clock value at the time of firing included in the firing step notation.

$$Seq = M_0 \xrightarrow{(t_0,S_0,\tau_0)} M_1 \xrightarrow{(t_1,S_1,\tau_1)} \dots \xrightarrow{(t_{n-1},S_{n-1},\tau_{n-1})} M_n$$

Note that there is no constraint on the global clock values. This means that time values can actually decrease while moving forward in the firing sequence. We can, however, enforce different restrictions on firing sequences in order to achieve a certain behaviour:

The sequence *Seq* is called *time-monotonic*, if for $0 \leq i < n$, there is $\tau_i \leq \tau_{i+1}$.

The firing sequence employs *eager firing*, if for all firing steps $M_i \xrightarrow{(t_i,S_i,\tau_i)} M_{i+1}$ in *Seq*, there is no firing step $M_i \xrightarrow{(t'_i,S'_i,\tau'_i)} M'_{i+1}$ with $\tau'_i < \tau_i$.

Example 5.34 (Firing Sequences). The somewhat liberal (compared to e. g. timed CPNs) definition of firing sequences allows clock values in the firing sequence without any restriction regarding their sequence. For example, the following is a valid firing sequence in the timed net TN shown in Figure 18, as long as each firing step exists:

$$M_1 \xrightarrow{(t_1,S_1,100)} M_2 \xrightarrow{(t_2,S_2,55)} M_3 \xrightarrow{(t_3,S_3,250)} M_4$$

However, the most common usage of firing sequences are time-monotonic firing sequences, which requires the clock values to be monotonically increasing, such as the following sequence:

$$M_1' \xrightarrow{(t_2,S_2,55)} M_2' \xrightarrow{(t_1,S_1,100)} M_3' \xrightarrow{(t_3,S_3,350)} M_4'$$

Note that firing steps do not actually change the global clock. Only time steps can change the global clock, while for the other net classes the clock value is determined by the clock value in the firing step.

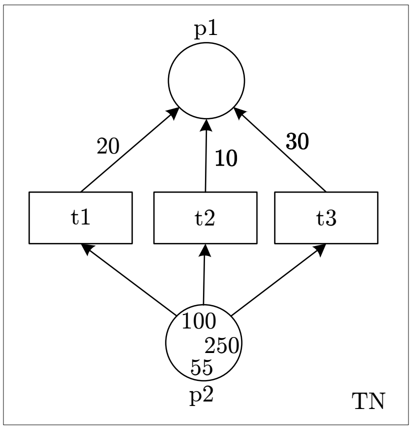


Figure 18: Timed P/T net TN

5.3 Application to Case Studies

With the notion of timed P/T nets defined, we can now apply the definitions to the case studies from Section 3. We revisit the network and production line examples, simulating the models using the newly defined timed P/T firing behaviour.

5.3.1 Network Infrastructure

We can now extend the example P/T net from Section 3 into a timed P/T net (Figure 19), which means that each edge gets assigned a time duration, representing the time it takes for that specific transition to finish.

In the timed P/T in Figure 19, we model that clients differ from each other in terms of speed, resulting in a higher latency for the *slowclient* and a lower latency for the *fastclient* compared to *client3* and *client4* (of which *client3* is a little slower than *client4*). For

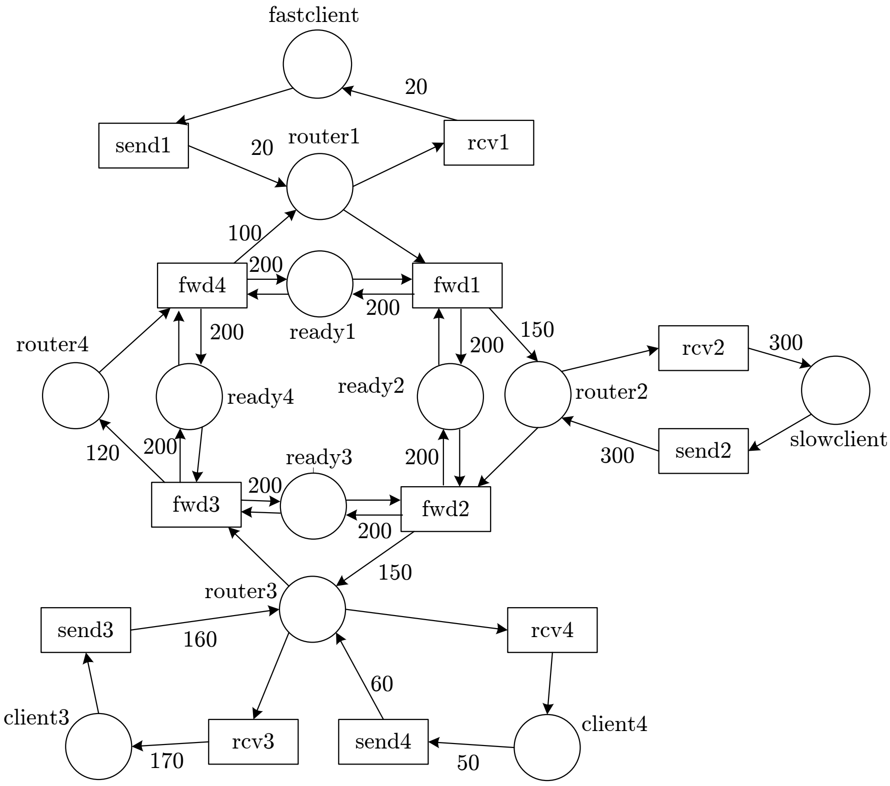


Figure 19: Network infrastructure - Timed P/T net

example, it takes *client3* 60 units of time to send a packet to the connected switch, while it takes 50 units of time to move a packet from that switch to the client.

The routers take 200 time units after forwarding a packet, therefore the output edges of the forward-transitions connected to the ready-places get assigned the duration 200. This is a longer time duration than the actual forwarding takes, and will effectively delay the use of a router's forwarding transitions after it has delivered a packet to another router.

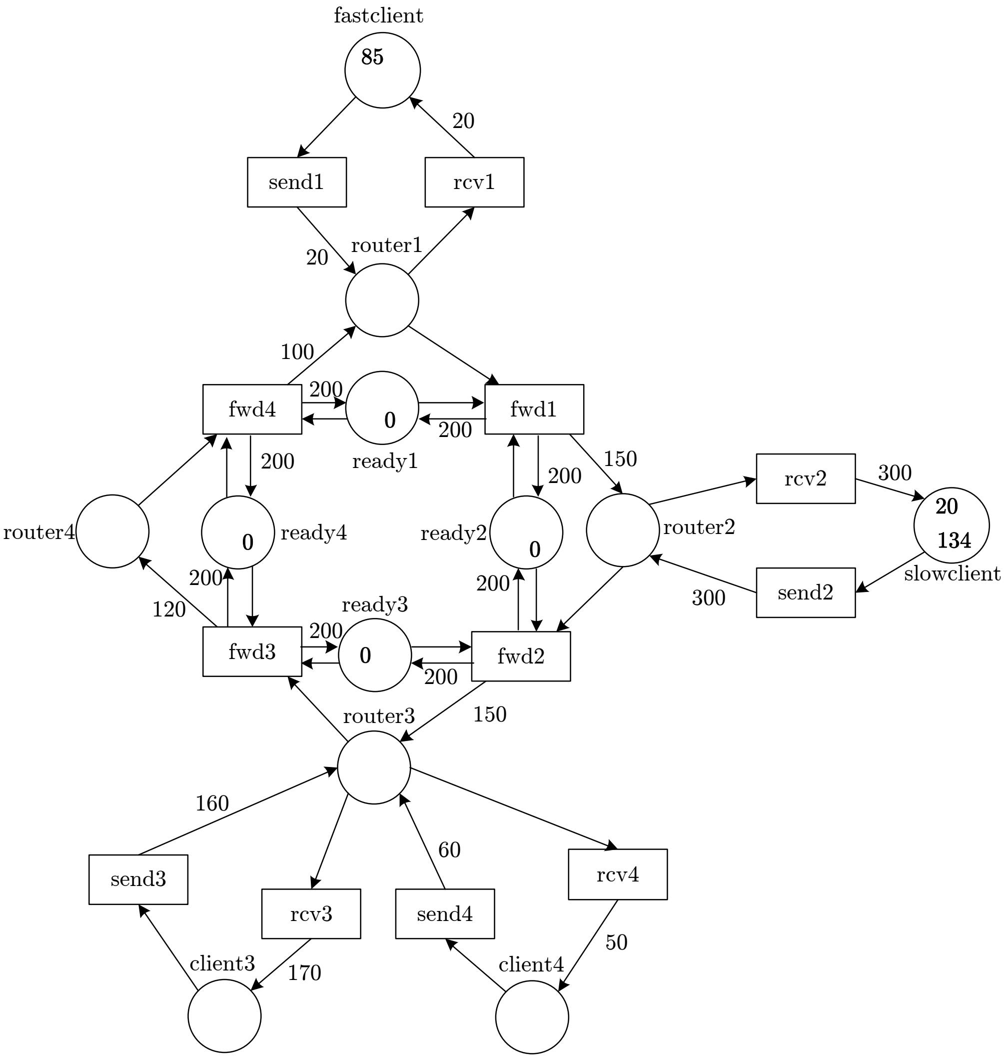


Figure 20: Network infrastructure - Timed P/T net with marking

In Figure 20, we add a *timed marking*, assigning some tokens with timestamps to client places. When simulating this model (i.e. continuously conducting the "token game"), the packets that are present on the places representing the fast and slow clients are passed around the network, and depending on which transitions are chosen to fire, can arrive at other clients. The timestamps of the tokens continuously rise over the course of the simulation, since all delays are positive (which is the intuitive usage of delays).

The following illustrations (Fig. 21 - 24) show the markings obtained by firing the transitions send1, fwd1, fwd2, rcv3, in order, which simulates the sending of a packet from fastclient to client3.

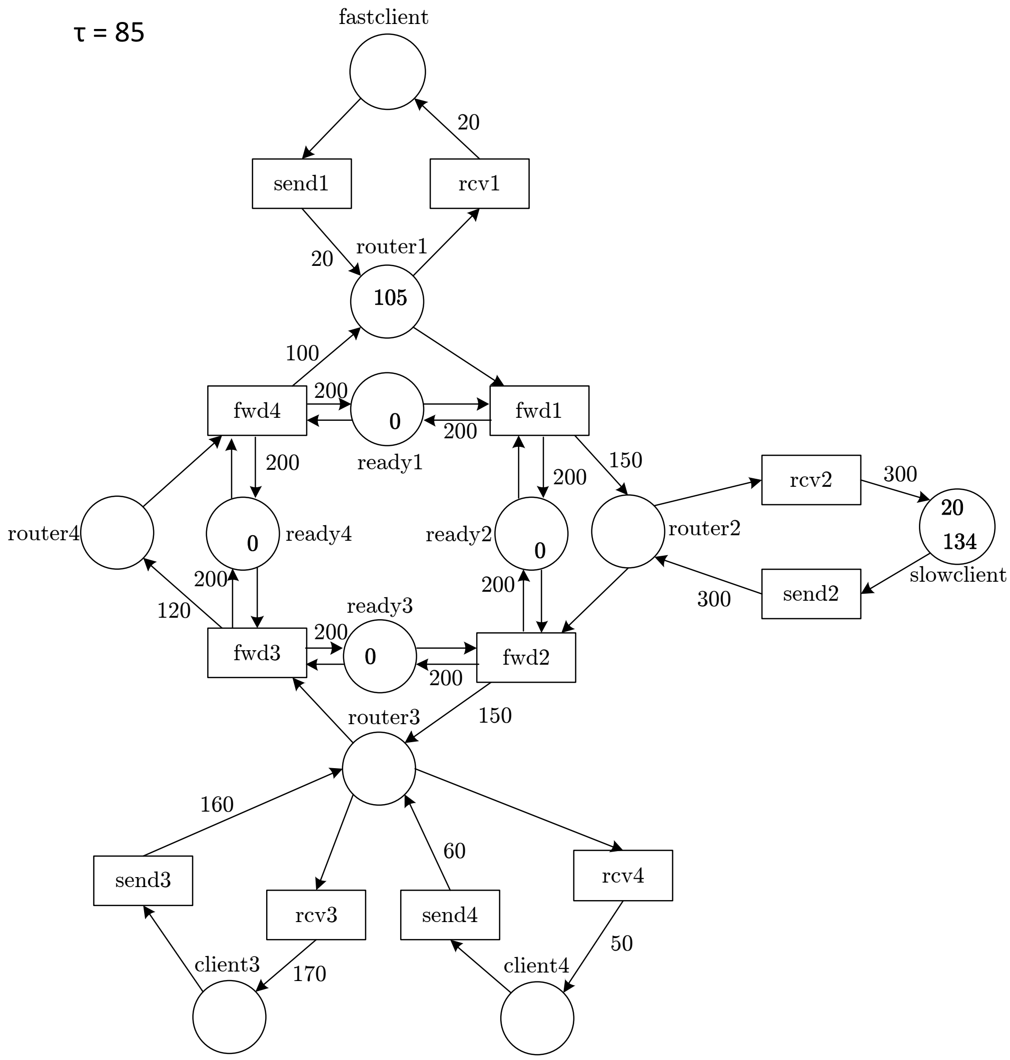


Figure 21: After firing of send1 at global clock value 85

Figure 21 shows the net after send1 has been fired at global clock value 85. Note that this is the earliest possible time at which the token on fastclient could have been used, since its time stamp is 85, and the input edge of send1 is 0 (left empty in the visualisation, as per notation).

The token created by send1 (representing the packet sent through the network) is assigned a delay of 20 time units, which is added to the clock value when firing, resulting in a time stamp of 105.

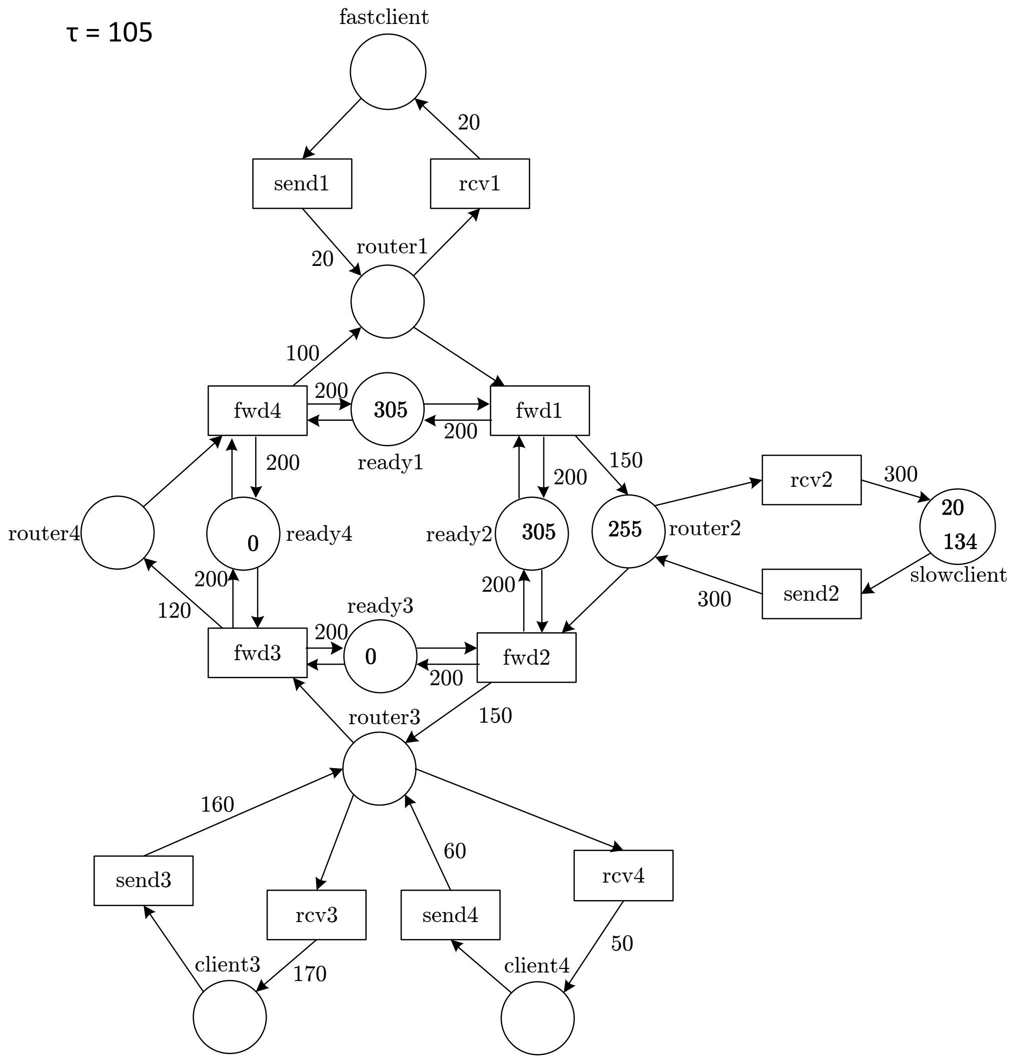


Figure 22: After firing of fwd1 at 105

Figure 22 shows the net after fwd1 has been fired at global clock value 105. Note that there are tokens created on the ready places of both involved routers with the designated delay of 200 time units.

The “packet token” is now located on router2 with a timestamp of 255.

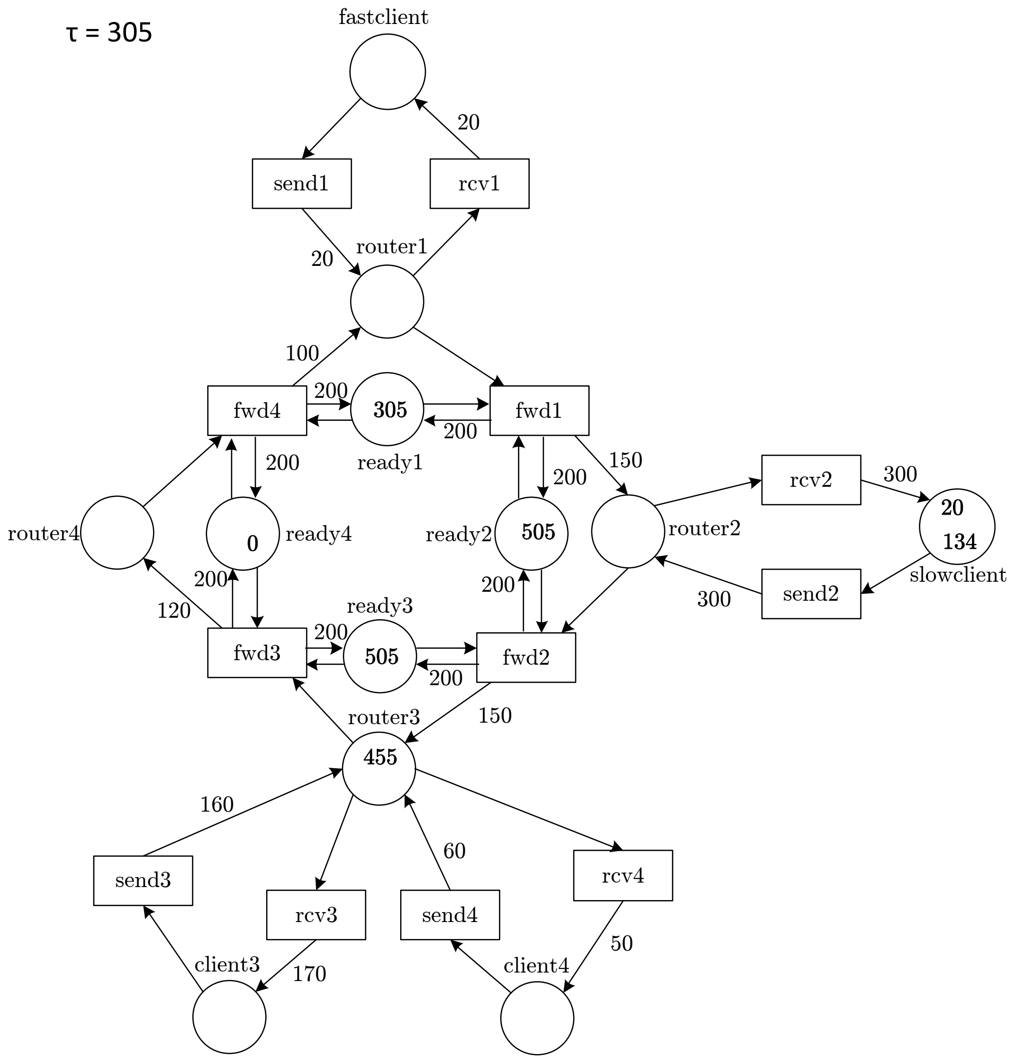


Figure 23: After firing of fwd2 at global clock value 305

Figure 23 shows the net after fwd2 has been fired at global clock value 305. In this case, 305 is the earliest point in time at which fwd2 could have been fired due to the timestamp of 305 of the token on ready2 (instead of 255, which is the timestamp of the "packet token", which could have been theoretically used for firing of rcv2 at time 255). The "packet token" now has a timestamp of 455.

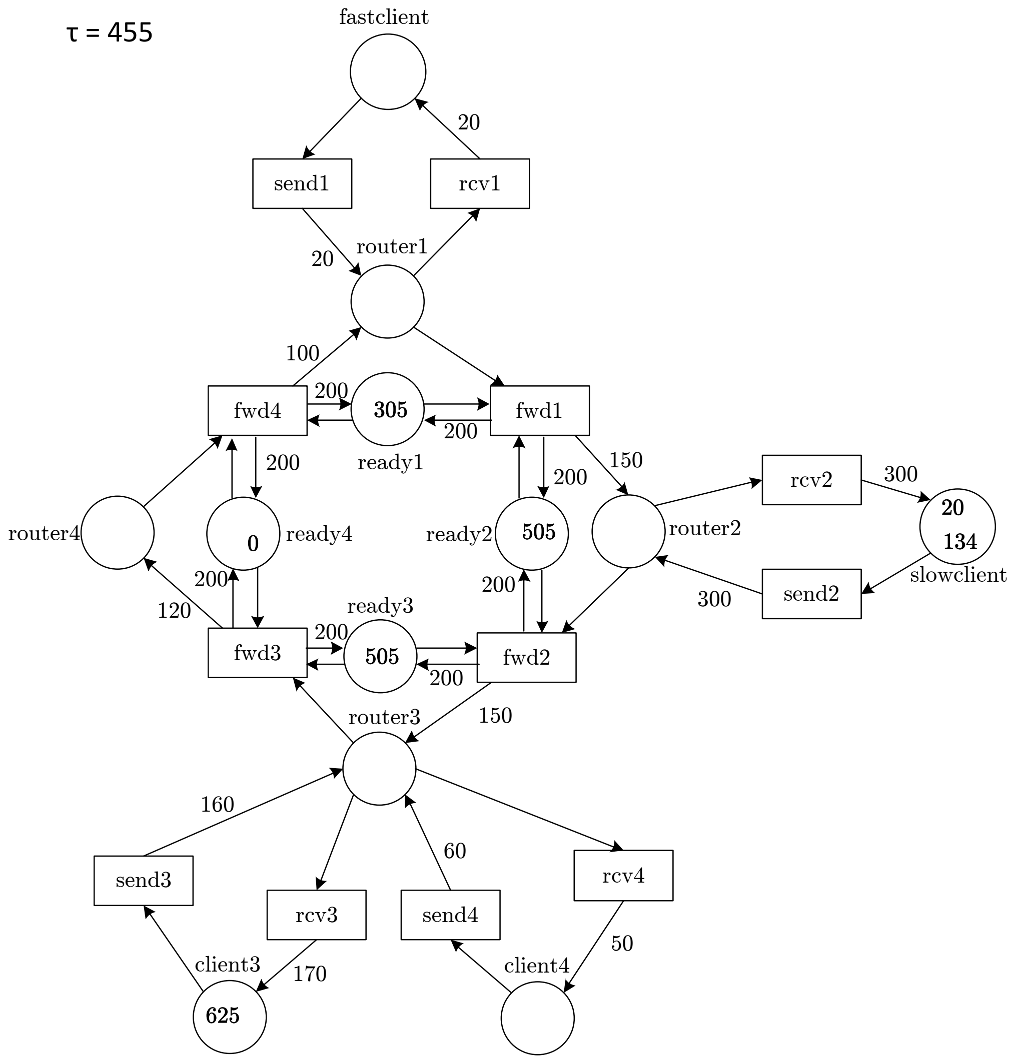


Figure 24: After firing of rcv3 at global clock value 455

Figure 24 shows the net after rcv3 has been fired at global clock value 455. Since the only input place of rcv3 is router3, rcv3 can be fired at the clock value dictated by the "packet token". The newly created token gets assigned the timestamp 625, which is the point in time at which the packet sent arrives at its destination.

5.3.2 Production Line

In the production line example, we can now assign time durations to the transitions. Consider the timed P/T net in Figure 25. The production step, represented by transition produce, takes 100 time units for the product to be ready, while the utility (represented by the token on the *Utility* place) is only used for 50 time units. The utility token is ready to be used at global clock value 0, while the workers (represented by the two tokens on the *Worker* place) are "ready" at global time 10 and 25, respectively. The *break* transition has a time value of -5 for its input token, meaning that a worker token that is about to take a break can do so not earlier than 5 time units after they would be available according to their time stamp. This means that negative time values in the pre-domain of a transition may delay the time at which the transition can be fired.

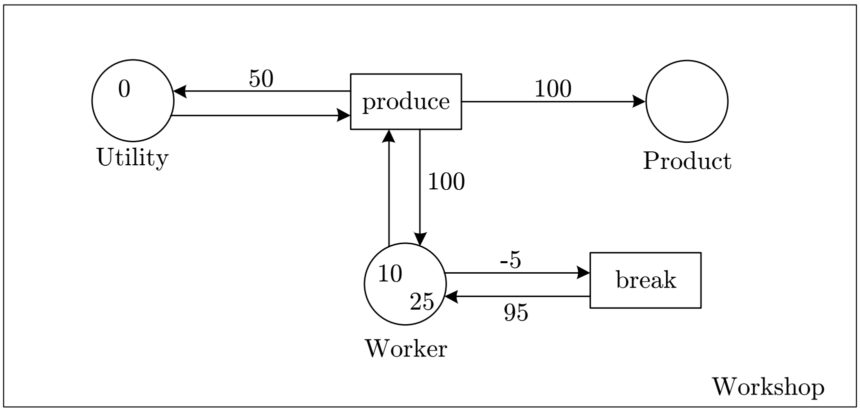


Figure 25: Production line as timed P/T net

Figure 26 shows the timed P/T net after *produce* has been fired at global clock value $\tau = 10$. The resulting product is ready 100 time units after (at clock value 110, indicated by the token's timestamp), whereas the utility is available again only 50 time units after firing (at clock value 60). The worker is occupied until the product is finished, so the worker token is also available at clock value 110.

Since the utility is available at global clock value 60, and there is a second worker available, who is available at clock value 25, the produce transition can already fire again at $\tau = 60$.

Figure 27 shows the timed P/T net after *produce* has been fired at global clock value $\tau = 60$, placing a second product token on the *Product* place. The utility is available at clock value 110, which is the same time at which the first worker token is ready again. Therefore, the transition *produce* could fire again at $\tau = 110$. However, we will fire the *break* transition next, which we demonstrate in detail in the next illustration, since it incorporates a negative time duration on the input edge.

Again, consider the state shown in Figure 27. Using the selection $S = (Worker, 110)$, the transition *break* is activated at $\tau = 115$, since $pre(break)^{+\tau} \overset{\leftarrow}{=} S$, i.e. $(Worker, -5 + 115) = (Worker, 110) \overset{\leftarrow}{=} (Worker, 110)$. Therefore, the transition *break* can fire at $\tau = 115$. The negative value of -5 on the input edge results in the selected worker token having to "wait" 5 time units after it is ready according to the global clock before it can be used in the

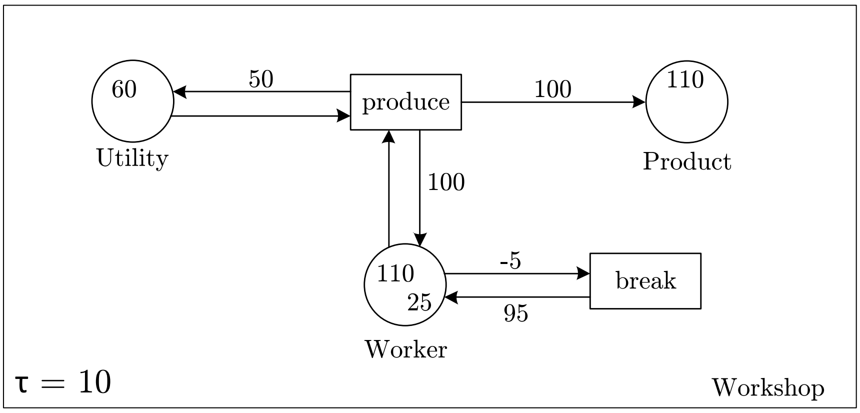


Figure 26: Production line after firing of produce at $\tau = 10$

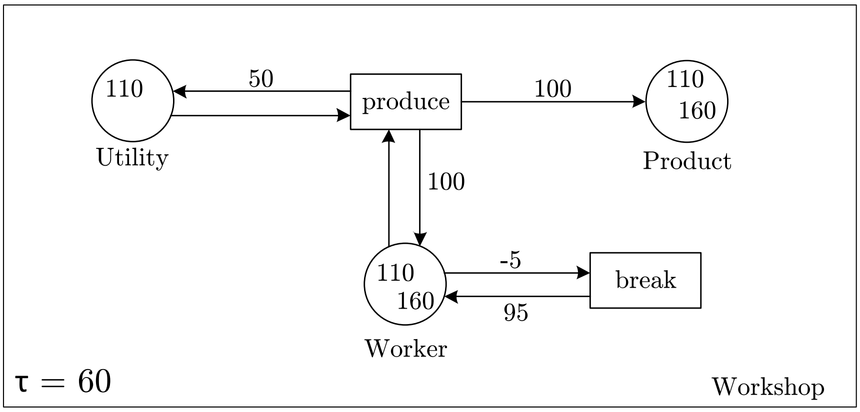


Figure 27: Production line after firing of produce at $\tau = 60$

transition. Therefore, even though the token has a timestamp of 110, the transition is not activated before the global clock value of 115. The output edge delay works the same way as before, so the token created by *break* is usable at clock value 210.

Figure 28 shows the resulting state after *break* has been fired at time $\tau = 115$.

A small modification to the production line net, as seen in Figure 29, shows another possible application of having time values other than zeroes assigned to the input edges of transitions. The produce transition now has a time value of 25 for the input token. This means that the Utility can now be used 25 time units before the time indicated by the token's timestamp (possibly due to the utility being able to be shared between workers).

For the activation, this means that the transition is activated at a clock value 25 time units before the input token's timestamp would allow. In this case, *produce* is activated at $\tau = 70$ under the selection $S = (Utility, 95) \oplus (Worker, 10)$, because then $S \overset{\rightrightarrows}{=} pre(t)^{+\tau}$ holds true, since $pre(produce) = (Utility, 25) \oplus (Worker, 0)$ and thus $pre(produce)^{+\tau} = (Utility, 95) \oplus (Worker, 70) \overset{\leftleftarrows}{=} (Utility, 95) \oplus (Worker, 10)$.

Therefore, produce can fire already at clock value 70, resulting in the marking shown

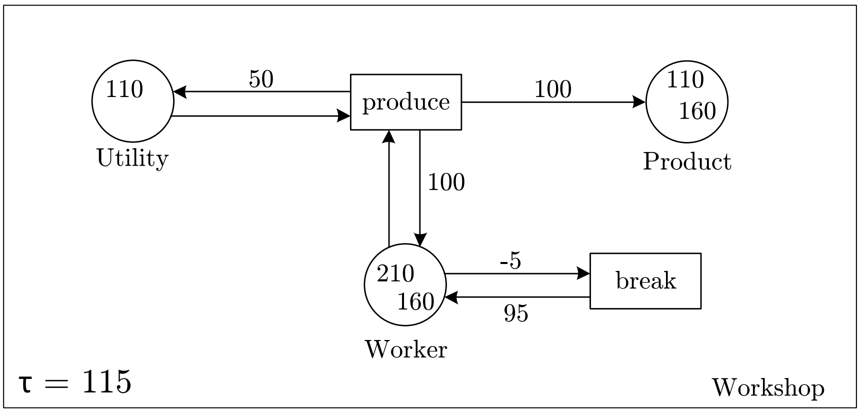


Figure 28: Production line after firing of break at $\tau = 115$

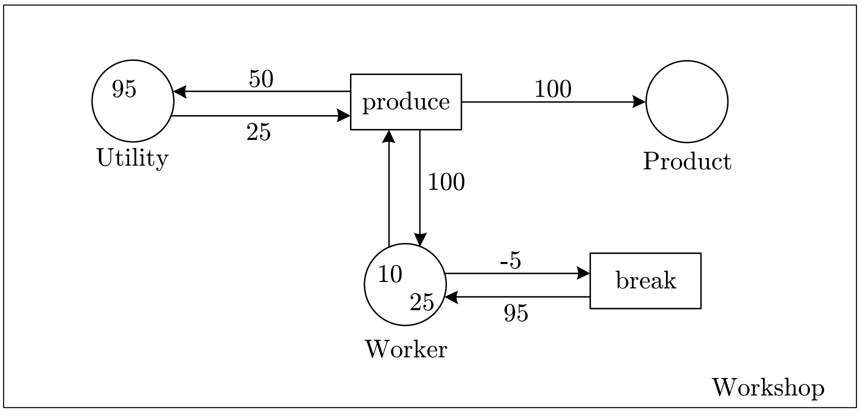


Figure 29: Production line with pre-emptive token removal

in Figure 30. Afterwards, produce can fire at clock value 95, leading to the final marking shown in Figure 31.

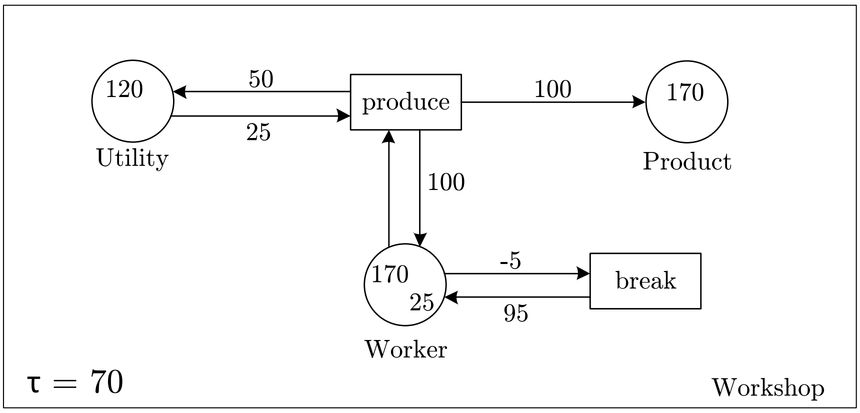


Figure 30: Production line after firing of produce at $\tau = 70$

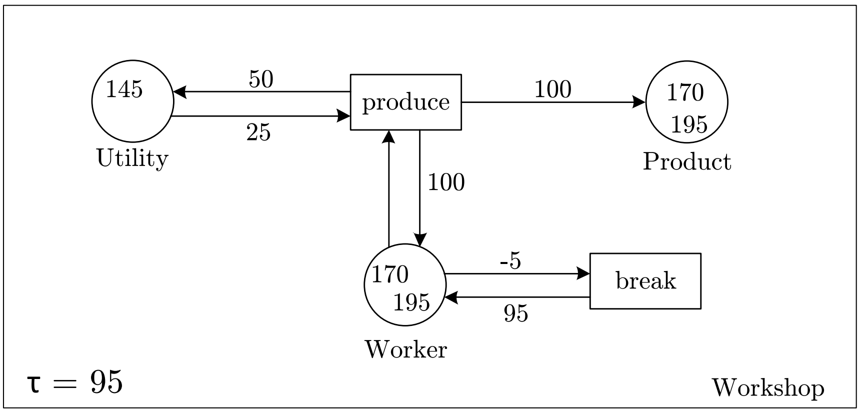


Figure 31: Production line after firing of produce at $\tau = 95$

6 Categories of Timed Net Classes

In this subsection we define the categories of timed P/T nets, timed P/T systems, as well as timed P/T states. Based on the definition of timed P/T nets (Definition 5.1), we define timed P/T systems analogously to P/T systems. Furthermore, we define the category of timed P/T states, based on Definition 5.7.

The categories of timed P/T systems and timed P/T states are used to establish a correlation between systems and states, showing that any timed P/T state can be expressed as a timed P/T system and vice versa, using functors. We do this by showing that these functors preserve the firing behaviour of the translated timed systems and states, respectively, and that the functors establish an equivalence of the categories (meaning that there is a relation between the categories that implies they are essentially the same).

Furthermore, we define "skeleton" functors that translate timed P/T nets and -systems to regular P/T nets and systems, while preserving the firing behaviour of the respective nets.

Remark 6.1 (Examples). For the examples in this subsection, we use a subnet of the network infrastructure case study in Section 3.1, using only the places client3, client4 and router3, as well as the transitions connecting these places (rcv3, rcv4, send3, send4). The illustration in Figure 32 shows which part of the network infrastructure timed P/T net is used.

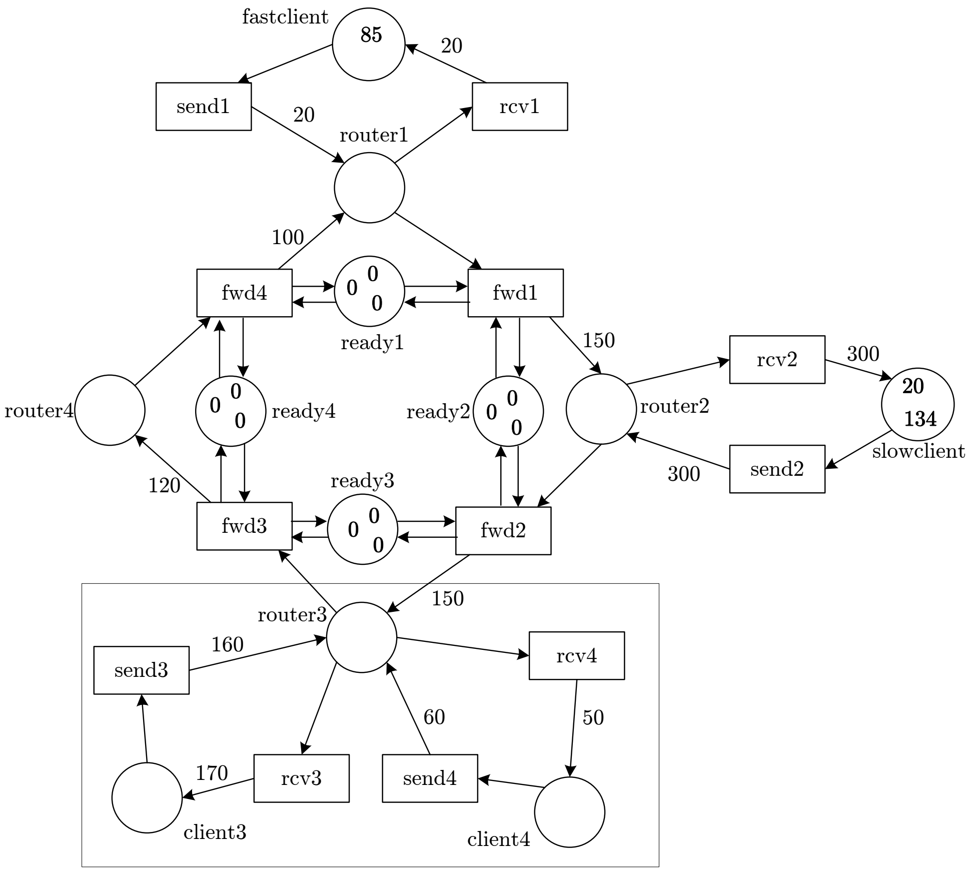


Figure 32: Subnet of the network infrastructure net

6.1 Category of Timed P/T Nets

Definition 6.2 (Timed P/T Morphism)**.** Given timed P/T-nets $TN_i = (P_i, T_i, pre_i, post_i)$, for $i \in \{1, 2\}$. A *timed P/T-net-morphism* $f : TN_1 \to TN_2$ is defined by $f = (f_P, f_T)$, with $f_P : P_1 \to P_2$ and $f_T : T_1 \to T_2$, such that for all $t \in T_1$:

- $pre_2 \circ f_T(t) \overset{\leftleftarrows}{=} f^{\oplus}_{P\times\mathbb{R}} \circ pre_1(t)$, and
- $post_2 \circ f_T(t) \overset{\rightrightarrows}{=} f^{\oplus}_{P\times\mathbb{R}} \circ post_1(t)$.

A timed P/T-morphism is called *time-strict* if for all $t \in T$

- $pre_2 \circ f_T(t) = f^{\oplus}_{P\times\mathbb{R}} \circ pre_1(t)$, and
- $post_2 \circ f_T(t) = f^{\oplus}_{P\times\mathbb{R}} \circ post_1(t)$.

If a morphism f is time-strict and injective, we shortly say that f is time-strict injective.

Fact 6.3 (Category **TPTNets** of Timed P/T Nets)**.** The category of timed P/T nets, **TPTNets** consists of the class of all timed P/T nets as objects, as well as timed P/T morphisms. The composition of two timed P/T morphisms $g \circ f$ is defined componentwise as $g \circ f = ((g \circ f)_P, (g \circ f)_T) = (g_P \circ f_P, g_T \circ f_T)$. The identity morphism for each timed P/T net A is defined as $id_A : A \to A : id = (id_P, id_T)$.

Proof. For the proof of Fact 6.3, see Appendix B.1. □

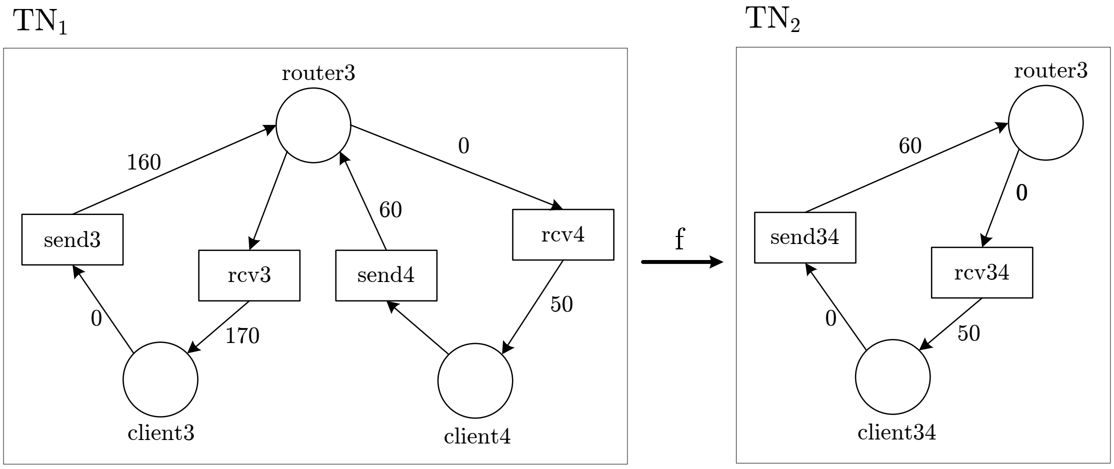


Figure 33: Timed P/T nets

Example 6.4 (Timed P/T Morphism). Consider the two timed nets $TN_i = (P_i, T_i, pre_i, post_i)$ for $i = 1, 2$ from Figure 33. Let $f = (f_P, f_T) : TN_1 \to TN_2$ with

- $f_T(send3) = f_T(send4) = send34, f_T(rcv3) = f_T(rcv4) = rcv34,$
- $f_P(router3) = router3$ and $f_P(client3) = f_P(client4) = client34.$

The following holds:

- $f^{\oplus}_{P\times\mathbb{R}} \circ pre_1(send4) = f^{\oplus}_{P\times\mathbb{R}} \circ pre_1(send3) = (client34, 0) = pre_2 \circ f_T(send3) = pre_2 \circ f_T(send4),$
- $f^{\oplus}_{P\times\mathbb{R}} \circ post_1(send3) = (router3, 160) \stackrel{\leftarrow}{=} (router3, 60) = post_2 \circ f_T(send3),$
- $f^{\oplus}_{P\times\mathbb{R}} \circ post_1(send4) = (router3, 60) = post_2 \circ f_T(send3).$
- $f^{\oplus}_{P\times\mathbb{R}} \circ pre_1(rcv3) = f^{\oplus}_{P\times\mathbb{R}} \circ pre_1(rcv4) = (router3, 0) = pre_2 \circ f_T(rcv3) = pre_2 \circ f_T(rcv4),$
- $f^{\oplus}_{P\times\mathbb{R}} \circ post_1(rcv3) = (client34, 170) \stackrel{\leftarrow}{=} (client34, 50) = post_2 \circ f_T(rcv3),$
- $f^{\oplus}_{P\times\mathbb{R}} \circ post_1(rcv4) = (client34, 170) = post_2 \circ f_T(rcv3).$

Thus, f is a timed P/T-morphism.

Next, we show that timed P/T morphisms preserve firing steps. For this, we define lemmas regarding the delay of sums and differences.

Lemma 6.5 (Location of Sums) Given a set P and timed markings $A, B, C, D \in (P \times \mathbb{R})^{\oplus}$ with $\pi_P^{\oplus}(A) = \pi_P^{\oplus}(B)$ and $\pi_P^{\oplus}(C) = \pi_P^{\oplus}(D)$. Then we also have that $\pi_P^{\oplus}(A \oplus C) = \pi_P^{\oplus}(B \oplus D)$.

Proof. $\pi_P^{\oplus}(A \oplus C) = \pi_P^{\oplus}(A) \oplus \pi_P^{\oplus}(C) = \pi_P^{\oplus}(B) \oplus \pi_P^{\oplus}(D) = \pi_P^{\oplus}(B \oplus D)$. □

Lemma 6.6 (Delay of Sums) Given a set P and timed markings $A, B, C, D \in (P \times \mathbb{R})^{\oplus}$ with $A \overset{\leftarrow}{=} B$ and $C \overset{\leftarrow}{=} D$. Then we have $A \oplus C \overset{\leftarrow}{=} B \oplus D$.

Proof-Idea. We show that $(A \oplus C)$ and $(B \oplus D)$ have the same location using Lemma 6.5. Then by restriction to a single place p, we show that $\pi_P^{\oplus}(A|_p \oplus C|_p) = \pi_P^{\oplus}(B|_p \oplus D|_p)$, which holds for the complete sums, since it holds for all places p. For the detailed proof, we refer to Appendix B.5. □

Lemma 6.7 (Delay of Differences) Given a set P and timed markings $A, B, C, D \in (P \times \mathbb{R})^{\oplus}$. Then if $A \overset{\leftarrow}{=} B$, $D \leq B$ and $C = D \downarrow A$ we have $A \ominus C \overset{\leftarrow}{=} B \ominus D$.

Proof-Idea. Again, we show that $(A \ominus C)$ and $(B \ominus D)$ have the same location. Then, we show that $(A \ominus C)[p] \geq (B \ominus D)[p]$ via the element-wise removal of elements from the respective sums. For the detailed proof, we refer to Appendix B.6. □

Theorem 6.8 (Timed P/T Morphisms Preserve Firing Behaviour) Given timed nets $TN_i = (P_i, T_i, pre_i, post_i)$ with $i = 1, 2$, with marking M of TN_1, selection $S \leq M$ and a timed P/T morphism $f = (f_P, f_T), f : TN_1 \to TN_2$. Let $t \in T_1$ be activated under S and $M \xrightarrow{(t,S,\tau)} M'$ a firing step in TN_1 with $M' = M \ominus S \oplus post_1(t)^{+\tau}$.

Then, there is a firing step $f_{P\times\mathbb{R}}^{\oplus}(M) \xrightarrow{(f_T(t), f_{P\times\mathbb{R}}^{\oplus}(S), \tau)} M''$ in TN_2 with $f_{P\times\mathbb{R}}^{\oplus}(M') \overset{\leftarrow}{=} M''$.

Proof. $t \in T_1$ activated under S means that $S \overset{\rightarrow}{=} pre_1(t)^{+\tau}$. Via Fact B.1 (monotonicity of the time-enhanced function) and the timed P/T morphism condition follows $f_{P\times\mathbb{R}}^{\oplus}(S) \overset{\rightarrow}{=} f_{P\times\mathbb{R}}^{\oplus}(pre_1(t)^{+\tau}) = f_{P\times\mathbb{R}}^{\oplus}(pre_1(t))^{+\tau} \overset{\rightarrow}{=} pre_2(f_T(t))^{+\tau}$.

Since $\overset{\rightarrow}{=}$ is an order, we get $f_{P\times\mathbb{R}}^{\oplus}(S) \overset{\rightarrow}{=} pre_2(f_T(t))^{+\tau}$, which means that $f_T(t)$ is activated under $f_{P\times\mathbb{R}}^{\oplus}(S)$.

Also from Fact B.1, we get $f_{P\times\mathbb{R}}^{\oplus}(S) \leq f_{P\times\mathbb{R}}^{\oplus}(M)$, so $f_{P\times\mathbb{R}}^{\oplus}(S)$ is a selection of $f_{P\times\mathbb{R}}^{\oplus}(M)$.

Therefore, there is a firing step $f_{P\times\mathbb{R}}^{\oplus}(M) \xrightarrow{(f_T(t), f_{P\times\mathbb{R}}^{\oplus}(S), \tau)} M''$ in TN_2.

As for the follower marking, $f_{P\times\mathbb{R}}^{\oplus}(M') = f_{P\times\mathbb{R}}^{\oplus}(M \ominus S \oplus post_1(t)^{+\tau})$, we have $f_{P\times\mathbb{R}}^{\oplus}(M \ominus S \oplus post_1(t)^{+\tau}) = f_{P\times\mathbb{R}}^{\oplus}(M \ominus S) \oplus f_{P\times\mathbb{R}}^{\oplus}(post_1(t)^{+\tau}) = f_{P\times\mathbb{R}}^{\oplus}(M) \ominus f_{P\times\mathbb{R}}^{\oplus}(S) \oplus f_{P\times\mathbb{R}}^{\oplus}(post_1(t)^{+\tau})$.

Then, via Lemma 6.6, 6.7 and the morphism condition follows $f_{P\times\mathbb{R}}^{\oplus}(M) \ominus f_{P\times\mathbb{R}}^{\oplus}(S) \oplus f_{P\times\mathbb{R}}^{\oplus}(post_1(t)^{+\tau}) \overset{\leftarrow}{=} f_{P\times\mathbb{R}}^{\oplus}(M) \ominus f_{P\times\mathbb{R}}^{\oplus}(S) \oplus post_2(f_T(t))^{+\tau} = M''$. □

Example 6.9 (Timed P/T Morphisms Preserve Firing Steps). Figure 34 shows the nets from Example 6.4, respectively with a marking M.

Figure 35 shows the same nets after firing of transition $rcv3$ and $f_T(rcv3) = rcv34$ respectively at time $\tau = 100$, resulting in the marking M' of TN_1 and marking M'' of TN_2.

Now,

$$\begin{aligned} M'' &= (client34, 199) \oplus (client34, 250) \oplus (client34, 270) \\ &\overset{\rightarrow}{=} f^{\oplus}_{P\times\mathbb{R}}((client3, 250) \oplus (client3, 270) \oplus (client4, 200)) \\ &= f^{\oplus}_{P\times\mathbb{R}}(M') \end{aligned}$$

Therefore, f preserves firing behaviour.

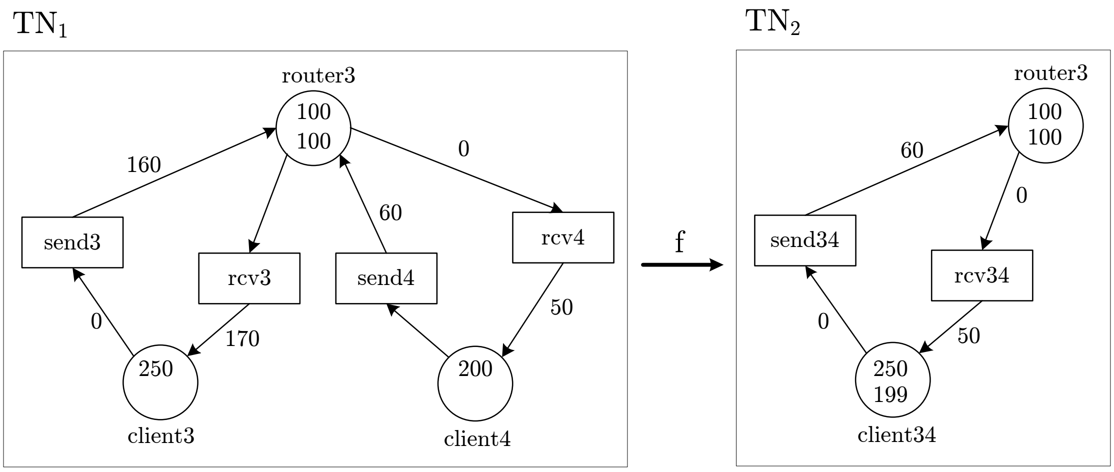


Figure 34: Timed P/T nets before firing

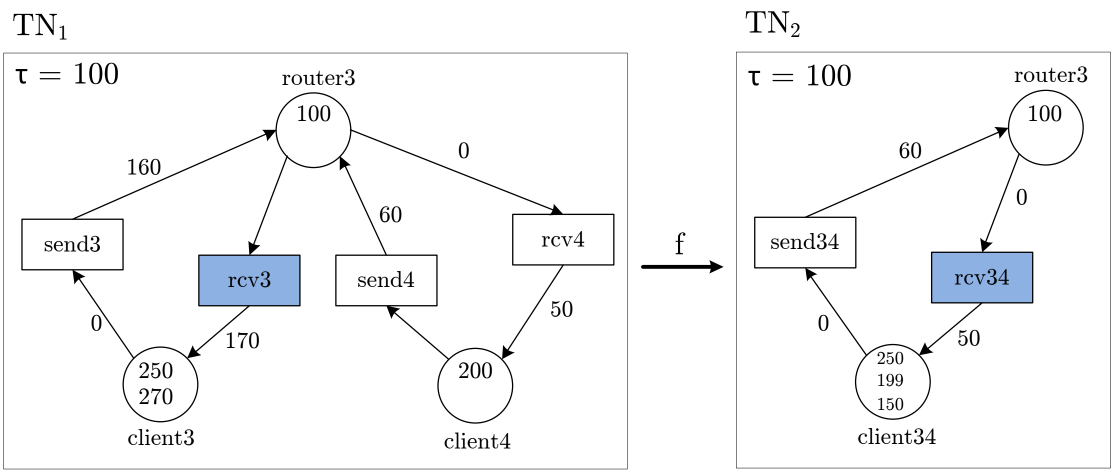


Figure 35: Timed P/T nets after firing

The following lemma states a useful decomposition property of timed P/T morphisms.

Lemma 6.10 (Decomposition of Timed P/T Morphisms) Given timed P/T morphisms $f : TN_0 \to TN_2$, $h : TN_1 \to TN_2$, and functions $g_P : P_0 \to P_1$, $g_T : T_0 \to T_1$ with

$h_P \circ g_P = f_P$ and $h_T \circ g_T = f_T$. If h is time-strict injective then $g = (g_P, g_T)$ is a timed P/T morphism.

$$TN_0 \xrightarrow{g} TN_1 \xrightarrow{h} TN_2, \quad f: TN_0 \to TN_2 \qquad P_0 \xrightarrow{g_P} P_1 \xrightarrow{h_P} P_2, \quad f_P: P_0 \to P_2 \qquad T_0 \xrightarrow{g_T} T_1 \xrightarrow{h_T} T_2, \quad f_T: T_0 \to T_2$$

Proof. We have to show that for all $t \in T_0$ it holds that $pre_1 \circ g_T(t) \overset{\leftarrow}{=} g^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t)$ and $post_1 \circ g_T(t) \overset{\rightarrow}{=} g^{\oplus}_{P\times\mathbb{R}} \circ post_0(t)$.

So let $t \in T_0$. We have

$$\begin{aligned} h^{\oplus}_P \circ \pi^{\oplus}_P \circ pre_1 \circ g_T(t) &= \pi^{\oplus}_P \circ h^{\oplus}_{P\times\mathbb{R}} \circ pre_1 \circ g_T(t) = \pi^{\oplus}_P \circ pre_2 \circ h_T \circ g_T(t) \\ &= \pi^{\oplus}_P \circ pre_2 \circ f_T(t) = \pi^{\oplus}_P \circ f^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t) \\ &= f^{\oplus}_P \circ \pi^{\oplus}_P \circ pre_0(t) = (h_P \circ g_P)^{\oplus} \circ \pi^{\oplus}_P \circ pre_0(t) \\ &= h^{\oplus}_P \circ g^{\oplus}_P \circ \pi^{\oplus}_P \circ pre_0(t) = h^{\oplus}_P \circ \pi^{\oplus}_P \circ g^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t). \end{aligned}$$

Since h is injective, also $h^{\oplus}_P$ is injective which means that it is a monomorphism in **Sets**. Thus, by the equation above, we have $\pi^{\oplus}_P \circ pre_1 \circ g_T(t) = \pi^{\oplus}_P \circ g^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t)$ which means that $pre_1 \circ g_T(t)$ and $g^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t)$ have the same location.

Due to time-strictness of h, we have $pre_2 \circ h_T = h^{\oplus}_{P\times\mathbb{R}} \circ pre_1$. Moreover, for a marking $M \in P_1 \times \mathbb{R}$ with $M = \sum_{i=1}^{n}(p_i, r_i)$ we have

$$h^{\oplus}_{P\times\mathbb{R}}(M) = h^{\oplus}_{P\times\mathbb{R}}(\sum_{i=1}^{n}(p_i, r_i)) = \sum_{i=1}^{n}(h_P(p_i), r_i)$$

This implies that for all $p \in P_1$ there is

$$M[p] = h^{\oplus}_{P\times\mathbb{R}}(M[h_P(p)])$$

because h_P is injective. Thus, we obtain

$$\begin{aligned} pre_1 \circ g_T(t)[p] &= h^{\oplus}_{P\times\mathbb{R}} \circ pre_1 \circ g_T(t)[h_P(p)] = pre_2 \circ h_T \circ g_T(t)[h_P(p)] \\ &= pre_2 \circ f_T(t)[h_P(p)] \geq f^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t)[h_P(p)] \\ &= h^{\oplus}_{P\times\mathbb{R}} \circ g^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t)[h_P(p)] = g^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t)[p] \end{aligned}$$

Hence, we have $pre_1 \circ g_T(t) \overset{\leftarrow}{=} g^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t)$. The proof for post domains works analogously.

□

6.2 Category of Timed P/T Systems

Analogously to the category of timed P/T nets, we define the category of marked timed P/T nets, called timed P/T systems.

Definition 6.11 (Timed P/T Systems and Morphisms)**.** A *timed P/T system* (or *marked timed P/T net*) is a pair (TN, M) with timed P/T net $TN = (P, T, pre, post)$ and M is a marking of TN.

Given marked timed P/T-nets $MN_i = (TN_i, M_i)$, for $i \in \{1, 2\}$, a *timed P/T system morphism* (or *marked timed P/T-net-morphism*) $f : MN_1 \to MN_2$ is a timed P/T morphism $f = (f_P, f_T)$ such that:

$$f^{\oplus}_{P\times\mathbb{R}}(M_1) \overset{\leftarrow}{\leq} M_2$$

A marked timed P/T-morphism f is called *marking-strict* if f is time-strict (see Definition 6.2) and

$$f^{\oplus}_{P\times\mathbb{R}}(M_1(p)) = M_2(f_P(p)) \text{ for all } p \in P$$

Fact 6.12 (Category **TPTSys** of Timed P/T Systems)**.** The category of marked timed P/T systems, **TPTSys** consists of the class of all marked timed P/T nets as its objects, as well as timed P/T system morphisms. Composition and identity are defined by composition and identity of the respective timed P/T-morphisms, respectively nets.

Proof. For the detailed proof of Fact 6.12, we refer to Appendix B.2. □

Example 6.13 (Timed P/T System Morphism)*.* Consider the two timed P/T nets TN_1, TN_2 and the timed P/T morphism f from Example 6.4 with their respective markings shown in Figure 36, constituting the timed P/T systems (TN_1, M_1) and (TN_2, M_2).

As shown in Example 6.4, $f = (f_P, f_T)$ is a timed P/T-morphism.

The morphism condition of timed P/T-system morphisms requires that $f^{\oplus}_{P\times\mathbb{R}}(M_1) \overset{\leftarrow}{\leq} M_2$. Since

$$\begin{aligned} f^{\oplus}_{P\times\mathbb{R}}(M_1) &= f^{\oplus}_{P\times\mathbb{R}}((router3, 100) \oplus (router3, 100) \oplus (client3, 250) \oplus (client4, 200)) \\ &= (router3, 100) \oplus (router3, 100) \oplus (client34, 250) \oplus (client34, 200) \\ &= M_2, \end{aligned}$$

f is a timed P/T-system morphism.

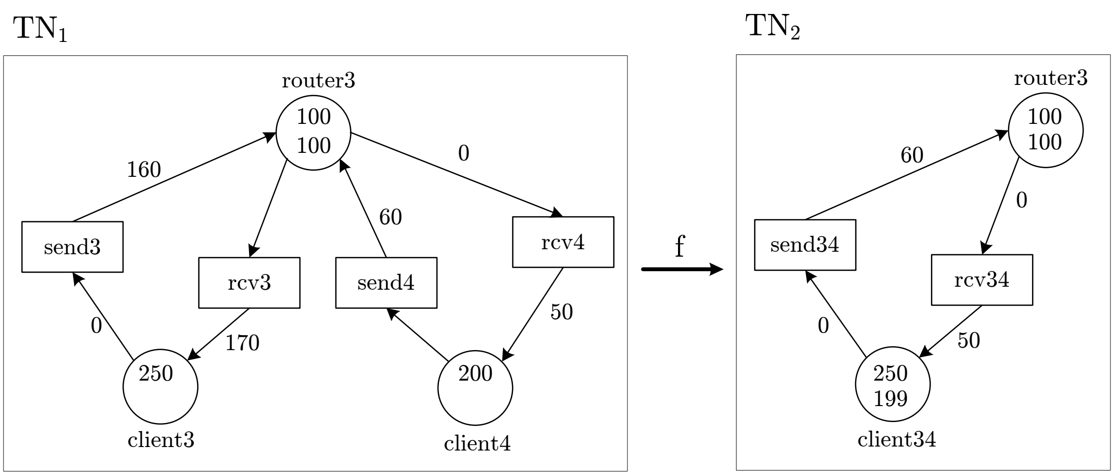


Figure 36: Timed P/T systems before firing

Theorem 6.14 (Timed P/T System Morphisms Preserve Firing Behaviour) Given timed P/T-systems $(TN_1, M_1), (TN_2, M_2)$ and a P/T-system morphism $f : (TN_1, M_1) \to (TN_2, M_2)$ with $f = (f_P, f_T)$.
Let $(TN_1, M_1) \xrightarrow{(t_1, S_1, \tau)} (TN_1, M_1')$ be a firing step with $S_1 \leq M_1$.

Then, there is a firing step $(TN_2, M_2) \xrightarrow{(f_T(t_1), S_2, \tau)} (TN_2, M_2')$ with $S_2 = f^{\oplus}_{P\times\mathbb{R}}(S_1) \downarrow M_2^*$ and $S_2 \leq M_2^* \leq M_2$ and f can be considered as a timed P/T-system morphism $f : (TN_1, M_1') \to (TN_2, M_2')$.

Proof-Idea. We prove the theorem by showing that the existence of a firing step in the original system leads to the existence of an analogue firing step in the translated system via the definitions of activation and timed P/T-system morphisms. Afterwards, we compute the follower markings and show that the morphism condition is also fulfilled for the follower markings in both nets.

For the complete proof, see Appendix B.7. □

Example 6.15 (Timed P/T System Morphisms Preserve Firing Behaviour). Consider the two timed P/T-systems (TN_1, M_1) and (TN_2, M_2) from Example 6.13. The illustration in Figure 37 shows the systems (TN_1, M_1') and (TN_2, M_2') respectively, after $rcv3$ and $f_T(rcv3) = rcv34$ have been fired at $\tau = 100$.

For f to preserve firing behaviour, f has to be able to be considered as a P/T-system morphism $f : (TN_1, M_1') \to (TN_2, M_2')$, i.e. $f^{\oplus}_{P\times\mathbb{R}}(M_1') \overset{\leftarrow}{\leq} M_2'$.

Since

$$\begin{aligned} f^{\oplus}_{P\times\mathbb{R}}(M_1') &= f^{\oplus}_{P\times\mathbb{R}}((router3, 100) \oplus (client3, 250) \oplus (client3, 270) \oplus (client4, 200)) \\ &\overset{\leftarrow}{=} (router3, 100) \oplus (client34, 250) \oplus (client34, 270) \oplus (client34, 199) \\ &= M_2', \end{aligned}$$

f is a P/T-system morphism and thus preserves firing behaviour.

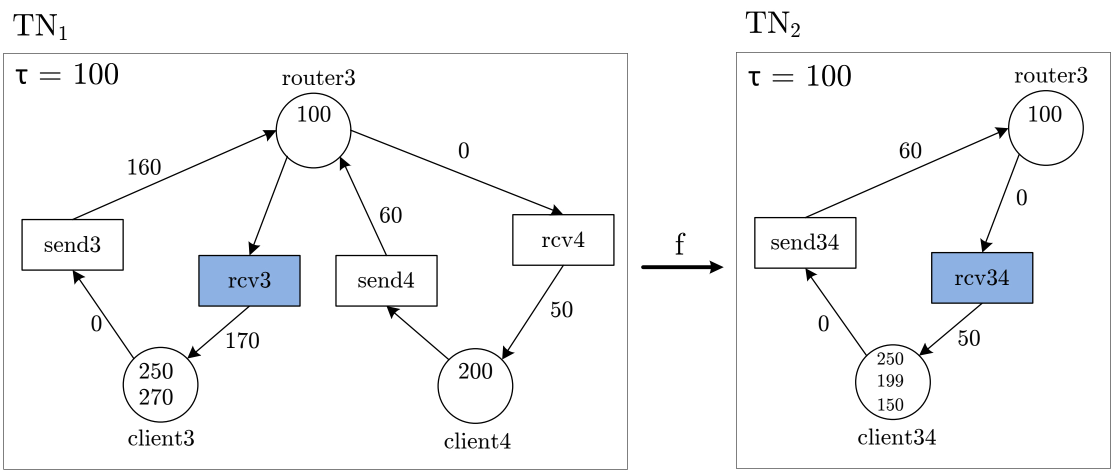


Figure 37: Timed P/T systems after firing

6.3 Category of Timed P/T States

Analogously to the category of timed P/T systems, we define the category of timed P/T states. Beforehand, we define the notion of timed P/T states.

Definition 6.16 (Timed P/T State and Morphisms)**.** A *timed P/T state* (or TPT state) is a 3-tuple (TN, M, τ) with timed P/T net $TN = (P, T, pre, post)$, a marking M of TN and a global clock value $\tau \in \mathbb{R}$.

Given timed P/T states (TN_i, M_i, τ_i), for $i = 1, 2$. A *timed P/T state morphism* f : $(TN_1, M_1, \tau_1) \to (TN_2, M_2, \tau_2)$ is a timed P/T morphism $f = (f_P, f_T)$ such that:

$$f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\Delta\tau} \overset{\leftarrow}{\leq} M_2 \text{ where } \Delta\tau = \tau_2 - \tau_1.$$

The conditions for strictness of timed P/T-system morphisms also apply to timed P/T state morphisms: A timed P/T state morphism f is strict, if it is time-strict and

$$f^{\oplus}_{P\times\mathbb{R}}(M_1(p)) = M_2(f_P(p)) \text{ for all } p \in P.$$

Fact 6.17 (Category **TPTStates** of Timed P/T States)**.** The category of timed P/T states, **TPTStates** consists of the class of all timed P/T states as its objects, as well as timed P/T state morphisms. The composition of two timed P/T state morphisms $g \circ f$ is defined componentwise as $g \circ f = ((g \circ f)_P, (g \circ f)_T) = (g_P \circ f_P, g_T \circ f_T)$. The identity morphism for each timed P/T state $A = (TN, M, \tau)$ is defined as $id_A : A \to A : id = (id_P, id_T)$.

Proof. For the detailed proof of Fact 6.17, see Appendix B.3. □

Example 6.18 (Timed P/T State Morphism)*.* Consider the two timed P/T nets TN_1, TN_2 from Figure 38 with their respective markings and clock values, constituting the timed P/T-systems (TN_1, M_1, τ_1) and (TN_2, M_2, τ_2) with $\tau_1 = 100, \tau_2 = 150$.

As shown in example 6.4, $f = (f_P, f_T)$ is a timed P/T morphism.

The timed P/T-state morphism condition requires that $f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\Delta\tau} \overset{\leftarrow}{\leq} M_2$ with $\Delta\tau = \tau_2 - \tau_1$.

We have $\Delta\tau = \tau_2 - \tau_1 = 50$ and

$$\begin{aligned} f^{\oplus}_{P\times\mathbb{R}}(M_1) &= f^{\oplus}_{P\times\mathbb{R}}((router3, 100) \oplus (router3, 100) \oplus (client3, 250) \oplus (client4, 200)) \\ &= (router3, 100) \oplus (router3, 100) \oplus (client34, 250) \oplus (client34, 200), \end{aligned}$$

so

$$f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\Delta\tau} = (router3, 150) \oplus (router3, 150) \oplus (client34, 300) \oplus (client34, 250) = M_2$$

Therefore, f is a timed P/T-state morphism.

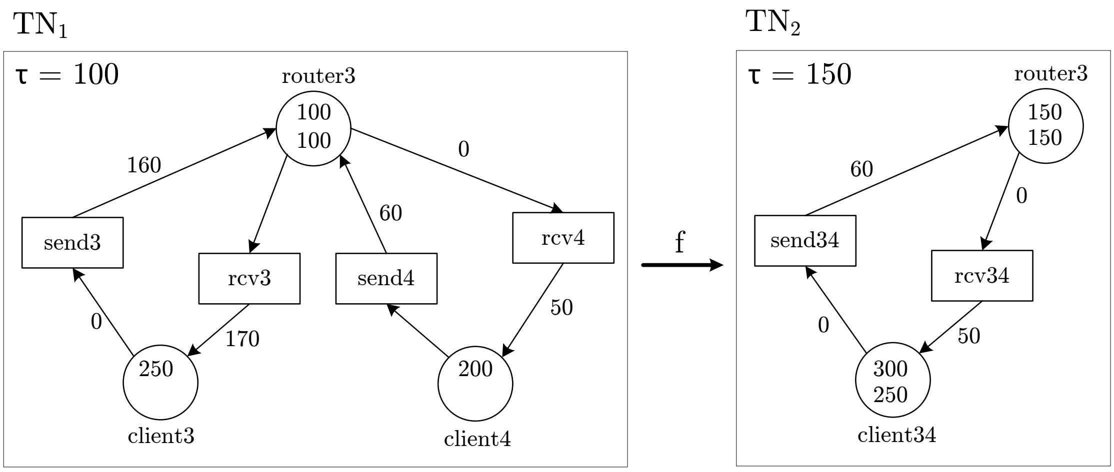


Figure 38: Timed P/T states before firing

Theorem 6.19 (Timed P/T State Morphisms Preserve Firing Behaviour) Given timed P/T-states $(TN_1, M_1, \tau_1), (TN_2, M_2, \tau_2)$ and P/T-state morphism $f : (TN_1, M_1, \tau_1) \to (TN_2, M_2, \tau_2)$ with $f = (f_P, f_T)$.
Let $(TN_1, M_1, \tau_1) \xrightarrow{(t_1, S_1, \tau_1)} (TN_1, M'_1, \tau_1)$ be a firing step with $S_1 \leq M_1$.

Then, there is a firing step $(TN_2, M_2, \tau_2) \xrightarrow{(f_T(t_1), S_2, \tau_2)} (TN_2, M'_2, \tau_2)$ with $S_2 := f^{\oplus}_{P\times\mathbb{R}}(S_1) \downarrow M^*_2$ and $S_2 \leq M^*_2 \leq M_2$ and f can be considered as a timed P/T-state morphism $f : (TN_1, M'_1, \tau_1) \to (TN_2, M'_2, \tau_2)$.

Proof-Idea. We prove the theorem by showing that the existence of a firing step in the original system leads to the existence of an analogue firing step in the translated system via the definitions of activation and timed P/T-state morphisms. Afterwards, we compute the follower markings and show that the morphism condition is also fulfilled for the follower markings in both nets.

For the complete proof, see Appendix B.8. □

Example 6.20 (Timed P/T State Morphisms Preserve Firing Behaviour). Consider the two timed P/T-states (TN_1, M_1, τ_1) and (TN_2, M_2, τ_2) from Figure 6.18. The illustration in Figure 39 shows the states (TN_1, M'_1, τ_1) and (TN_2, M'_2, τ_2) respectively, after $rcv3$ has been fired at the respective clock values $\tau_1 = 100$ and $\tau_2 = 150$.

For f to preserve firing behaviour, f has to be able to be considered as a P/T-state morphism $f : (TN_1, M'_1, \tau_1) \to (TN_2, M'_2, \tau_2)$, i.e. $f^{\oplus}_{P\times\mathbb{R}}(M'_1)^{+\Delta\tau} \overset{\leftarrow}{\leq} M'_2$ with $\Delta\tau = \tau_2 - \tau_1 = 50$.

Since

$$\begin{aligned}
f^{\oplus}_{P\times\mathbb{R}}(M'_1)^{+\Delta\tau} &= f^{\oplus}_{P\times\mathbb{R}}((router3, 100) \oplus (client3, 250) \oplus (client3, 270) \\
&\quad \oplus (client4, 200))^{+\Delta\tau} \\
&= (router3, 150) \oplus (client34, 300) \oplus (client34, 320) \oplus (client34, 250) \\
&= M'_2,
\end{aligned}$$

we have that f preserves firing behaviour.

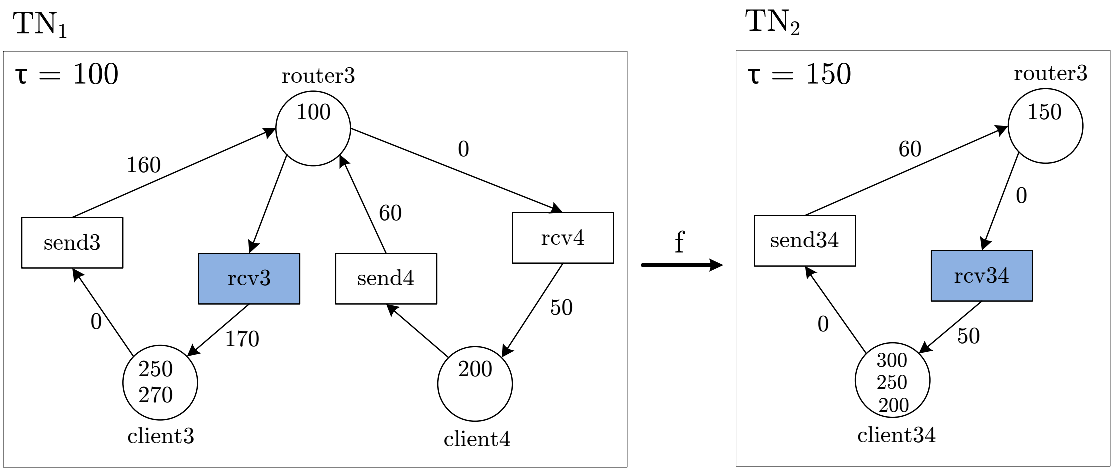


Figure 39: Timed P/T states after firing

6.4 Functorial Relations of Timed Net Classes

In this subsection, we define two functors *Rel* and *Abs* between the categories **TPTSys** and **TPTState** defined in Section 6.

First, we define the functor *Rel*, which maps timed P/T-states to timed P/T-systems (and their morphisms accordingly). This functor subtracts the global clock value of the timed P/T-state from all time stamps in the marking of the net, resulting in a timed P/T system with a marking with a relative time offset from the original marking, dependent on the clock value of the state.

Definition 6.21 (Functor *Rel*)**.** The functor *Rel* is defined as Rel : **TPTStates** $\rightarrow$ **TPTSys** with

$$Rel(TN, M, \tau) = (TN, M^{-\tau})$$

for the objects of **TPTStates** and

$$Rel(f_P, f_T) = (f_P, f_T)$$

for the morphisms.

Well-definedness.

$Rel(f)$ **is timed P/T system morphism:**

First, we have to show that for a timed P/T state morphism $f : TS \rightarrow TS'$, there is a timed P/T system morphism $Rel(f) : Rel(TS) \rightarrow Rel(TS')$. For this, since the components f_P, f_T are preserved by *Rel*, we have to show that the timed P/T system morphism condition is fulfilled. For nets $TS = (TN, M, \tau)$ and $TS' = (TN', M', \tau')$, by definition of *Rel*, we have $Rel(TS) = (TN, M^{-\tau})$ and analogously $Rel(TS') = (TN', M'^{-\tau'})$.

Since there is $f : TS \rightarrow TS'$, it holds that $f^{\oplus}_{P\times\mathbb{R}}(M)^{+\tau'-\tau} \overset{\leftarrow}{\leq} M'$. Then, we can subtract τ' from both sides, leading to $f^{\oplus}_{P\times\mathbb{R}}(M^{-\tau}) \overset{\leftarrow}{\leq} M'^{-\tau'}$. This is required by the timed P/T system morphism condition, which is therefore fulfilled.

Preservation of identity and composition:

Next, we need to show that the functor preserves identities and composition of morphisms:

Identity: $Rel(id_N) = Rel(id_P, id_T)_N = (id_P, id_T)_{Rel(N)} = id_{Rel(N)}$.

Composition: $Rel(g \circ f) = Rel(g_P \circ f_P, g_T \circ f_T) = (g_P \circ f_P, g_T \circ f_T) = (g_P, g_T) \circ (f_P, f_T) = Rel(g_P, g_T) \circ Rel(f_P, f_T)$

□

Analogously, the functor Abs maps timed P/T-systems and -morphisms to timed P/T-states and -morphisms. This functor simply retains the net and marking and adds the absolute global clock value of 0 to obtain a timed P/T state.

Definition 6.22 (Functor Abs)**.** The functor Abs is defined as Abs : **TPTSys** → **TPTStates** with

$$Abs(TN, M) = (TN, M, 0)$$

for the objects of **TPTSys** and

$$Abs(f_P, f_T) = (f_P, f_T)$$

for the morphisms, respectively.

Well-definedness.

$Abs(f)$ **is timed P/T state morphism:**

Again, we first have to show that for a timed P/T system morphism $f : TS \to TS'$, there is a timed P/T state morphism $Abs(f) : Abs(TS) \to Abs(TS')$. For this, since the components f_P, f_T are preserved by Abs, we have to show that the timed P/T state morphism condition is fulfilled. For nets $TS = (TN, M)$ and $TS' = (TN', M')$, by definition of Rel, we have $Rel(TS) = (TN, M, 0)$ and analogously $Rel(TS') = (TN', M', 0)$.

Since there is $f : TS \to TS'$, it holds that $f^{\oplus}_{P\times\mathbb{R}}(M) \overset{\leftarrow}{\leq} M'$. Since $\tau = \tau' = 0$, we have $M^{+\tau'-\tau} = M^{+0-0} = M$, and thus $f^{\oplus}_{P\times\mathbb{R}}(M)^{+\tau'-\tau} \overset{\leftarrow}{\leq} M'$ also holds. This is required by the timed P/T state morphism condition, which is therefore fulfilled.

Preservation of identity and composition:

Next, we need to show that the functor preserves identities and composition of morphisms:

Identity: $Abs(id_N) = Abs(id_P, id_T)_N = (id_P, id_T)_{Abs(N)} = id_{Abs(N)}$.

Composition: $Abs(g \circ f) = Abs(g_P \circ f_P, g_T \circ f_T) = Abs(g_P \circ f_P) \circ Abs(g_T \circ f_T) = Abs(g) \circ Abs(f)$.

□

We now show that the categories **TPTStates** and **TPTSys** are equivalent, i.e. there is a relation between the two categories that indicates that they are essentially the same.

Theorem 6.23 (Equivalence of Categories **TPTStates** and **TPTSys**) The categories **TPTStates** and **TPTSys** are equivalent.

For the definition of category equivalence, we refer to Definition A.9 in Appendix A.

Proof. We have to show that $Rel \circ Abs \cong Id_{TPTSys}$ and $Abs \circ Rel \cong Id_{TPTStates}$.

$Rel \circ Abs \cong Id_{TPTSys}$**:** For objects (TN, M, τ), we have

$$Rel(Abs(TN, M)) = Rel(TN, M, 0) = (TN, M^{-0}) = (TN, M).$$

For morphisms $f = (f_P, f_T)$, we have that

$$Rel(Abs(f)) = Rel(Abs(f_P, f_T)) = Rel(f_P, f_T) = (f_P, f_T) = f.$$

Therefore, $Rel \circ Abs = Id_{TPTSys}$, which implies that $Rel \circ Abs \cong Id_{TPTSys}$.

$Abs \circ Rel \cong Id_{TPTStates}$**:** For objects, we have

$$Abs(Rel(TN, M, \tau)) = Abs(TN, M^{-\tau}) = (TN, M^{-\tau}, 0)$$

and for morphisms $f = (f_P, f_T)$ that

$$Abs(Rel(f)) = Abs(Rel(f_P, f_T)) = Abs(f_P, f_T) = (f_P, f_T) = f.$$

We have to show that there is a natural transformation $\alpha : Id_{TPTStates} \to Abs \circ Rel$ that is an isomorphism. So we have to show that for all TPT states $TS = (TN, M, \tau)$ there is a TPT-state morphism $\alpha_{TS} : TS \to Abs \circ Rel(TS)$ that is an isomorphism.

From the definitions of Abs and Rel follows

$$Abs \circ Rel(TN, M, \tau) = (TN, M^{-\tau}, 0).$$

Then, there exists a morphism $\alpha : (TN, M, \tau) \to (TN, M^{-\tau}, 0)$ with $\alpha = (id_P, id_T)$ for which the timed P/T state morphism condition is fulfilled:

$$\alpha^{\oplus}_{P\times\mathbb{R}}(M)^{+\Delta\tau} = M^{-\tau} \text{ with } \Delta\tau = 0 - \tau = -\tau.$$

Then, there exists a morphism $\beta : (TN, M^{-\tau}, 0) \to (TN, M, \tau)$ with $\beta = (id_P, id_T)$. For β, the morphism condition is also fulfilled:

$$\beta^{\oplus}_{P\times\mathbb{R}}(M)^{-\tau+\Delta\tau} = M \text{ with } \Delta\tau = \tau - 0 = \tau.$$

Thus, α is an isomorphism, hence $Abs \circ Rel \cong Id_{TPTState}$.

Therefore, the categories **TPTStates** and **TPTSys** are equivalent. □

Example 6.24. Figure 40 shows the application of $Rel \circ Abs$ and $Abs \circ Rel$, respectively. For the former case, the resulting timed P/T system is identical to the original system, with the intermediary timed P/T state being different only in the contained clock value of $\tau = 0$.

For the latter case, there are morphisms from $TState$ to $TState'$ and vice versa, since they only differ in the global time offset of 150 time units. Therefore the morphism condition

$$f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\Delta\tau} \overset{\leftarrow}{\leq} M_2 \text{ where } \Delta\tau = \tau_2 - \tau_1$$

is fulfilled for both morphisms.

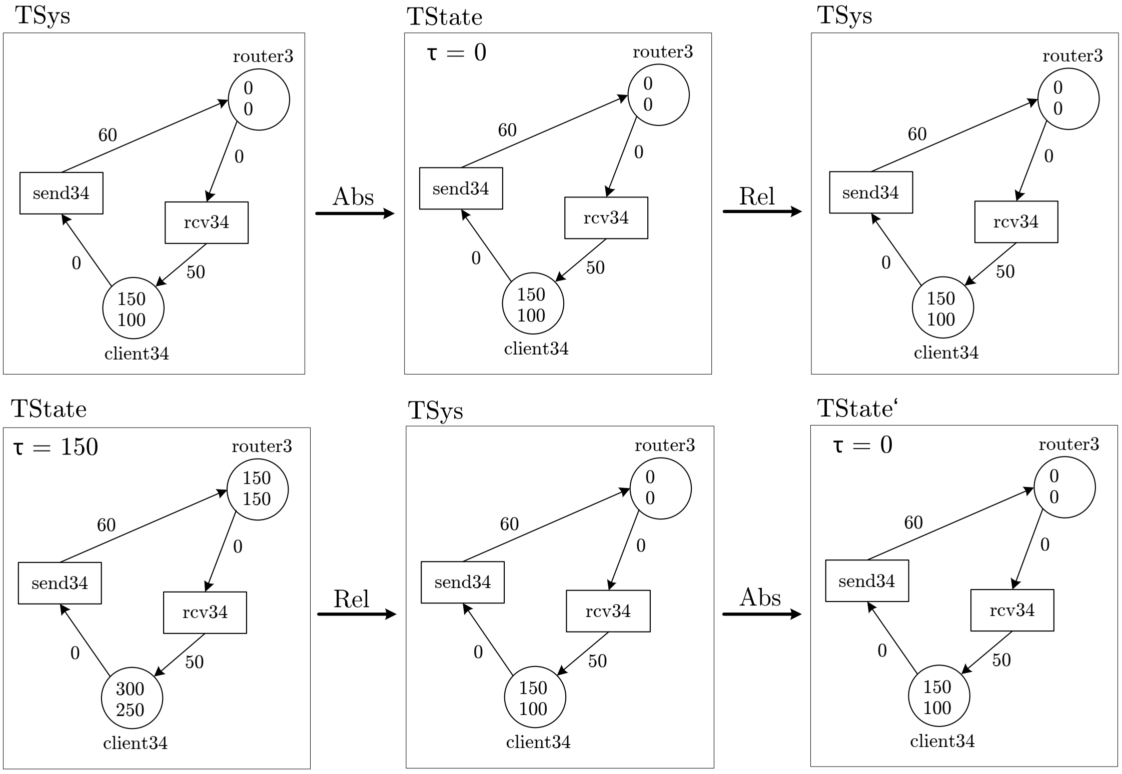


Figure 40: Equivalence of **TPTStates** and **TPTSys**

Remark 6.25 (Normalisation of Timed States). The composition of $Abs \circ Rel$ provides a way to normalise a timed state to the global clock value of zero. When applying the composition of the two functors to a timed state, we obtain a new timed state where all timestamps in the marking are reduced by the clock value of the original state. This way, all timed states that are only different by a time offset can be mapped to their respective "normal form".

Example 6.26 (Normalisation of Timed States). Figure 41 shows two timed states (TS_1 and TS_2) that are normalised to the same timed state (TS_3), their normal form.

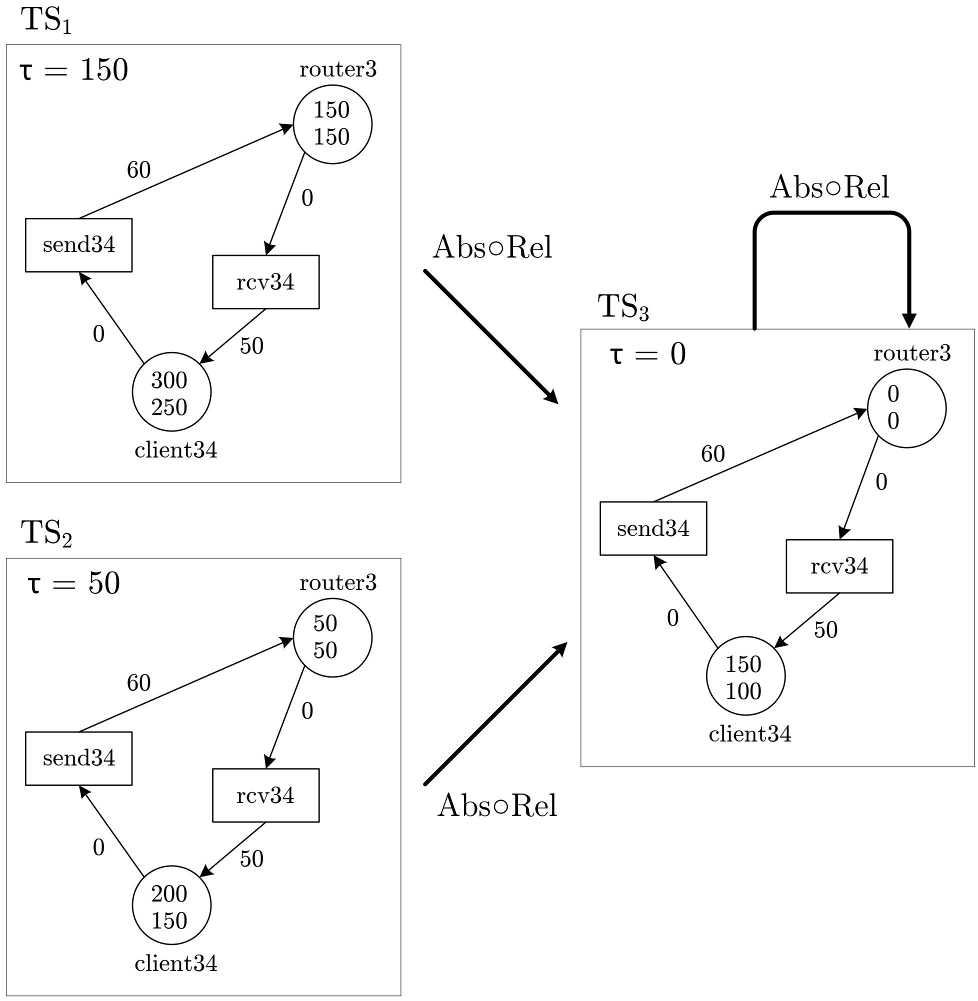


Figure 41: Normalised timed states

6.5 Functorial Relations to Untimed Net Classes

Here, we define functors *TSkel* and *TSkelSys*, which map timed P/T nets and timed P/T systems to P/T nets and P/T systems (as well as their morphisms), respectively. These functors remove all information regarding time-stamps from the nets, resulting in timed P/T nets and systems without any time-values, but retaining the locations of markings as well as pre-/post domains of transitions.

We also show that both functors preserve firing behaviour, which means that a firing step in the translated timed P/T nets and systems indicate the existence of a firing step in the respective resulting P/T nets and systems.

Definition 6.27 (Skeleton Functor *TSkel*)**.** The functor $TSkel : \mathbf{TPTNets} \to \mathbf{PTNets}$ is defined by $TSkel(P, T, pre, post) = (P, T, pre^*, post^*)$, where $pre^*(t) = \pi_P^{\oplus}(pre(t))$, $post^*(t) = \pi_P^{\oplus}(post(t))$ for all $t \in T$ for the objects of **TPTStates**. For the morphisms, we define $TSkel(f_P, f_T) = (f_P, f_T)$.

Well-definedness. Given timed P/T nets $TN_1 = (P_1, T_1, pre_1, post_1)$ and

$TN_2 = (P_2, T_2, pre_2, post_2)$ and timed P/T morphism $f : TN_1 \to TN_2$. Since the locations of the pre- and post domains of any transition t in TN_1 and $f_T(t)$ in TN_2 are the same, i.e. $\pi_P^{\oplus}(pre_1(t)) = f_P^{\oplus}(\pi_P^{\oplus}(pre_2(f_T(t))))$ (analogous for post), the P/T system morphism condition of $TSkel(f)$ for any timed P/T morphism f is satisfied:

$$pre_2^*(f_T(t)) = \pi_P^{\oplus}(f_T(pre_2(t))) = f_P^{\oplus}(\pi_P^{\oplus}(pre_1(t))) = f_P^{\oplus}(pre_1^*(t))$$

Therefore, for all objects A, B in **TPTNets** with morphism $f : A \to B$, a morphism $TSkel(f) : TSkel(A) \to TSkel(B)$ exists in **PTNets**. The preservation of identities and composition follows directly from the definition of the morphism component of the functor. □

Example 6.28 (Skeleton Functor *TSkel*). Figure 42 shows timed P/T nets TN, TN' with morphism $f : TN \to TN'$, as well as the P/T nets obtained from applying *TSkel* to the nets and morphism. Note that the edge inscriptions in the timed P/T nets are sums of time values, while in the regular P/T nets, they denote the amount of tokens created or consumed. The transitions *rcv*3 and *rcv*4 each create two tokens in both net variants. The morphism condition for regular P/T nets is also fulfilled, so $f : N \to N'$ exists.

Fact 6.29 (Functor *TSkel* Preserves Firing Behaviour)**.** Given timed P/T net $TN = (P, T, pre, post)$, marking M of TN and a selection $S \leq M$, a transition $t \in T$ and a clock value $\tau \in \mathbb{R}$. Let $TSkel(TN) = N^*$ with $N^* = (P, T, pre^*, post^*)$. Then for every $M^* \geq \pi_P^{\oplus}(M)$ with a firing step $M \xrightarrow{t,S,\tau} M'$ in TN, there is also a firing step $M^* \xrightarrow{t} M'^*$ in N^*.

Proof. t is activated in TN under S at time τ, i.e. $t \in T$ activated under S means that $S \stackrel{\rightrightarrows}{=} pre(t)^{+\tau}$. Since $S \leq M$, we have $M \geq S \stackrel{\rightrightarrows}{=} pre(t)^{+\tau}$. Then,

$$\begin{aligned} S \stackrel{\rightrightarrows}{=} pre(t)^{+\tau} &\Rightarrow \pi_P^{\oplus}(S) = pre^*(t)^{+\tau} \Rightarrow \pi_P^{\oplus}(S) \leq \pi_P^{\oplus}(M) \\ &\Rightarrow pre^*(t)^{+\tau} \leq \pi_P^{\oplus}(M) \leq M^* \end{aligned}$$

Therefore, $M^* \geq pre^*(t^*)$, so t is activated in TN^* and there is a firing step $M^* \xrightarrow{t} M'^*$ in TN^*. □

Definition 6.30 (Skeleton Functor *TSkelSys*)**.** The functor *TSkelSys* is defined as *TSkelSys* : **TPTSys** $\to$ **PTSys** with $TSkelSys(TN, M) = (N^*, M^*)$, where $N^* = (P, T, pre^*, post^*)$ with $pre^*(t) = \pi_P^{\oplus}(pre(t))$, $post^*(t) = \pi_P^{\oplus}(M)$ for all $t \in T$ and $M^* = \pi_P^{\oplus}(post(t))$ for the objects of **TPTSys**. For the morphisms, we define $Rel(f_P, f_T) = (f_P, f_T)$.

Well-definedness. Given timed P/T systems (TN_1, M_1), (TN_2, M_2), with $TN_i = (P_i, T_i, pre_i, post_i)$ for $i = 1..2$ and timed P/T system morphism $f : TN_1 \to TN_2$. Since the locations of the pre- and post domains of any transition t in TN and $f_T(t)$ in TN' is the same, i.e. $\pi_P^{\oplus}(pre_1(t)) = f_P^{\oplus}(\pi_P^{\oplus}(pre_2(f_T(t))))$ (analogous for post), the P/T system morphism condition of $TSkelSys(f)$ for any timed P/T-system morphism f is satisfied:

$$pre_2^*(f_T(t)) = \pi_P^{\oplus}(f_T(pre_2(t))) = f_P^{\oplus}(\pi_P^{\oplus}(pre_1(t))) = f_P^{\oplus}(pre_1^*(t))$$

The same is true for the Marking, since the location is the same: $f_{P\times\mathbb{R}}^{\oplus}(M_1) = (M_2)$.

Therefore, for all objects A, B in **TPTSys** with morphism $f : A \to B$, a morphism $TSkel(f) : TSkel(A) \to TSkel(B)$ exists in **PTSys**. The preservation of identities and composition follows directly from the definition of the morphism component of the functor. □

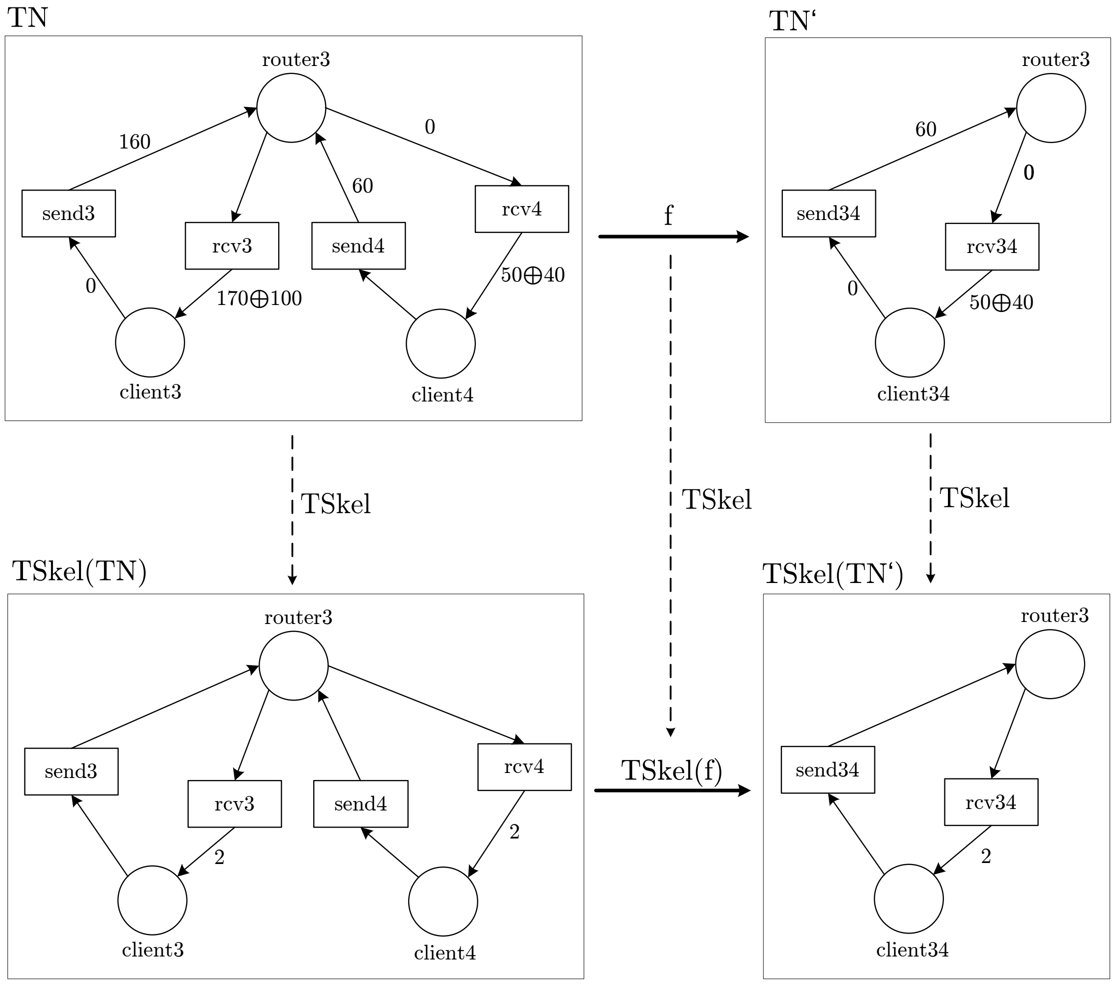


Figure 42: Functor TSkel

Example 6.31 (Skeleton Functor *TSkelSys*)*.* Figure 43 shows timed P/T systems $TSys, TSys'$ with morphism $f : TSys \to TSys'$, as well as the P/T systems obtained from applying *TSkelSys* to the nets and morphism. Again, the edge inscriptions in the timed P/T systems are sums of time values, while in the regular P/T systems, they denote the amount of tokens created or consumed. The markings are preserved as black tokens. The morphism condition for regular P/T nets is also fulfilled, so $f : N \to N'$ exists.

Fact 6.32 (Functor *TSkelSys* Preserves Firing Behaviour)**.** Given timed P/T-system (TN, M) with $TN = (P, T, pre, post)$, marking M of TN and a selection S of M, a transition $t \in T$ and a clock value $\tau \in \mathbb{R}$. Let $TSkelSys(TN, M) = (N^*, M^*)$ with $N^* = (P, T, pre^*, post^*)$. If there is a firing step $(TN, M) \xrightarrow{t,S,\tau} (TN', M')$ in TN, there is also a firing step $(N^*, M^*) \xrightarrow{t} (N'^*, M'^*)$ in N^* .

Proof. Since the only difference between timed P/T systems and timed P/T nets is the explicit marking contained in the system, the proof is analogous to Proof 6.5. □

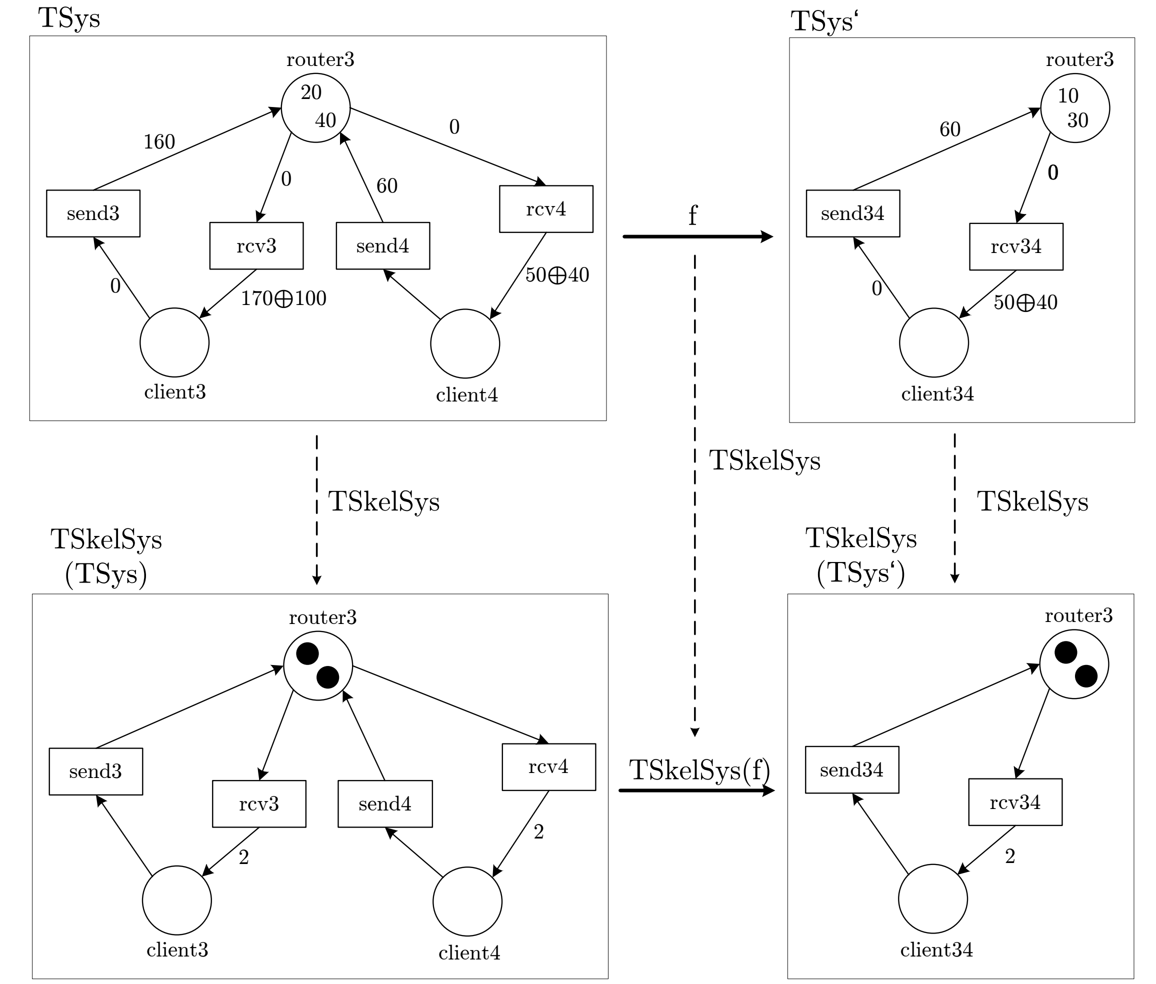


Figure 43: Example mapping of functor $TSkelSys$

Example 6.33 (*TSkelSys* Preserves Firing Behaviour). Figure 44 shows the timed P/T systems $TSys = (TN, M), TSys' = (TN, M')$, representing the same timed P/T net, with the marking M' being the marking after $rcv34$ has been fired at global clock value 10 with selection $S = (router3, 10)$. Analogously, the corresponding P/T systems $TSkelSys(TSys)$, $TSkelSys(TSys')$ are shown, which are obtained by applying *TSkelSys* to both timed P/T systems. The resulting system $TSkelSys(TSys)$ can also fire with the translated marking, which results in the system $TSkelSys(TSys')$.

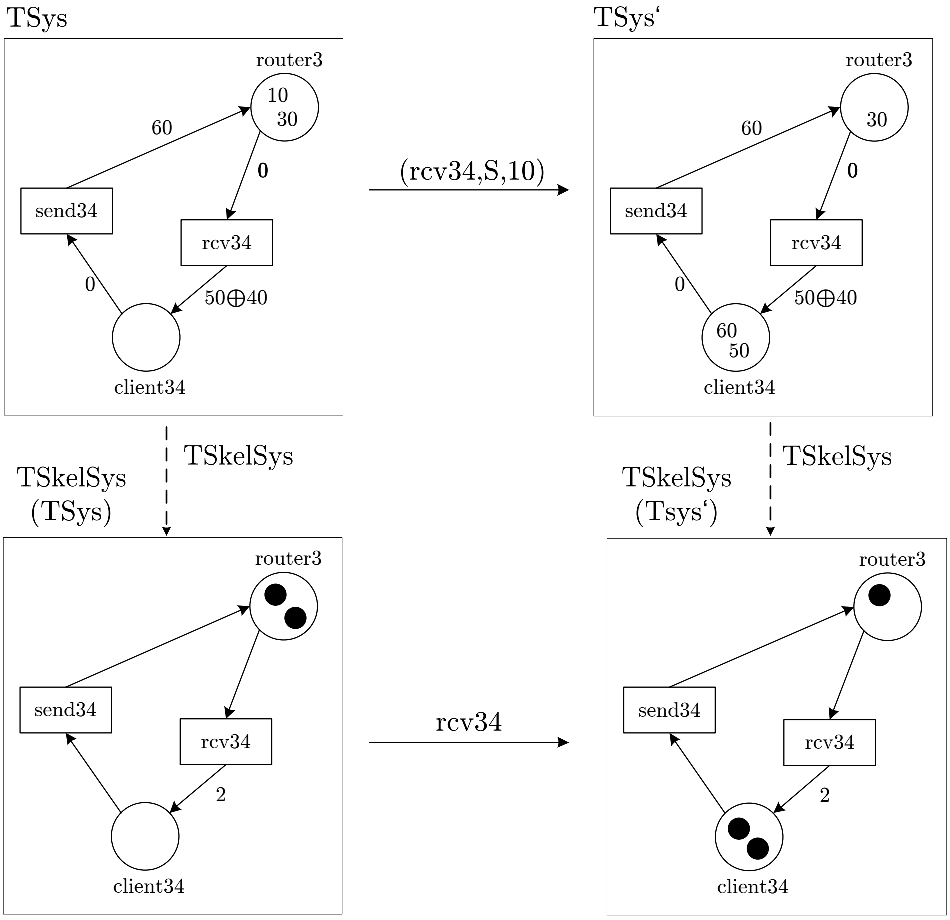


Figure 44: Preservation of Firing Behaviour by TSkelSys

7 Structuring Techniques for Timed P/T Nets

In this section, we define three structuring techniques for timed P/T nets: union, fusion and restriction. These are based on the categorical constructs of pushouts, coequalisers, and pullbacks, respectively.

Union and fusion are both means of structuring timed P/T nets by identifying parts of one (in the case of fusion) or more (in the case of union) timed P/T nets. Both are defined in [EHKP91b, EHKP91a, PPE+05] for algebraic P/T nets, based on pushouts and coequalisers (see also Section 4). We provide similar definitions, with adaptions respecting the definition of timed P/T nets. Further, the restriction of a timed P/T net is a structuring technique that allows to restrict a timed P/T morphism (and especially its domain) to a given subnet of its codomain.

Moreover, we show that the pushouts and pullbacks of timed P/T nets—computed as union and restriction of timed P/T nets, respectively—are compatible with each other in the sense of the vertical van Kampen property, leading to an $\mathcal{M}$-adhesive category [EGH10] of timed P/T nets.

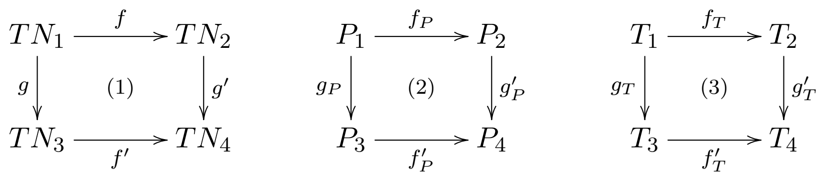
7.1 Union of Timed P/T Nets

Union allows obtaining one net from two single nets, identifying certain places and transitions in the union object, determined by the so-called interface net, with morphisms mapping its places and transitions to those of the nets to unify.

First, we define the construction of a gluing in the category **TPTNets**, which yields a union object when applied to two nets with an interface. We then show that the gluing construction is a pushout in **TPTNets**. The definitions and proofs are analogous to those for algebraic P/T nets, however the presence of time values in the pre-/post domains require certain additional prerequisites.

For the pushouts, we restrict the definition to pushouts along one time-strict, injective morphism. Even with this constraint, the union is still usable as a structuring technique, since it still allows for unification of places and transitions.

In comparison to a general union, this variant places some restrictions on the time values in the pre-/post domains due to the time-strict morphism.



(a) Pushout in **TPTNets**

(b) Pushout of places in **Sets**

(c) Pushout of transitions in **Sets**

Figure 45: Pushouts of timed P/T nets, places and transitions

Definition 7.1 (Gluing of Timed P/T Nets)**.** Given timed P/T nets $TN_i = (P_i, T_i, pre_i, post_i)$, with $i = 1 \ldots 3$, injective and time-strict timed P/T morphism $f : TN_1 \to TN_2$ and timed P/T morphism $g : TN_1 \to TN_3$. Then, the gluing $TN_4 = (P_4, T_4, pre_4, post_4)$ of TN_2 and TN_3 along f and g, written $TN_4 = TN_2 +_{TN_1,f,g} TN_3$ with morphisms $f' : TN_3 \to TN_4, g' : TN_2 \to TN_4$ is constructed as follows:

- P_4 is constructed as pushout in **Sets**, as depicted in Figure 45b,
- T_4 is constructed as pushout in **Sets**, as depicted in Figure 45c,
- $pre_4(t) = \begin{cases} f'^{\oplus}_{P\times\mathbb{R}}(pre_3(t^*)) & \text{if } \exists t^* \in T_3, f'_T(t^*) = t \\ g'^{\oplus}_{P\times\mathbb{R}}(pre_2(t')) & \text{if } \nexists t^* \in T_3, f'_T(t^*) = t \wedge \exists t' \in T_2 : g'_T(t') = t, \text{ and} \end{cases}$
- $post_4(t) = \begin{cases} f'^{\oplus}_{P\times\mathbb{R}}(post_3(t^*)) & \text{if } \exists t^* \in T_3, f'_T(t^*) = t \\ g'^{\oplus}_{P\times\mathbb{R}}(post_2(t')) & \text{if } \nexists t^* \in T_3, f'_T(t^*) = t \wedge \exists t' \in T_2 : g'_T(t') = t \end{cases}$

Well-definedness of f', g', TN_4: To show: f', g' are well-defined timed P/T morphisms. Since f is injective, f' is also injective. Given transition $t \in T_4$. For $pre_4(t)$, we either have $pre_4(t) = g'^{\oplus}_{P\times\mathbb{R}}(pre_2(t'))$ with $t' \in T_2, g'_T(t') = t$ or $pre_4(t) = f'^{\oplus}_{P\times\mathbb{R}}(pre_3(t'))$ with $t' \in T_2, g'_T(t') = t$. Since f is time-strict, from the timed P/T morphism condition follows $f'^{\oplus}_{P\times\mathbb{R}} \circ g^{\oplus}_{P\times\mathbb{R}} \circ pre_1(t_1) \overset{\leftarrow}{=} g'^{\oplus}_{P\times\mathbb{R}} \circ f^{\oplus}_{P\times\mathbb{R}} \circ pre_1(t_1)$ for all $t_1 \in T_1$. Therefore, the morphism condition for f' is satisfied, since $pre_4(t) = f'^{\oplus}_{P\times\mathbb{R}}(pre_3(t'))$, with $f'_T(t') = t, t' \in T_3$. In the remaining case, $pre_4(t) = g'^{\oplus}_{P\times\mathbb{R}}(pre_2(t'))$, with $g'_T(t') = t, t' \in T_2$. For post, the proof is analogous.

For the well-definedness of TN_4, since the cases in the definitions of pre_4 and $post_4$ are mutually exclusive, f' and g' are jointly surjective. Therefore TN_4 is well-defined.

□

Fact 7.2 (Gluing of Timed P/T Nets is Pushout)**.** Given a gluing $TN_4 = TN_2 +_{TN_1,f,g} TN_3$ with $TN_i = (P_i, T_i, pre_i, post_i)$, injective and time-strict timed P/T morphism $f : TN_1 \to TN_2$ and timed P/T morphism $g : TN_1 \to TN_3$ and morphisms $f' : TN_3 \to TN_4, g' : TN_2 \to TN_4$. Then, the gluing object TN_4 is a pushout of TN_2 and TN_3 along TN_1 in **TPTNets**.

Proof-Idea. We show that the universal property is fulfilled by the gluing construction by showing that the comparison morphism induced by the pushout construction is a well-defined timed P/T morphism, and that it is unique. For the complete proof, see Appendix B.9. □

Example 7.3 (Union of Timed P/T Nets). Figure 46 shows a union of two timed nets TN_2, TN_3. TN_1 serves as the interface, with $f_{1P}(p1) = p1$, $f_{1P}(p2) = p2$ and $f_{2P}(p1) = f_{2P}(p2) = p1, 2$. This results in the two nets being unified in $TN4$ with $p1, 2$ being the unification place. The non-injective mapping of $p1$ and $p2$ by f_2 results in $p1$ and $p2$ from TN_2 being glued together, resulting in the place $p1, 2$.

In **Sets**, monomorphisms are closed under pushout, meaning that in the following diagram, if f_T is a monomorphism, so is f'_T.

$$\begin{array}{ccc} A & \xrightarrow{f_T} & B \\ {\scriptstyle g_T}\downarrow & (1) & \downarrow{\scriptstyle g'_T} \\ C & \xrightarrow[f'_T]{} & D \end{array}$$

We show that in the category of timed P/T nets, this also holds true for time-strict injective morphisms, i.e. in a pushout square, the morphism opposite of a time-strict injective morphism is also time-strict and injective.

Fact 7.4 (Time-Strict Injective Morphisms are Closed under Pushouts)**.** Given the pushout of timed P/T nets in Figure 45a. If (1) is a pushout and $f = (f_P, f_T)$ is time-strict and injective, then $f' = (f'_P, f'_T)$ is time-strict and injective also.

Proof. By Fact 7.2, a pushout along time-strict injective morphism f is a gluing as defined in Definition 7.1. Therefore, we have pushouts (2),(3) (as shown in Figures 45b,45c) in **Sets** by definition of pushouts in Definition 7.1. Since f_P, f_T are injective, and monomorphisms are closed under pushout in **Sets**, f'_P and f'_T are injective as well, therefore f' is injective.

It remains to show that for all $t \in T_3$, we have $pre_4(f'_T(t)) = f^{\oplus}_{P\times\mathbb{R}}(pre_3(t))$ and $post_4(f'_T(t)) = f^{\oplus}_{P\times\mathbb{R}}(post_3(t))$ respectively, which follows directly from the gluing construction in Definition 7.1. □

In order to show that the category **TPTNets** has binary coproducts, which is the categorical equivalent of the disjoint union, we first show that it has initial objects, which, if used as the interface object of a, pushout results in a binary coproduct.

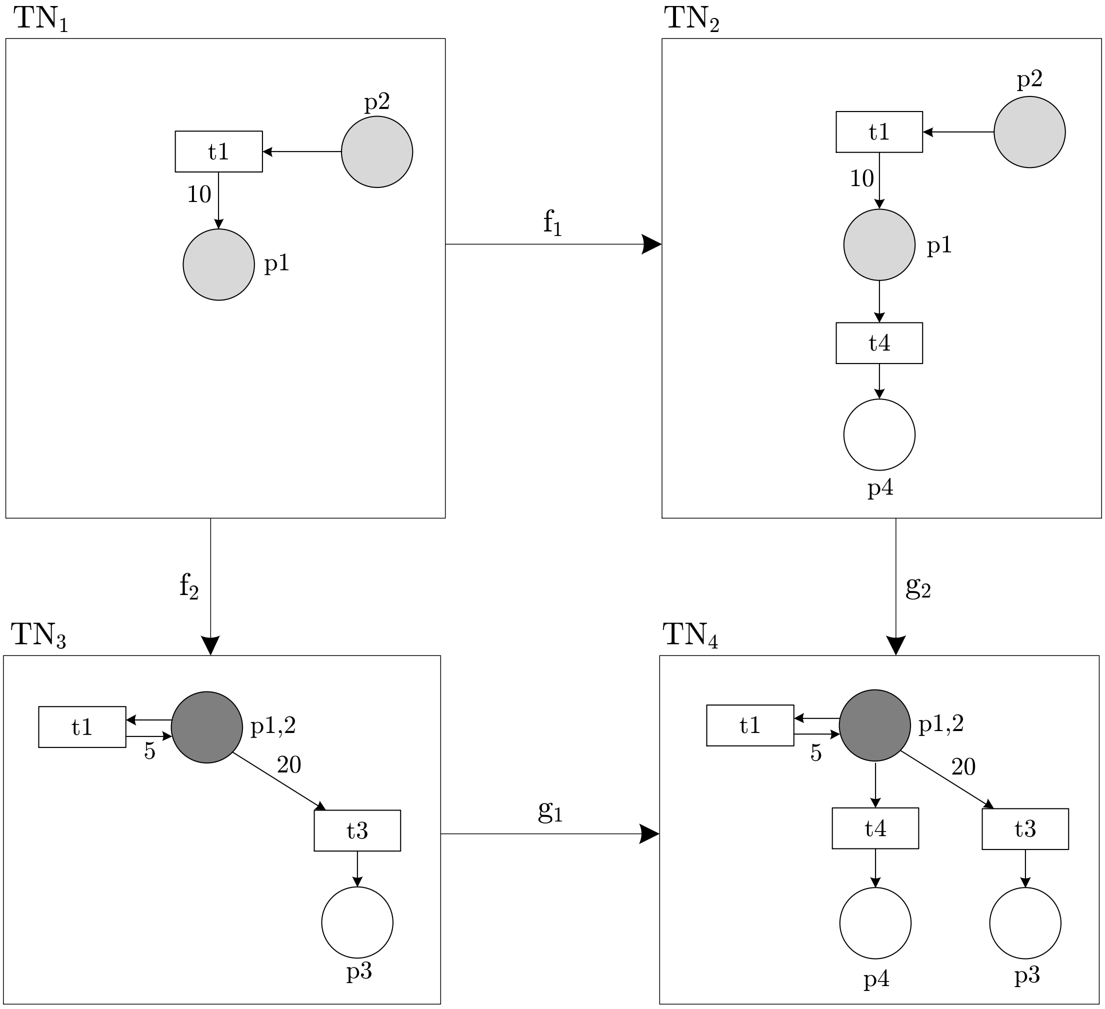


Figure 46: Union of Timed P/T Nets

Fact 7.5 (**TPTNets** has Initial Objects)**.** The empty timed P/T net, $E = (P_E, T_E, pre_E, post_E)$ with $P_E = \emptyset, T_E = \emptyset, pre_E : \emptyset \to \emptyset, post_E : \emptyset \to \emptyset$, is initial object in **TPTNets**. Moreover, the induced morphism $e : E \to TN$ for every timed net TN is time-strict.

Proof. For any timed P/T net $TN = (P, T, pre, post)$, there is exactly one morphism $e : E \to TN$ with $e = (e_P, e_T), e_P : \emptyset \to P$, $e_T : \emptyset \to T$, namely the empty morphism. Now, e_P, e_T are the unique functions in **Sets** induced by initial object $\emptyset$, therefore e is the unique morphism $E \to TN$ for all timed nets TN. The morphism condition is satisfied for the empty morphism because there are no transitions that can violate the condition, therefore it is well-defined. For the same reason, it is also time-strict. □

Fact 7.6 (**TPTNets** has Binary Coproducts)**.** The category **TPTNets** has binary coproducts, i.e. for two timed P/T nets $TN_x = (P_x, T_x, pre_x, post_x)$ for $x = 1, 2$, there is a timed P/T net C with morphisms $i_1 : TN_1 \to C$, $i_2 : TN_2 \to C$ such that for all nets D with morphisms $j_1 : TN_1 \to D$, $j_2 : TN_2 \to D$, there exists a unique morphism $h : C \to D$ with $h \circ i_1 = j_1$ and $h \circ i_2 = j_2$.

Proof. Since **TPTNets** has initial objects (Fact 7.5), pushouts along time-strict injective morphisms (Fact 7.2) and furthermore the empty morphism is time-strict and injective, **TPTNets** has binary coproducts, which can be computed as pushout over the initial object. □

Example 7.7 (**TPTNets** has Binary Coproducts). Figure 47 shows the coproduct C of timed P/T nets TN_1, TN_2 with inclusion morphisms i_1, i_2. Note that no gluing takes place, even though both TN_1 and TN_2 have a place named $router1$. Instead, $i_1(router1) = router1$ and $i_2(router1) = router1'$.

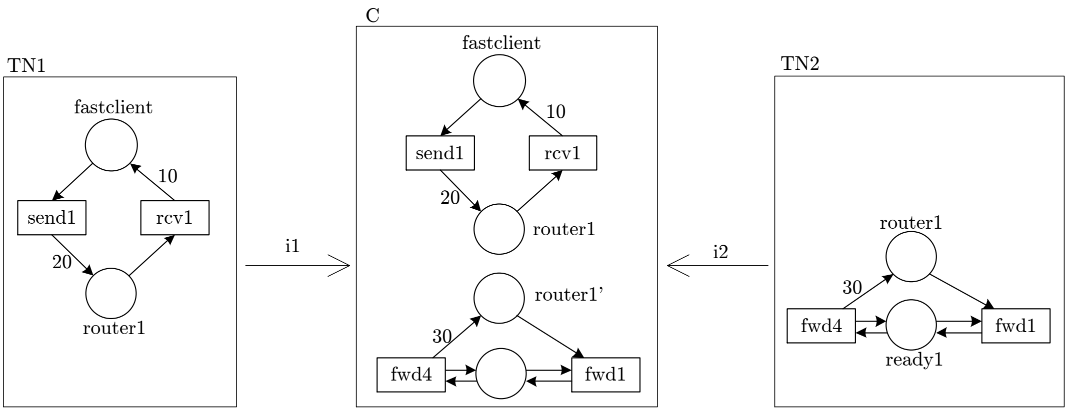


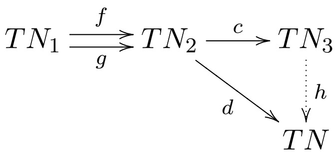
Figure 47: Coproduct of Timed P/T Nets

7.2 Fusion of Timed P/T Nets

Fusion allows the identification of two or more places (or transitions) in a net. This is achieved by using one interface net with two morphisms. The elements that are mapped equally by different elements in the interface net become unified in the fusion net.

First, we define the construction of the fusion of timed P/T nets. Like with union, we restrict fusions to time-strict injective morphisms. We then show that the fusion is a coequaliser in the category of timed P/T nets.

Definition 7.8 (Fusion of Timed P/T Nets)**.** Given time-strict injective timed P/T morphisms $f, g : TN_1 \to TN_2$ with $TN_i = (P_i, T_i, pre_i, post_i)$.



Then, the fusion object TN_3 of f and g with morphism $c : TN_2 \to TN_3$ are constructed as follows:

- (P_3, c_P) is coequaliser of f_P, g_P in **Sets**,
- (T_3, c_T) is coequaliser of f_T, g_T in **Sets**,

- $pre_3(t) = c^{\oplus}_{P\times\mathbb{R}}(pre_2(t'))$, with $c_T(t') = t$ for all $t' \in TN_2$, and
- $post_3(t) = c^{\oplus}_{P\times\mathbb{R}}(post_2(t'))$, with $c_T(t') = t$ for all $t' \in TN_2$.

Well-definedness.

Well-definedness of TN_3: To show: $pre_3, post_3$ are functions.

1 For all $t \in T_3$, there is $M \in (P \times \mathbb{R})^{\oplus}$ with $pre(t) = M$.
Since c_P, c_T are coequalisers in **Sets**, these functions are epimorphisms, i. e. they are surjective functions. Therefore, for all $t \in T_3$, there exists $t' \in T_2$ with $c_T(t') = t$, and we have $pre_3(t) = c^{\oplus}_{P\times\mathbb{R}}(pre_2(t'))$.

2 It remains to show that the result of $pre_3(t)$ is unique for every $t \in T_3$. Thus, we have to show that for $t_1, t_2 \in T_2$ with $c_T(t_1) = c_T(t_2)$, we have $c^{\oplus}_{P\times\mathbb{R}}(pre_2(t_1)) = c^{\oplus}_{P\times\mathbb{R}}(pre_2(t_2))$.
Let $t_1, t_2 \in T_2$ with $c_T(t_1) = c_T(t_2)$. By construction of coequalisers in **Sets**, there is $t_0 \in T_1$ with $f_T(t_0) = t_1$ and $g_T(t_0) = t_2$. Since f,g are time-strict and injective, we obtain

$$
\begin{aligned}
&c^{\oplus}_{P\times\mathbb{R}}(pre_2(t_1)) = c^{\oplus}_{P\times\mathbb{R}}(pre_2(f_T(t_0))) = c^{\oplus}_{P\times\mathbb{R}}(f^{\oplus}_{P\times\mathbb{R}}(pre_1(t_0))) \\
&= (c_P \circ f_P)^{\oplus}_{\times\mathbb{R}}(pre_1(t_0)) \overset{Coeq.}{=} (c_P \circ g_P)^{\oplus}_{\times\mathbb{R}}(pre_1(t_0)) \\
&= c^{\oplus}_{P\times\mathbb{R}}(g^{\oplus}_{P\times\mathbb{R}}(pre_1(t_0))) = c^{\oplus}_{P\times\mathbb{R}}(pre_2(g_T(t_0))) \\
&= c^{\oplus}_{P\times\mathbb{R}}(pre_2(t_2)).
\end{aligned}
$$

Thus, pre_3 is well-defined. The well-definedness of $post_3$ follows analogously.

Therefore, TN_3 is a well-defined timed P/T net.

Well-definedness of c: Since (P_3, c_P) and (T_3, c_T) are Coequalisers of f_P, g_P, respectively f_T, g_T, $c \circ f = (c_P \circ f_P, c_T \circ f_T) = (c_P \circ g_P, c_T \circ g_T) = c \circ g$. From the definition of pre_3 and $post_3$ follows $pre_3 \circ c_T = c^{\oplus}_{P\times\mathbb{R}} \circ pre_2$, and $post_3 \circ c_T = c^{\oplus}_{P\times\mathbb{R}} \circ post_2$.

Therefore, c is a well-defined timed P/T morphism.

□

Fact 7.9 (Fusion of Timed P/T Nets is Coequaliser)**.** Given time-strict injective timed P/T morphisms $f, g : TN_1 \to TN_2$ with $TN_i = (P_i, T_i, pre_i, post_i)$.

Then, the fusion object TN_3 of f and g is coequaliser object of f and g and the morphism $c : TN_2 \to TN_3$ is coequaliser.

Proof. **Universal property:** Given the timed P/T net TN_4 with morphism $d : TN_2 \to TN_4$ (as seen in the figure above) with $d \circ f = d \circ g$. From the componentwise construction of the coequaliser in **Sets** follows that there are unique morphisms h_P, h_T with $h_T \circ c_T = d_T$, $h_P \circ c_P = d_P$.

Let $c_T(t') = t$ with $t' \in T_2, t \in T_3$.

For the morphism condition, we have

$$
\begin{aligned}
&h^{\oplus}_{P\times\mathbb{R}} \circ pre_3(t) = h^{\oplus}_{P\times\mathbb{R}} \circ c^{\oplus}_{P\times\mathbb{R}} \circ pre_2(t) \\
&= d^{\oplus}_{P\times\mathbb{R}} \circ pre_2(t) = pre_4 \circ d_T(t) \\
&= pre_4 \circ h_T \circ c_T(t) = pre_4 \circ h_T(t')
\end{aligned}
$$

Therefore, h is well-defined.

The uniqueness follows from the uniqueness of h_P, h_T in **Sets**.

□

Example 7.10 (Fusion of Timed P/T Nets). Figure 48 shows a fusion of two timed nets. Given are timed P/T morphisms f, g with $f_T(p1) = p1$ and $g_T(p1) = p1'$. This results in these two places being unified in $TN4$ in the place $p1$, so that the pre- and post domains of $p1$ in TN_3 matches those of $p1$ and $p1'$ in TN_2.

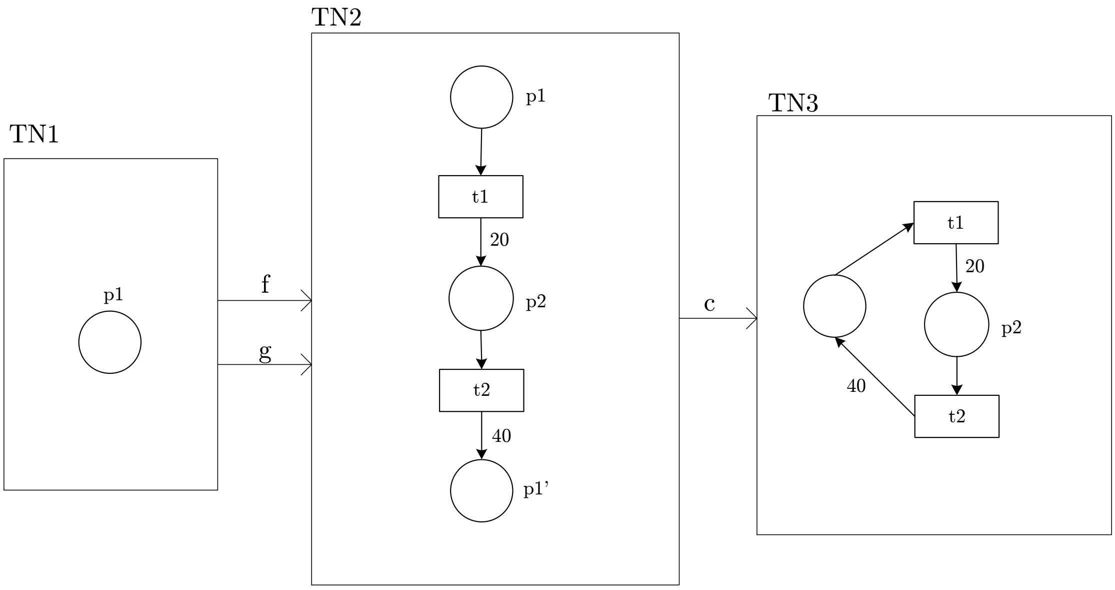


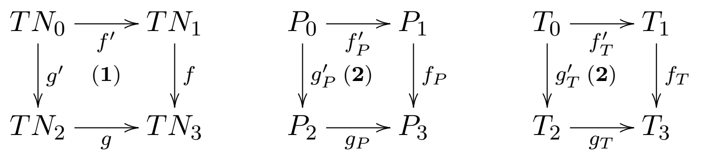
Figure 48: Fusion of Timed P/T Nets

7.3 Restriction of Timed P/T Nets

The restriction of a timed P/T net is a structuring technique that allows to restrict a timed P/T morphism (and especially its domain) to a given subnet of its codomain.

Definition 7.11 (Restriction of Timed P/T Nets)**.** Given timed P/T morphisms $f : TN_1 \to TN_3$ and $g : TN_2 \to TN_3$, where f is time-strict injective. The restriction $g' : TN_0 \to TN_1$ of g along f together with $f' : TN_0 \to TN_2$ is defined as follows:

- $TN_0 = (P_0, T_0, pre_0, post_0)$ with
  - P_0 is constructed as pullback (2) in **Sets** as depicted in Figure 49b,
  - T_0 is constructed as pullback (3) in **Sets** as depicted in Figure 49c,
  - $pre_0 = f'^{-1\oplus}_{P\times\mathbb{R}} \circ pre_2 \circ f'_T$, and
  - $post_0 = f'^{-1\oplus}_{P\times\mathbb{R}} \circ post_2 \circ f'_T$;
- $f' = (f'_P, f'_T)$, and $g' = (g'_P, g'_T)$.



(a) Restriction of timed P/T Nets (b) Pullback of Places (c) Pullback of Transitions

Figure 49: Restriction and pullback diagrams

Well-definedness. We have to show that pre_0 and $post_0$ are well-defined functions, and that f' and g' are timed P/T morphisms.

Pre and post functions. Since we assume that f is time-strict injective, there are f_P and f_T injective and thus monomorphisms in **Sets**. Then, by closure of monomorphisms under pullbacks, we obtain that also f'_P and f'_T are monomorphisms and hence they are injective functions.

So, for the well-definedness of pre_0 and $post_0$ we have to show that for all $t \in T_0$ and $(p,r) \leq pre_2(f'_T(t))$ there is also a place $p_0 \in P_0$ with $f'_P(p_0) = p$, and the same for the post domain. So, let $t \in T_0$ and $(p,r) \leq pre_2(f'_T(t))$ which means that $p \leq \pi_P^{\oplus}(pre_2(f'_T(t)))$. Since timed P/T morphisms preserve the location of pre and post domains, we also have

$$\begin{aligned} g_P(p) &\leq \pi_P^{\oplus}(g^{\oplus}_{P\times\mathbb{R}}(pre_2(f'_T(t)))) = \pi_P^{\oplus}(pre_3(g_T(f'_T(t)))) = \pi_P^{\oplus}(pre_3(f_T(g'_T(t)))) \\ &= \pi_P^{\oplus}(f^{\oplus}_{P\times\mathbb{R}}(pre_1(g'_T(t)))) = f_P^{\oplus}(\pi_P^{\oplus}(pre_1(g'_T(t)))). \end{aligned}$$

This means that there is also $p' \in P_1$ such that $f_P(p') = g_P(p)$ which by pullback (2) in **Sets** implies that there exists $p_0 \in P_0$ with $f'_P(p_0) = p$ and $g'_P(p_0) = p'$.

The proof for the post domain works analogously.

f' is timed P/T morphism. We have to show that for all $t \in T_0$ there is $pre_2 \circ f'_T(t) \stackrel{\leftarrow}{=} f'^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t)$ and $post_2 \circ f'_T(t) \stackrel{\rightarrow}{=} f'^{\oplus}_{P\times\mathbb{R}} \circ post_0(t)$.

Let $t \in T_0$, then we have

$$\begin{aligned} f'^{\oplus}_{P\times\mathbb{R}} \circ pre_0(t) &= f'^{\oplus}_{P\times\mathbb{R}} \circ f'^{-1\oplus}_{P\times\mathbb{R}} \circ pre_2 \circ f'_T(t) \\ &= (f'_{P\times\mathbb{R}} \circ f'^{-1}_{P\times\mathbb{R}})^{\oplus} \circ pre_2 \circ f'_T(t) = pre_2 \circ f'_T(t) \end{aligned}$$

and

$$\begin{aligned} f'^{\oplus}_{P\times\mathbb{R}} \circ post_0(t) &= f'^{\oplus}_{P\times\mathbb{R}} \circ f'^{-1\oplus}_{P\times\mathbb{R}} \circ post_2 \circ f'_T(t) \\ &= (f'_{P\times\mathbb{R}} \circ f'^{-1}_{P\times\mathbb{R}})^{\oplus} \circ post_2 \circ f'_T(t) = post_2 \circ f'_T(t). \end{aligned}$$

Thus, f' is a timed P/T morphism. Note that we also have shown that f' is time-strict. Moreover, since injective functions are monomorphisms which are closed under pullbacks, from injective functions f_P and f_T we know that also f'_P and f'_T are injective. Hence, f' is time-strict injective.

g' is a timed P/T morphism.

Due to timed P/T morphism $g \circ f' : TN_0 \to TN_3$ and time-strict injective morphism $f : TN_1 \to TN_3$, by Lemma 6.10 we obtain that also g' is a timed P/T morphism.

□

Fact 7.12 (Restriction of Timed P/T Nets is Pullback)**.** Given timed P/T morphisms $f : TN_1 \to TN_3$ and $g : TN_2 \to TN_3$, where f is time-strict injective, and the restriction $g' : TN_0 \to TN_1$ with $f' : TN_0 \to TN_2$ of g along f. Then diagram (1) in Figure 49a is a pullback in **TPTNets**.

Proof. Note due to definition of restrictions, we have also the pullbacks (2) and (3) in **Sets** depicted in Figure 49. We have to show that (1) commutes and that the universal property of pullbacks is satisfied. The commutativity of (1) follows by commutativity of its components in (2) and (3).

Let TN_4 be a timed P/T net and $h : TN_4 \to TN_1$, $k : TN_4 \to TN_2$ timed P/T morphisms with $f \circ h = g \circ k$. Then we also have $f_P \circ h_P = g_P \circ k_P$ and $f_T \circ h_T = g_T \circ k_T$ which by pullbacks (2) and (3) in **Sets** imply unique functions $m_P : P_4 \to P_0$ with $g'_P \circ m_P = h_P$ and $f'_P \circ m_P = k_P$, and $m_T : T_4 \to T_0$ with $g'_T \circ m_T = h_T$ and $f'_T \circ m_T = k_T$.

As shown in the proof of the well-definedness of f' in Definition 7.11, we have that f' is time-strict injective. Then, by morphism $k : TN_4 \to TN_2$ and time-strict injective morphism $f' : TN_0 \to TN_2$ due to Lemma 6.10 we know that $m = (m_P, m_T)$ is a timed P/T morphism. The uniqueness of m follows from uniqueness of its components. □

Corollary 7.13 (Time-Strict Injective Morphisms are Closed under Pullbacks) Given a pullback (1) of timed P/T nets as in Figure 49a along time-strict injective morphism f. Then also f' is time-strict injective.

Proof. By Fact 7.12 we know that the pullback can be constructed as restriction. The fact that f' is time-strict injective is already shown in the proof of the well-definedness of f' in Definition 7.11. □

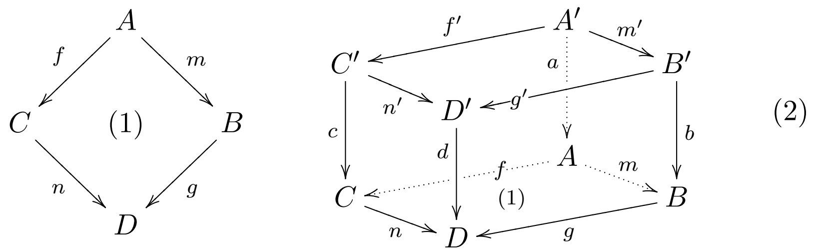
7.4 $\mathcal{M}$-Adhesive Category of Timed P/T Nets

An $\mathcal{M}$-adhesive category [EGH10] consists of a category **C** together with a class $\mathcal{M}$ of monomorphisms as defined in Definition 7.14 below. The concept of $\mathcal{M}$-adhesive categories generalizes that of adhesive [LS04], adhesive HLR [EHPP06], and weak adhesive HLR categories [EEPT06].

The concepts of adhesive [LS04] and (weak) adhesive high-level-replacement (HLR) [EEPT06] categories have been a break-through for the double pushout approach (DPO) of algebraic graph transformations [Roz97]. Almost all main results in the DPO-approach have been formulated and proven in these categorical frameworks and instantiated to a large variety of HLR systems, including different kinds of graph and Petri net transformation systems. These main results include the Local Church-Rosser, Parallelism, and Concurrency Theorems, the Embedding and Extension Theorem, completeness of critical pairs, and the Local Confluence Theorem. In [EGH10] it is shown that these results are also valid in the more general framework of $\mathcal{M}$-adhesive categories.

Definition 7.14 ($\mathcal{M}$-Adhesive Category)**.** An $\mathcal{M}$-adhesive category $(\mathbf{C}, \mathcal{M})$ is a category **C** together with a class $\mathcal{M}$ of monomorphisms satisfying:

- the class $\mathcal{M}$ is closed under isomorphisms, composition ($f, g \in \mathcal{M} \Rightarrow g \circ f \in \mathcal{M}$) and decomposition ($g \circ f, g \in \mathcal{M} \Rightarrow f \in \mathcal{M}$),
- **C** has pushouts and pullbacks along $\mathcal{M}$-morphisms,
- $\mathcal{M}$-morphisms are closed under pushouts and pullbacks, and
- the vertical weak van Kampen (short VK) property holds. This means that pushouts along $\mathcal{M}$-morphisms are $\mathcal{M}$-VK squares, i. e., a pushout (1) in Figure 50a with $m \in \mathcal{M}$ is an $\mathcal{M}$-VK square, if for all commutative cubes (2) in Figure 50b with (1) in the bottom, all vertical morphisms $a, b, c, d \in \mathcal{M}$ and pullbacks in the back faces we have that the top face is a pushout if and only if the front faces are pullbacks.



(a) $\mathcal{M}$-VK square

(b) VK cube

Figure 50: $\mathcal{M}$-VK square and VK cube

Fact 7.15 (Monomorphisms and Isomorphisms of Timed P/T Nets)**.** Given a timed P/T morphism $f : TN_1 \to TN_2$.

Monomorphisms. f is a monomorphism in **TPTNets** if and only if f_P and f_T are monomorphisms in **Sets**.

Isomorphisms. f is an isomorphism in **TPTNets** if and only if f_P and f_T are isomorphisms in **Sets** and f is time-strict.

Proof. For the proof see Appendix B.10. □

Fact 7.16 (Closure-Properties of Time-Strict Injective Morphisms)**.** Time-strict injective timed P/T morphisms are closed under composition, decomposition and isomorphisms in the following sense:

Composition. Given two time-strict injective morphisms $f : TN_1 \to TN_2$ and $g : TN_2 \to TN_3$ then also $g \circ f$ is time-strict injective.

Decomposition. Given two morphisms $f : TN_1 \to TN_2$ and $g : TN_2 \to TN_3$ such that g and $g \circ f$ are time-strict injective, then also f is time-strict injective.

Isomorphism. Given an isomorphism $f : TN_1 \to TN_2$, then f is time-strict injective.

Proof. For the proof see Appendix B.11. □

Theorem 7.17 (Timed P/T Nets Are $\mathcal{M}$-Adhesive) The category $(\mathbf{TPTNets}, \mathcal{M}_{strict})$ is an $\mathcal{M}$-adhesive category, where $\mathcal{M}_{strict} = \{f \in Mor_{\mathbf{TPTNets}} \mid f \text{ is time-strict injective}\}$.

Proof Idea. We have to show that $(\mathbf{TPTNets}, \mathcal{M}_{strict})$ satisfies the conditions of $\mathcal{M}$-adhesive categories in Definition 7.14. First, by Facts 7.15 and 7.16 we know that the class $\mathcal{M}_{strict}$ is a class of monomorphisms closed under composition, decomposition and isomorphism.

From Fact 7.2 it follows that the category **TPTNets** has pushouts along $\mathcal{M}_{strict}$-morphisms. Moreover, from Fact 7.12 it follows that the category **TPTNets** has pullbacks along $\mathcal{M}_{strict}$-morphisms.

Further, by Fact 7.4 $\mathcal{M}_{strict}$-morphisms are closed under pushouts and by Corollary 7.13 $\mathcal{M}_{strict}$-morphisms are also closed under pullbacks.

It remains to show that the vertical weak VK property is satisfied which is explicitly shown in the detailed proof in Appendix B.12. □

8 Conclusion

In this technical report, we have established a formalism for timed P/T nets, based on algebraic P/T nets and different features and approaches from other Petri net extensions, namely PTI nets and timed CPNs.

The algebraic approach presented in this paper allows formalising relations between timed nets via morphisms, allowing us to define categories for the different timed P/T net classes, specify processes of a net and apply structuring techniques such as union and fusion to timed P/T nets. Furthermore, the rather liberal definition of the firing behaviour allows for more freedom designing and simulating timed P/T nets.

The resulting timed P/T net formalism allows the modelling and analysis of time-critical systems or processes that contain events or sub-processes which take up a specified amount of time, and where timely arrival at a target state is crucial.

We have presented two examples of models (network infrastructure and workshop models) with desired requirements regarding the transition firing behaviour of the respective P/T nets, which would be either impossible to implement using classic P/T nets or require extensive changes to the model in order to achieve the desired behaviour.

After summarising other approaches and discussing different possible approaches to implementing the required features, we have given a formalisation that extends that of algebraic P/T nets, adding features for implementing the notion of time where needed: A global clock is employed, which is used to determine at which point in time a transition fires. Each token now possesses a timestamp that represents the earliest point in time (the earliest global clock value) at which it can be used in a transition. This is accounted for in the definitions of the net markings and the pre- and post domain of transitions, which are now sums of tuples of places and time value (instead of sums of places). The activation of transitions is altered accordingly, requiring the global clock value to be higher or equal than the timestamps of the tokens that are to be consumed. The tokens that are consumed upon firing, and thus subtracted from the net marking, are contained in a so-called selection. The firing step then subtracts the token selection from the current marking and adds tokens according to the post domain of the transition fired. These newly created tokens are assigned timestamps relative to the clock value at which the transition fires, indicating when they can be used for another firing step.

We have then defined categories of timed P/T nets, timed P/T-systems (analogous to P/T systems, where the objects are nets with markings) and timed P/T-states, which contain a net, a marking and a global clock value. For the categories of timed P/T-systems and timed P/T-states, we have defined functors *Rel* and *Abs*, which map timed P/T-systems onto equivalent timed P/T-states and vice versa. We have shown the categories of timed P/T-systems and timed P/T-states to be equivalent, meaning that they are essentially the same.

Additionally, we have defined skeleton functors *TSkel* and *TSkelSys* that map timed P/T-systems and timed P/T-states to regular P/T-nets and P/T-systems respectively, while preserving their firing behaviour.

Finally, we have defined structuring techniques union, fusion and restriction, analogous to structuring techniques for untimed algebraic P/T nets, and shown that the results are pushouts, coequalisers, and pullbacks, respectively. Using the structuring techniques union and restriction, we have shown that the category of timed P/T nets with the class $\mathcal{M}$ of time-strict injective morphisms fits into the abstract categorical framework of $\mathcal{M}$-adhesive categories. This means that our approach is suitable for rule-based transformation of timed P/T nets in the sense of graph transformation.

8.1 Outlook and Future Work

Aside from the approach for formalising timed Petri nets shown in this paper, there is a variety of topics not covered yet, as well as different possibilities for the enhancement of the timed P/T net notion.

One interesting topic is the definition of timed P/T processes analogously to P/T processes (see Section 4). This can be done by defining timed occurrence nets K in the way that their timed skeleton $TSkel(K)$ is an untimed occurrence net. A timed P/T process of a timed P/T net N is then a timed P/T morphism $p : K \to N$ where K is a timed occurrence net. In future research we want to analyse the properties of such timed processes in order to analyse the concurrent behaviour of timed P/T nets.

Another interesting topic is the rule-based reconfiguration of timed P/T nets. As we have shown in this work, the category of timed P/T net fits into the framework of $\mathcal{M}$-adhesive categories. Therefore, in principle it is possible to use the well-known analysis results of $\mathcal{M}$-adhesive categories for the analysis of the reconfiguration of timed P/T nets. In future work we will analyse transformation systems of timed P/T nets. For this purpose it will be important to have a condition for the existence of transformations of timed P/T nets like the gluing condition for P/T nets which is a necessary and sufficient condition for the existence of direct transformations of P/T nets.

For more complex models, a timed version of algebraic high-level nets is a viable extension, allowing for more detailed and complex models using expressions and conditions.

Algebraic high-level (AHL) nets are a powerful modelling technique in theoretical computer science. Based on algebraic P/T nets, AHL-nets use algebraic specifications as a basis for data types and firing conditions, as well as expressions for consumed and created tokens (i.e. the edge inscriptions).

An AHL-Net $AN = (\Sigma, P, T, pre, post, cond, type, A)$ consists of a signature $\Sigma = (S, OP; X)$ with additional variables X, a set of places P, a set of transitions T, pre and post domain functions $pre, post : T \to (T_\Sigma(X) \otimes P)^\oplus$, firing conditions $cond : T \to \mathcal{P}_{fin}(Eqns(\Sigma; X))$, the typing function for places $type : P \to S$, and a Σ-Algebra A. The signature $\Sigma = (S, OP)$ consists of sorts S and operation symbols OP, while $T_\Sigma(X)$ is the set of terms with variables over X. The restricted product $\otimes$ is defined by

$$(T_\Sigma(X) \otimes P) = \{(term, p) | term \in T_\Sigma(X)_{type(p)}, p \in P\}$$

and $Eqns(\Sigma; X)$ are all equations over the signature Σ with variables X.

A marking of an AHL-net AN is given by $M \in CP^{\oplus}$, where $CP = (A \otimes P) = \{(a,p)|a \in A_{type(p)}, p \in P\}$, and $M = \sum_{i=1}^{n} \lambda_i(a_i, p_i)$ means that $p_i \in P$ contains $\lambda_i \in \mathbb{N}$ data tokens $a_i \in A_{type(p_i)}$.

The set of variables $Var(t) \subseteq X$ of a transition $t \in T$ are the variables of the net inscriptions in $pre(t), post(t)$ and $cond(t)$. Let $v : Var(t) \to A$ be a variable assignment with term evaluation $\overline{v} : T_{\Sigma}(Var(t)) \to A$, then (t, v) is a consistent transition assignment iff $cond_{AN}(t)$ is validated in A under v. The set CT of consistent transition assignments is defined by $CT = \{(t,v)|(t,v)$ consistent transition assignment $\}$.

A transition $t \in T$ is enabled in M under v iff $(t,v) \in CT$ and $pre_A(t,v) \leq M$, where $pre_A : CT \to CP^{\oplus}$ is defined by $pre_A(t,v) = \hat{v}(pre(t)) \in (A \otimes P)^{\oplus}$ and $\hat{v} : (T_{\Sigma}(Var(t)) \otimes P)^{\oplus} \to (A \otimes P)^{\oplus}$ is the obvious extension of $\overline{v}$ to sums of terms and places (similar $post_A : CT \to CP^{\oplus}$). Then the follower marking is computed by $M' = M \ominus pre_A(t,v) \oplus post_A(t,v)$.

An AHL-net morphism $f : AN_1 \to AN_2$ is given by $f = (f_P, f_T)$ with functions $f_P : P_1 \to P_2$ and $f_T : T_1 \to T_2$, and is compatible with the pre and post domain, condition and typing functions. The category **AHLNet** consists of AHL-nets (with a signature Σ and algebra A), and AHL-net morphisms, with the composition of AHL-net morphisms defined componentwise for the sets of places and transitions.

By including a signature morphism and a generalized algebra morphism in the AHL-net morphisms, it is also possible to define a category of AHL-nets with different signatures and algebras for each net (see [PER95]).

The firing behaviour of AHL-nets is defined analogously to that of low-level P/T nets, with the difference that in AHL-nets, tokens contain data values instead of being dataless black tokens. In addition, for the activation of a transition, an assignment *asg* of the variables in the environment (the pre- and post domains) of the transition is required, such that the assigned pre domain is included in the current marking and the firing conditions of the transition are satisfied under *asg*. The follower marking is then computed by evaluating the edge expressions, using the assignment.

In order to incorporate time dependency in AHL-nets, all the concepts of (low-level) timed P/T nets have to be applied to AHL-nets, meaning that tokens now carry time stamps (in addition to their data), denoting the earliest point in time at which it can be used by a transition. Also, a time duration is assigned to every edge expression, yielding the same behaviour as low-level timed nets.

Consider the following example, which is based on a timed CPN network example from [JK09], remodelled with a timed AHL net.

This type of timed AHL nets is similar to the timed CPN approach, however (as seen in the definition of timed P/T nets) there are vast differences in the underlying formalisms.

Example 8.1 (Timed AHL-Net). Figure 53 shows the timed AHL variant of a network example taken from [JK09], which is shown in Figure 51. The necessary ML definitions are given in Figure 52. The firing behaviour of the timed AHL net is analogous to that of timed P/T nets, with the edge inscriptions now being sums of tuples of time values and variable names. The packet type is a product of a natural number and a string. For the operations, we have *add*, which adds two natural numbers and returns the result, *concat*, which concatenates two strings, and *packet*, which is a constructor for the packet type and takes a natural number and a string.

The two variants of the net are similar, but there are some differences due to the features of the modelling techniques used. Instead of a conditional edge inscription for packet loss, as seen in the timed CPN, for the timed AHL net, we use two transitions which are in conflict, thus deciding non-deterministically whether or not a packet arrives. Also, two transitions are used for the receiving of a packet, the used one depending on whether or not the received packet is the expected packet.

In order to compare the two nets, we choose equivalent markings for both nets. The $PacketstoSend$ place holds the packets that are to be sent across the network. A packet is a tuple of a natural number, indicating its position in the order the packets are sent, and the string, which is the data payload. An example packet would be (2,"b"). The $NextSend$ and $NextRec$ places both hold the natural number 1. These represent the number of the packet that is expected to be received. All these tokens possess a timestamp of 0, i. e. they can be used immediately. For this example, we place the packet (3,"c") on the place B in both nets with a timestamp of 120. The *Received* place contains the previously received string "ab" with a timestamp of 100, and the $NextRec$ place contains the number 3 (also with a timestamp of 100), indicating that the packet that is expected to be received has the number 3.

We first cover the original timed CPN net variant as shown in Figure 51. In a timed CPN, a *step* $Y \in BE_{MS}$ (which contains tuples (t, b) of transitions t and variable assignments b) is activated at time t' under the marking (M, t^*), if

- $\forall (t, b) \in Y : G(t)\langle b \rangle$, i.e. all transition guards are fulfilled under the variable assignments,
- $\overset{++}{MS} \sum_{(t,b)\in Y} E(p, t)\langle b \rangle \ll= M(p)$ $\forall$ untimed $p \in P$, i.e. there are enough input tokens available for the untimed places
- $\overset{+++}{MS} \sum_{(t,b)\in Y} (E(p, t)\langle b \rangle)_{+t'} \ll= M(p)$ $\forall$ timed $p \in P$, i.e. there are enough input tokens available with appropriate timestamps for the timed places,
- $t^* \leq t'$, i.e. the new clock value is larger or equal to the old clock value, and
- t' is the lowest time value for which the previous conditions are true.

In our example, there are no transition guards, the number of tokens is sufficient (with appropriate timestamps). We define $t^* = 100$ and $t' = 120$, so $t^* \leq t'$.

From the current marking M, the follower marking M' is then computed as

$$M'(p) = (M(p) - - - \overset{+++}{MS} \sum_{(t,b)\in Y} (E(p, t)\langle b \rangle)_{+t'}) + + + \overset{+++}{MS} \sum_{(t,b)\in Y} (E(t, p)\langle b \rangle)_{+t'},$$

removing the input tokens and adding the output tokens, always adding the new time value t'.

In our example, the resulting marking has tokens
1`4@137 on $NextRec$ and $C$ and 1`"acb''@137 on *Received*.

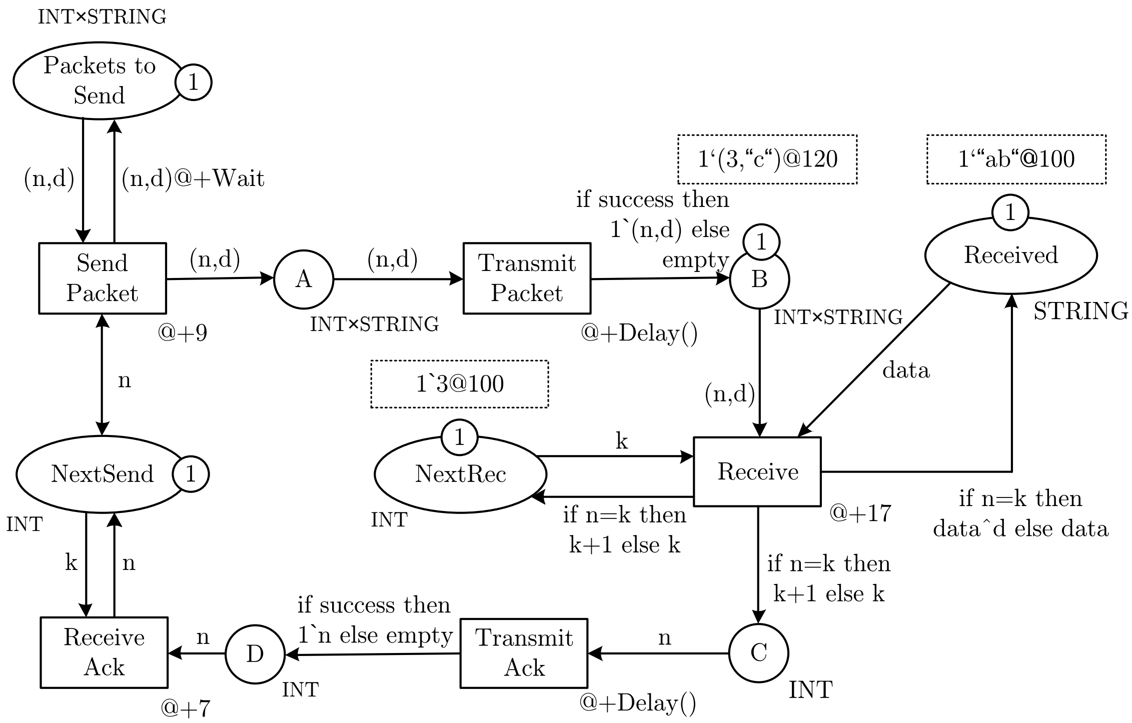


Figure 51: Network as Timed CPN

```
colset INT = int timed;
colset STRING = string timed;
colset INTxSTRING = product INT * STRING timed;
colset BOOL = bool;
var n,k :  INT;
var d,data :  STRING;
var success :  BOOL;
val Wait = 100;
fun Delay () = discrete(25,75);
```

Figure 52: ML definitions for the network timed CPN

Next, we discuss the AHL-net variant shown in Figure 53. There are slight changes from regular AHL-nets in the notation due to the added time values. Markings (and edge inscriptions) are now sums of tuples of a variable and a time value. If no time value is given for a token, the timestamp associated with that token is assumed to be zero.

In the AHL-net variant, there are some structural changes from the original timed CPN. The non-deterministic loss of packets is now done via an extra transitions for the places where packet loss can occur. These transitions, when fired, remove a packet from the input place and have no output edges. The operation *packet* is a constructor for the packet type, which are tuples of natural numbers and text strings. The *num* operation takes a packet and returns its natural number, while the *eq* operation checks two numbers for equality. Finally, the *concat* operation concatenates two strings.

In order for the transition *Receive* to be activated in marking M under the variable assignment v, the conditions of *Receive* have to be valid under the assignment and there has to be a sufficient number of tokens on the input places. In our example, the only possible assignment is $v(p) = (3, \text{“}c''\text{)}$ and $v(k) = 3$. The input tokens are obviously sufficient.

In addition to the AHL firing conditions, the timed P/T net firing conditions are in effect. This means that *Receive* can fire at global clock value 120, since one of the three input tokens has a timestamp of 120, while the other two have timestamps of 100.

Firing the transition at clock value $\tau = 120$ yields the follower marking computed by $M' = M \ominus pre_A(t,v) \oplus post_A(t,v)^{+\tau}$, with $M'(Received) = (\text{“}abc'', 137)$, $M'(nextRec) = (4, 137)$ and $M'(C) = (4, 137)$.

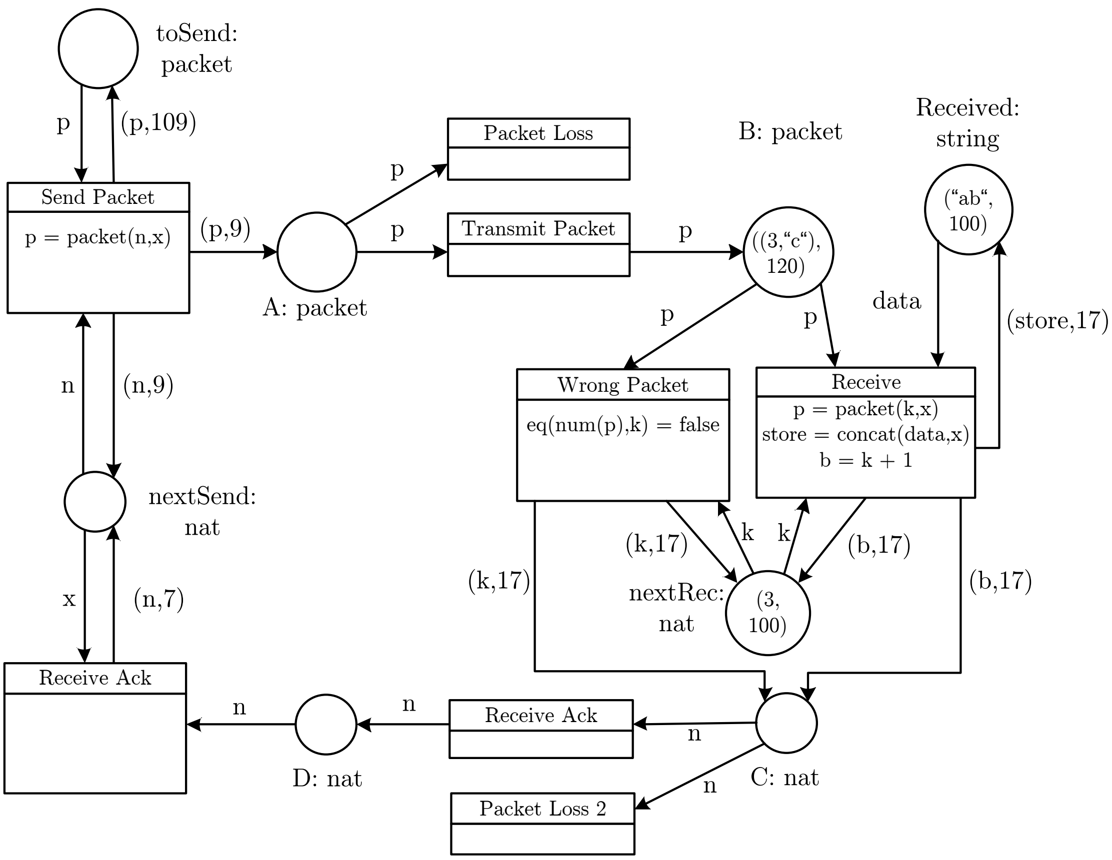


Figure 53: Network as Timed AHL net

A Categorical Fundamentals

This section contains a short overview on the required fundamentals of category theory, as defined e. g. in [EEPT06, EMC+01].

Definition A.1 (Category)**.** A category $\mathbf{C} = (Ob_C, Mor_C, \circ, id)$ consists of

- a class Ob_C of objects,
- a set of morphisms $Mor_C(A, B)$ for any two objects $A, B \in Ob_C$
- the composition operation $\circ$ for any three objects $A, B, C \in Ob_C$ with $\circ : Mor_C(A, B) \times Mor_C(B, C) \to Mor_C(A, C)$
- for each object $A \in Ob_C$ the identity $id_A \in Mor_C(A, A)$

such that the following conditions are fulfilled:

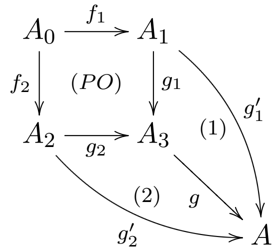
Associativity For all $f \in Mor_C(A, B), g \in Mor_C(B, C), h \in Mor_C(C, D)$ the following holds: $(h \circ g) \circ f = h \circ (g \circ f)$.

Neutrality For all $f \in Mor_C(A, B)$ the following holds: $f \circ id_A = f$ and $id_B \circ f = f$.

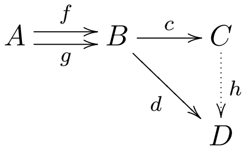
Definition A.2 (Category **Sets**)**.** The category **Sets** of sets and functions is defined as **Sets** $= (Ob_{Sets}, Mor_{Sets}, \circ, id)$ with

- the class of sets Ob_{Sets} as objects,
- $Mor_{Sets}(M, N)$, the set of functions from M to N for any two sets M and N as morphisms,
- the composition $\circ$, which is the function composition, meaning for $f : M \to N$ and $g : N \to K$, $g \circ f : M \to K$ is defined by $(g \circ f)(x) = g(f(x))$ for all $x \in M$, and
- the identity id, which are the identity functions, i.e. $id_M : M \to M$ is defined by $id_M(x) = x$ for all $x \in M$.

Definition A.3 (Pushout)**.** A pushout of two morphisms $f_1 : A_0 \to A_1$ and $f_2 : A_0 \to A_2$ of a category **C** is an object A_3, called the *pushout object*, along with two morphisms $g_1 : A_1 \to A_3$ and $g_2 : A_2 \to A_3$ in C, such that (PO) in the diagram below commutates and the following *universal property* is fulfilled: For all objects A and morphisms $g'_1 : A_1 \to A$ and $g'_2 : A_2 \to A$ in C with $g'_1 \circ f_1 = g'_2 \circ f_2$, exactly one morphism $g : A_3 \to A$ exists in C, so that (1) and (2) in the diagram below commutate:



Definition A.4 (Coequaliser)**.** Given a category $\mathbf{C} = (Ob_C, Mor_C, \circ, id)$ with $A, B \in Ob_C$, $f, g \in Mor_C$ with $f, g : A \to B$.



An object $C \in Ob_C$ with a morphism $c : B \to C$ is called *coequaliser* of f, g, if $c \circ f = c \circ g$ and the following universal property is fulfilled:

For all morphisms in $d : B \to D$ in Mor_C with $d \circ f = d \circ g$, there exists unique morphism $h : C \to D$ with $h \circ c = d$.

Definition A.5 (Initial Object)**.** An object A of a category $\mathbf{C}$ is an initial object in $\mathbf{C}$, if for each object $B \in Ob_C$ there exists a unique morphism $f : A \to B$. This means that for all objects $B \in Ob_C$, the set $Mor_C(A, B)$ contains exactly one element.

Definition A.6 (Functor)**.** Given two categories $\mathbf{C}, \mathbf{D}$. A functor $F = (F_{Ob}, F_{Mor}) : \mathbf{C} \to \mathbf{D}$ is given by

- A function $F_{Ob} : Ob_C \to Ob_D$
- For each two objects $A, B \in Ob_C$ a function

  $$F_{Mor(A,B)} : Mor_C(A, B) \to Mor_D(F_{Ob}(A), F_{Ob}(B))$$

  so that

  - for all $\mathbf{C}$-morphisms $f : A \to B$ and $g : B \to C$, the following holds:

    $$F_{Mor(A,C)}(g \circ^C f) = F_{Mor}(B, C)(g) \circ^D F_{Mor}(A, B)(f)$$

  - for all $A \in Ob_C$, the following holds:

    $$F_{Mor}(A, A)(id_A^C) = id_{F_{Ob}(A)}^D$$

Definition A.7 (Natural Transformation)**.** Given functors $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{C} \to \mathbf{D}$. Then a *functor transformation* $\alpha : F \Rightarrow G$ with $\alpha = (\alpha_A)_{A \in Ob_C}$ is a family of morphisms $\alpha_A : F(A) \to G(A)$ with $A \in Ob_C$, so that

$$\alpha_B \circ F(f) = G(f) \circ \alpha_A$$

for all $\mathbf{C}$-morphisms $f : A \to B$

Definition A.8 (Functor Category)**.** Given two categories $\mathbf{C}$ and $\mathbf{D}$.

The functor category $[\mathbf{C}, \mathbf{D}]$ is comprised of the class of all functors $F : \mathbf{C} \to \mathbf{D}$ as its objects and all natural transformations as its morphisms. The composition of natural transformations $\alpha : F \Rightarrow G$ and $\beta : G \Rightarrow H$ is the componentwise composition in $\mathbf{D}$, which means that $\beta \circ \alpha = (\beta_A \circ \alpha_A)_{A \in Ob_C}$. The identities are given by the identical natural transformations defined componentwise over the identities $id_F(A) \in \mathbf{D}$.

Definition A.9 (Equivalence of Categories)**.** Given two categories $\mathbf{C}$,$\mathbf{D}$. $\mathbf{C}$ and $\mathbf{D}$ are *equivalent*, if there are functors $I : \mathbf{C} \to \mathbf{D}$, $J : \mathbf{D} \to \mathbf{C}$, so that

$$J \circ I \simeq Id_C \in [\mathbf{C}, \mathbf{C}] \quad \text{and } I \circ J \simeq Id_D \in [\mathbf{D}, \mathbf{D}]$$

B Detailed Proofs

Fact B.1 (Monotonicity of $\square^{\oplus}_{\times\mathbb{R}}$)**.** Given a timed P/T-net $TN = (P, T, pre, post)$, timed markings $M_1, M_2 \in (P \times \mathbb{R})^{\oplus}$ of TN and a function $f_P : P \to P'$, then

$$M_1 \overset{\rightrightarrows}{=} M_2 \Rightarrow f^{\oplus}_{P\times\mathbb{R}}(M_1) \overset{\rightrightarrows}{=} f^{\oplus}_{P\times\mathbb{R}}(M_2).$$

Proof. Let $M_1 \overset{\rightrightarrows}{=} M_2$ and let us assume that

$$M'_1 := f^{\oplus}_{P\times\mathbb{R}}(M_1) \overset{\rightrightarrows}{\neq} f^{\oplus}_{P\times\mathbb{R}}(M_2) =: M'_2,$$

i.e. there is $p' \in P'$ such that for $M'_1[p'] = r'_1 \dots r'_n$ and $M'_2[p'] = s'_1 \dots s'_n$ there is $i \in \{1, \dots, n\}$ such that $r'_i > s'_i$. Since M'_2 is the image of M_2 w.r.t. $f^{\oplus}_{P\times\mathbb{R}}$ we know from $\sum_{j=1}^{i}(p', s'_j) \leq M'_2$ that for $1 \leq j \leq i$ there are $(p_j, s_j) \leq M_2$ such that $f_{P\times\mathbb{R}}(p_j, s_j) = (p', s'_j)$. By the definition of $f_{P\times\mathbb{R}}$ this implies that $f_P(p_j) = p'$.
Moreover, from $M_1 \overset{\rightrightarrows}{=} M_2$ we obtain that for $1 \leq j \leq i$ there are $(p_j, r_j) \leq M_1$ such that $r_j \leq s_j$. Thus, for $1 \leq j \leq i$ we obtain $(p', r_j) = f_{P\times\mathbb{R}}(p_j, r_j) \leq M'_1$ and there is

$$r_j \leq s_j \leq s'_i < r'_i.$$

So we have i many elements $(p', r_j) \leq M'_1$ such that $r_j < r_i$ which contradicts the fact that $M'_1[p']$ is a time-sorted list. Hence, our assumption was wrong and there is $f^{\oplus}_{P\times\mathbb{R}}(M_1) \overset{\rightrightarrows}{=} f^{\oplus}_{P\times\mathbb{R}}(M_2)$. □

B.1 Proof of Fact 6.3 (Category TPTNets)

TPTNets is Category. Given timed P/T nets $N_i = (P_i, T_i, pre_i, post_i)$ with $i = 1 \dots 3$ and timed P/T morphisms $f : N_1 \to N_2$ and $g : N_2 \to N_3$.

Composition is timed P/T morphism

Since f, g are timed P/T morphisms, the following applies:

$$(1)\ f^{\oplus}_{P\times\mathbb{R}} \circ pre_1(t) \overset{\rightrightarrows}{=} pre_2 \circ f_T(t) \ \wedge\ f^{\oplus}_{P\times\mathbb{R}} \circ post_1(t) \overset{\leftleftarrows}{=} post_2 \circ f_T(t)$$

for f, and analogously for g

$$(2)\ g^{\oplus}_{P\times\mathbb{R}} \circ pre_2(t) \overset{\rightrightarrows}{=} pre_3 \circ g_T(t) \ \wedge\ g^{\oplus}_{P\times\mathbb{R}} \circ post_2(t) \overset{\leftleftarrows}{=} post_3 \circ g_T(t)$$

To show: $(g \circ f)^{\oplus}_{P\times\mathbb{R}} \circ pre_1(t) = pre_3 \circ (g \circ f)_T(t)\ \forall t \in T$.

We show this using Fact B.1:

$$\begin{aligned}
&(g \circ f)^{\oplus}_{P\times\mathbb{R}} \circ pre_1(t) \\
&= g^{\oplus}_{P\times\mathbb{R}}(f^{\oplus}_{P\times\mathbb{R}}(pre_1(t))) \overset{\rightrightarrows(1)}{=} g^{\oplus}_{P\times\mathbb{R}}(pre_2(f_T(t))) \overset{\rightrightarrows(2)}{=} pre_3(g_T(f_T(t))) \\
&= pre_3((g \circ f)_T(t))
\end{aligned}$$

Analogously for *post*.

Therefore, the composition of two timed P/T morphisms is also a P/T morphism.

Associativity axiom is satisfied

To show: $(h \circ g) \circ f = h \circ (g \circ f)$.
Given strict timed P/T morphisms $f : A \to B, g : B \to C, h : C \to D$.
Via Associativity in Sets follows:
$(h \circ g) \circ f = ((h_P \circ g_P) \circ f_P, (h_T \circ g_T) \circ f_T) = (h_P \circ (g_P \circ f_P), h_T \circ (g_T \circ f_T)) = h \circ (g \circ f)$

Identity axiom is satisfied

To show: $f \circ id_A = f, id_B \circ f = f$.
Given strict timed P/T morphisms $f : A \to B, id_A : A \to A, id_B : B \to B$.

Via identity in Sets follows: $f \circ id_A = (f_P \circ id_{A_P}, f_T \circ id_{A_T}) = (f_P, f_T) = f$

Via identity in Sets follows: $id_B \circ f = (id_{B_P} \circ f_P, id_{B_T} \circ f_T) = (f_P, f_T) = f$

Therefore, **TPTNets** is a category.

□

B.2 Proof of Fact 6.12 (Category TPTSys)

TPTSys is Category. Given marked timed P/T nets (TN_i, M_i) with $i = 1 \dots 3$ and timed P/T morphisms $f : (TN_1, M_1) \to (TN_2, M_2)$ and $g : (TN_2, M_2) \to (TN_3, M_3)$.

From the definition of morphisms in TPTSys follows:

(1) $f^{\oplus}_{P\times\mathbb{R}}(M_1) \overset{\leftarrow}{\leq} M_2$

(2) $g^{\oplus}_{P\times\mathbb{R}}(M_2) \overset{\leftarrow}{\leq} M_3$

To show: $\forall p \in P_1 : (g \circ f)^{\oplus}_{P\times\mathbb{R}}(M_1(p)) \overset{\leftarrow}{\leq} M_3(g_P(p))$
$(g \circ f)^{\oplus}_{P\times\mathbb{R}}(M_1) = g^{\oplus}_{P\times\mathbb{R}}(f^{\oplus}_{P\times\mathbb{R}}(M_1)) \overset{\leftarrow(1)}{\leq} g^{\oplus}_{P\times\mathbb{R}}(M_2) \overset{\leftarrow(2)}{\leq} M_3$.
The associativity and identity axioms are fulfilled, as shown for the category **TPTNets**.
Therefore, TPTSys is a category. □

B.3 Proof of Fact 6.17 (Category TPTStates)

TPTStates is Category. Given timed P/T-states (TN_i, M_i, τ_i) with $i = 1 \dots 3$ and timed P/T-state morphisms $f : (TN_1, M_1, \tau_1) \to (TN_2, M_2, \tau_2)$
and $g : (TN_2, M_2, \tau_2) \to (TN_3, M_3, \tau_3)$.

Composition is timed P/T-state morphism

Since f, g are timed P/T-state morphisms, the following applies:

(1) $f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+(\tau_2-\tau_1)} \overset{\leftarrow}{\leq} M_2$

(2) $g^{\oplus}_{P\times\mathbb{R}}(M_2)^{+(\tau_3-\tau_2)} \overset{\leftarrow}{\leq} M_3$

To show: $(g \circ f)^{\oplus}_{P\times\mathbb{R}}(M_1)^{+(\tau_3-\tau_1)} \overset{\leftarrow}{\leq} M_3$

We show this using Fact B.1:

$$(g \circ f)^{\oplus}_{P\times\mathbb{R}}(M_1)^{+(\tau_3-\tau_1)}$$
$$= g^{\oplus}_{P\times\mathbb{R}}(f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+(\tau_2-\tau_1)})^{+(\tau_3-\tau_2)} \overset{\leftarrow(1)}{\leq} g^{\oplus}_{P\times\mathbb{R}}(M_2)^{+(\tau_3-\tau_2)} \overset{\leftarrow(2)}{\leq} M_3$$

The same can be shown analogously for *post*.

Therefore, the composition of two timed P/T-state morphisms is also a timed P/T-state morphism.

The associativity and identity axioms are fulfilled, as shown for the category **TPTSys**.

Therefore, **TPTStates** is a category. □

B.4 Lemma: Delay of Sums with Single Place

Lemma B.2 (Delay of Sums with Single Place) Given a set $p = \{p\}$ and timed markings $A, B, C, D \in (P \times \mathbb{R})^{\oplus}$ with $A \overset{\leftarrow}{=} B$ and $C \overset{\leftarrow}{=} D$.
Then we have $A \oplus C \overset{\leftarrow}{=} B \oplus D$.

Proof. We do a mathematical induction over the size n of A to show that the fact stated above holds for all $n \in \mathbb{N}$ (and $|A| = n$).

basis. n = 0.
This means that $|A| = 0$, i.e. A is an empty sum. Moreover, by $A \overset{\leftarrow}{=} B$, we have $\pi_P^{\oplus}(A) = \pi_P^{\oplus}(B)$ which implies that $|A| = |B|$ and hence B is empty as well.
Thus, we have $A \oplus C = 0 \oplus C = C \overset{\leftarrow}{=} D = 0 \oplus D = B \oplus D$.

basis. n = 1.
This means that $A = (p, a)$ and $B = (p, b)$ with $a, b \in \mathbb{R}$, and we have $a \geq b$. Moreover, we have $\pi_P^{\oplus}(A \oplus C) = \pi_P^{\oplus}(B \oplus B)$ by Lemma 6.5.

From $\pi_P^{\oplus}(C) = \pi_P^{\oplus}(D)$, we know that $C[p]$ and $D[p]$ have the same length m. So let $C[p] = [C_1, \ldots, C_m]$ and $D[p] = [D_1, \ldots, D_m]$.

Furthermore, we know that $(A \oplus C)[p]$ and $(B \oplus D)[p]$ have the same length of $m + 1$, so let $(A \oplus C)[p] = [E_1, \ldots, E_{m+1}]$ and $(B \oplus D)[p] = [F_1, \ldots, F_{m+1}]$.

We have to show that $(A \oplus C)[p] \geq (B \oplus D)[p]$, i.e. that for all $i \in \{1, \ldots, m + 1\}$, there is $E_i \geq F_i$.

It is important to note that $(A \oplus C)[p]$ is almost identical to the list $C[p]$ in that it has the same order of elements with the only difference being that the element a is inserted at some point in the list. The same holds for the lists $(B \oplus D)[p]$, which is basically identical to the list $D[p]$, in which the element b has been inserted.

Now, let $i \in 1, \ldots, m + 1$.

Case 1: $E_i < a$.
This means that a is inserted somewhere after index i which in turn means that $E_i = C_i$.

Case 1.1: $F_i < b$.
This means that b has been inserted after index i and we have $F_i = D_i$ and thus $E_i = C_i \geq D_i = F_i$.

Case 1.2: $F_i = b$.
This means that $b \leq D_i$, because $(B \oplus D)[p]$ is a time-sorted list. Thus, we have $E_i = C_i \geq D_i \geq b = F_i$.

Case 1.3: $F_i > b$**.**
This means that b has been inserted before index i and we have that $F_i = D_{i-1}$. From time-sorted list $(B \oplus D)[p]$ follows that $D_{i-1} \leq D_i$ and hence $E_i = C_i \geq D_i \geq D_{i-1} = F_i$.

Case 2: $E_i = a$**.**
This means that $E_{i-1} = C_{i-1} \leq E_i$.

Case 2.1: $F_i \leq b$**.**
Then, we have $E_i = a \geq b \geq F_i$.

Case 2.2: $F_i > b$**.**
This means that b has been inserted at an index greater than i, which means that $F_i = D_{i-1}$ and we have $E_i \geq C_{i-1} \geq D_{i-1} = F$.

Case 3: $E_i > a$**.**
This means that a is inserted before index i and we have $E_i = C_{i-1} > a$.

Case 3.1: $F_i \leq b$**.**
Then, we have $E_i > a \geq b \geq F_i$.

Case 3.2: $F_i > b$**.**
This means that b is inserted before index i as well, hence it follows that $F_i = D_{i-1}$ and we have $E_i = C_{i-1} \geq D_{i-1} = F_i$.

In all cases, we have that $E_i \geq F_i$ which means that $(A \oplus C)[p] \geq (B \oplus D)[p]$ and hence $A \oplus C \overset{\leftarrow}{\leqq} B \oplus D$ for $|A| = 1$.

induction hypothesis.
For $n \in \mathbb{N}$ and timed markings $A, B, C, D \in (P \times \mathbb{R})^{\oplus}$ with $|A| = n$, $A \overset{\leftarrow}{\leqq} B$ and $C \overset{\leftarrow}{\leqq} D$ it holds that $A \oplus C \overset{\leftarrow}{\leqq} B \oplus D$.

induction step.
We consider the case that $|A| = n + 1$.

Since $A \overset{\leftarrow}{\leqq} B$ and thus $\pi_P^{\oplus}(A) = \pi_P^{\oplus}(B)$, we have $|A| = |B| = n + 1$. So let $A[p] = [A_1, \ldots, A_n, A_{n+1}]$ and $B[p] = [B_1, \ldots, B_n, B_{n+1}]$ with $A_1, \ldots, A_{n+1}, B_1, \ldots, B_{n+1} \in \mathbb{R}$.

This means that $(p, A_{n+1}) \leq A$ and $(p, B_{n+1}) \leq B$, implying that there exist markings $E = A \ominus (p, A_{n+1})$ and $F = B \ominus (p, B_{n+1})$.

Moreover, we have time-sorted lists $E[p] = [A_1, \ldots, A_n]$ and $F[p] = [B_1, \ldots, B_n]$. Obviously, there is $|E| = |F| = n$ and we have $E \overset{\leftarrow}{\leqq} F$.

Using the induction hypothesis, we obtain $G := E \oplus C \overset{\leftarrow}{\leqq} F \oplus D =: H$.
Furthermore, we have
$A \oplus C = (A \ominus (p, A_{n+1})) \oplus (p, A_{n+1}) \oplus C = E \oplus (p, A_{n+1}) \oplus C = G \oplus (p, A_{n+1})$
and analogously $B \oplus D = H \oplus (p, B_{n+1})$.

It remains to show that $G \oplus (p, A_{n+1}) \leq H \oplus (p, B_{n+1})$. We have $\pi_P^{\oplus}(p, A_{n+1}) = p = \pi_P^{\oplus}(p, B_{n+1})$ and $A_{n+1} \geq B_{n+1}$ which means that $(p, A_{n+1}) \overset{\leftarrow}{\leqq} (p, B_{n+1})$.

As shown in the case for $n = 1$ in the induction basis it follows that $G \oplus (p, A_{n+1}) \overset{\leftarrow}{\leqq} H \oplus (p, B_{n+1})$. Therefore, we have $A \oplus C \overset{\leftarrow}{\leqq} B \oplus D$.

□

B.5 Proof of Lemma 6.6 (Delay of Sums)

Proof of Lemma 6.6 (Delay of Sums). Since $A \overset{\leftarrow}{=} B$ and $C \overset{\leftarrow}{=} D$, we have $\pi_P^{\oplus}(A) = \pi_P^{\oplus}(B)$ and $\pi_P^{\oplus}(C) = \pi_P^{\oplus}(D)$, implying $\pi_P^{\oplus}(A \oplus C) = \pi_P^{\oplus}(B \oplus D)$ by Lemma 6.6. Therefore, the sums have the same location. It remains to show that $(A \oplus C)[p] \geq (B \oplus D)[p]$ for all $p \in P$.

So let $p \in P$. Note that A and B, as well as C and D in particular have the same locations if restricted to p, i.e. $\pi_P^{\oplus}(A|_p) = \pi_P^{\oplus}(B|_p)$ and $\pi_P^{\oplus}(C|_p) = \pi_P^{\oplus}(D|_p)$.

Moreover, for any timed marking M, there is $M|_p[P] = M[P]$ by the definition of $M[p]$. So we have $A|_p[P] = A[p] \geq B[p] = B|_p[p]$ since $A \overset{\leftarrow}{=} B$, and analogously $C|_p[p] \geq D|_p[p]$ follows from $C \overset{\leftarrow}{=} D$.

Thus, we have $A|_p \overset{\leftarrow}{=} B|_p$ and $C|_p \overset{\leftarrow}{=} D|_p$. By Lemma B.2 we obtain $A|_p \oplus C|_p \overset{\leftarrow}{=} B|_p \oplus D|_p$.

Hence, we have
$(A \oplus C)[p] = (A \oplus C)|_p[p] = (A|_p \oplus C|_p)[p] \geq (B|_p \oplus D|_p)[p] = (B \oplus D)|_p[p] = (B \oplus D)[p]$. □

B.6 Proof of Lemma 6.7 (Delay of Differences)

Proof of Lemma 6.7 (Delay of Differences). First, note that by Fact 5.27, there is $A \leq C$ which means that $A \ominus C$ and $B \ominus D$ exist.

Now, we know that A and B have the same location, so let
$A = \sum_{i=1}^{n} \sum_{j=1}^{n_i} (p_i, a_j^i)$ and $B = \sum_{i=1}^{n} \sum_{j=1}^{n_i} (p_i, b_j^i)$ such that for $i \in 1, \ldots, n$, $p_i \in P$ there is $A[p_i] = [a_1^i, \ldots, a_{ni}^i]$ and $B[p_i] = [b_1^i, \ldots, b_{ni}^i]$.

By definition of projection of selections (definition 5.25), we have
$C = D \downarrow A = \sum_{(p_i, b_j^i) \in D} (p_i, a_j^i) = \sum_{(p_i, b_j^i) \in D} p_i = \pi_P^{\oplus}(D)$.

Thus, we have $\pi_P^{\oplus}(A \ominus C) = \pi_P^{\oplus}(A) \ominus \pi_P^{\oplus}(C) = \pi_P^{\oplus}(B) \ominus \pi_P^{\oplus}(D) = \pi_P^{\oplus}(B \ominus D)$.

It remains to show that for all $p \in P$ there is $(A \ominus C)[p] \geq (B \ominus D)[p]$. So let $p \in P$. Then there is some $i \in 1, \ldots, n$ such that $p = p_i$ according to sums A and B as denoted above.

For every $j \in 1, \ldots, n_i$ with $(p_i, a_j^i) \leq A \ominus C \leq A$, there is $(p_i, a_j^i) \not\leq C$. From the fact that $C = D \downarrow A$ it follows that $(p_i, b_j^i) \not\leq D$ and thus $(p_i, b_j^i) \leq B \ominus D$.

So let $(A \ominus C)[p_i] = [c_1, \ldots, c_m]$ and $(B \ominus D)[p_i] = [d_1, \ldots, d_m]$. The time-sorted lists $(A \ominus C)[p_i]$ and $(B \ominus D)[p_i]$ can be obtained from the lists $A[p_i]$ and $B[p_i]$ by removing elements in both of the lists at corresponding positions. Therefore, for every $k \in 1, \ldots, m$, there is some $j \in 1, \ldots, n_i$ such that $c_k = a_j^i$ and $d_k = b_j^i$. Hence, we have $c_k = a_j^i \geq b_j^i = d_k$ which means that $(A \ominus C)[p_i] \geq (B \ominus D)[p_i]$, and thus we have $A \ominus C \overset{\leftarrow}{=} B \ominus D$.

□

B.7 Proof of Theorem 6.14 (Timed P/T-system morphisms preserve firing steps)

Proof of Theorem 6.14. **Existence of Firing Step in** (TN_2, M_2')**:**

We have to show that when a firing step in (TN_1, M_1) exists, there also exists one in (TN_2, M_2).

Since there is a firing step $(TN_1, M_1) \xrightarrow{(t_1, S_1, \tau)} (TN_1, M_1')$, $t_1 \in T_1$ is activated under S_1 at τ, which means that

$S_1 \overset{\rightrightarrows}{=} pre_1(t_1)^{+\tau}$ (1)

We now show that there is a selection $S_2 \leq M_2$, so that $f_T(t_1)$ is activated under S_2 at τ, i.e. $pre_2(f_T(t_1))^{+\tau} \overset{\leftarrow}{\equiv} S_2$.

Since S_1 is a selection of M_1, via Fact B.1 it follows that $f^{\oplus}_{P\times\mathbb{R}}(S_1)$ is a selection of $f^{\oplus}_{P\times\mathbb{R}}(M_1)$.

From the timed P/T system morphism condition follows that $f^{\oplus}_{P\times\mathbb{R}}(M_1) \overset{\leftarrow}{\leq} M_2$, which (by definition 5.16) means that there exists $M_2^* \leq M_2$, so that $f^{\oplus}_{P\times\mathbb{R}}(M_1) \overset{\leftarrow}{\equiv} M_2^*$.

Then, by Lemma 5.25 (projection of selections), there exists $S_2 := f^{\oplus}_{P\times\mathbb{R}}(S_1) \downarrow M_2^*$ with $S_2 \leq M_2^* \leq M_2$, i.e. $S_2 \leq M_2$, and $S_2 \overset{\rightarrow}{\equiv} f^{\oplus}_{P\times\mathbb{R}}(S_1)$.

From this, via Fact B.1 (monotonicity of the time-enhanced function, referred to below as (2)) and the timed P/T morphism condition (referred to as (3)) follows

$$S_2 \overset{\rightarrow}{\equiv} f^{\oplus}_{P\times\mathbb{R}}(S_1) \overset{\rightarrow}{\equiv}^{(1),(2)} f^{\oplus}_{P\times\mathbb{R}}(pre_1(t_1)^{+\tau}) = f^{\oplus}_{P\times\mathbb{R}}(pre_1(t))^{+\tau} \overset{\rightarrow}{\equiv}^{(3)} pre_2(f_T(t))^{+\tau}.$$

Hence, $f_T(t_1)$ is activated under S_2 at τ).

Thus, a firing step $(TN_2, M_2) \xrightarrow{(f_T(t_1), f^{\oplus}_{P\times\mathbb{R}}(S_1)\downarrow M_2^*, \tau)} (TN_2, M_2')$ exists.

f is also timed P/T-system morphism $f : (TN_1, M_1') \to (TN_2, M_2')$:

For the morphism condition, we have to show that $f^{\oplus}_{P\times\mathbb{R}}(M_1') \overset{\leftarrow}{\leq} M_2'$, i.e. the morphism condition is also true for the follower markings.

From the definition of firing steps, we obtain the computation of the follower marking

$$f^{\oplus}_{P\times\mathbb{R}}(M_1') = f^{\oplus}_{P\times\mathbb{R}}(M_1 \ominus S_1 \oplus post_1(t)^{+\tau}).$$

Then, we obtain

$$\begin{aligned} & f^{\oplus}_{P\times\mathbb{R}}(M_1 \ominus S_1 \oplus post_1(t)^{+\tau}) \\ &= f^{\oplus}_{P\times\mathbb{R}}(M_1 \ominus S_1) \oplus f^{\oplus}_{P\times\mathbb{R}}(post_1(t)^{+\tau}) \\ &= f^{\oplus}_{P\times\mathbb{R}}(M_1) \ominus f^{\oplus}_{P\times\mathbb{R}}(S_1) \oplus f^{\oplus}_{P\times\mathbb{R}}(post_1(t)^{+\tau}). \end{aligned}$$

Via Lemma 6.6 and the morphism condition follows

$$f^{\oplus}_{P\times\mathbb{R}}(M_1) \ominus f^{\oplus}_{P\times\mathbb{R}}(S_1) \oplus f^{\oplus}_{P\times\mathbb{R}}(post_1(t)^{+\tau}) \overset{\leftarrow}{\equiv} f^{\oplus}_{P\times\mathbb{R}}(M_1) \ominus f^{\oplus}_{P\times\mathbb{R}}(S_1) \oplus post_2(f_T(t_1))^{+\tau}$$

From this, via Lemma 6.7, we obtain (replacing the resulting term with the letter X for better readability)

$$f^{\oplus}_{P\times\mathbb{R}}(M_1) \ominus f^{\oplus}_{P\times\mathbb{R}}(S_1) \oplus post_2(f_T(t_1))^{+\tau} \overset{\leftarrow}{\equiv} M_2^* \ominus S_2 \oplus post_2(f_T(t_1))^{+\tau} =: X$$

Moreover, we obtain

$$\begin{aligned} & X \leq (M_2^* \ominus S_2 \oplus post_2(f_T(t_1))^{+\tau}) \oplus (M_2 \ominus M_2^*) \\ &= M_2 \ominus M_2^* \oplus M_2^* \ominus S_2 \oplus post_2(f_T(t_1))^{+\tau} \\ &= M_2 \ominus S_2 \oplus post_2(f_T(t_1))^{+\tau} = M_2'. \end{aligned}$$

Thus, $f^{\oplus}_{P\times\mathbb{R}}(M_1') \overset{\leftarrow}{\leq} M_2'$.

Therefore, f can be considered as a P/T-system morphism
$f : (TN_1, M_1') \to (TN_2, M_2')$.

□

B.8 Proof of Theorem 6.19 (Timed P/T-state morphisms preserve firing steps)

Proof of Theorem 6.19. **Existence of Firing Step in** (TN_2, M_2', τ_2)**:**

We have to show that when a firing step in (TN_1, M_1, τ_1) exists, there also exists one in (TN_2, M_2, τ_2).

Since there is a firing step $(TN_1, M_1, \tau_1) \xrightarrow{(t_1,S_1,\tau_1)} (TN_1, M_1', \tau_1)$, $t \in T_1$ is activated under S_1 at τ_1, which means that

$$S_1 \overset{\rightrightarrows}{=} pre_1(t_1)^{+\tau_1} \quad (1)$$

We now show that there is a selection $S_2 \leq M_2$, so that $f_T(t_1)$ is activated under S_2 at τ_2, i.e. $pre_2(f_T(t_1))^{+\tau_2} \overset{\leftleftarrows}{=} S_2$. Since S_1 is a selection of M_1, via Fact B.1 it follows that $f^{\oplus}_{P\times\mathbb{R}}(S_1)$ is a selection of $f^{\oplus}_{P\times\mathbb{R}}(M_1)$.

From the timed P/T state morphism condition follows that $f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+(\tau_2-\tau_1)} \overset{\leftarrow}{\leq} M_2$, which means that there exists $M_2^* \leq M_2$, so that $f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+(\tau_2-\tau_1)} \overset{\leftleftarrows}{=} M_2^*$.

Then, by Lemma 5.25 (projection of selections), there exists $S_2 := f^{\oplus}_{P\times\mathbb{R}}(S_1) \downarrow M_2^*$ with $S_2 \leq M_2^* \leq M_2$, i.e. $S_2 \leq M_2$, with $S_2 \overset{\rightrightarrows}{=} f^{\oplus}_{P\times\mathbb{R}}(S_1)^{+\tau_2-\tau_1}$.

From this, via Fact B.1 (monotonicity of the time-enhanced function, referred to below as (2)), and the timed P/T morphism condition (referred to as (3)) follows

$$\begin{aligned} S_2 &\overset{\rightrightarrows}{=} f^{\oplus}_{P\times\mathbb{R}}(S_1)^{+\tau_2-\tau_1} \overset{\rightrightarrows}{=}^{(1),(2)} f^{\oplus}_{P\times\mathbb{R}}(pre_1(t_1)^{+\tau_1})^{+\tau_2-\tau_1} \\ &= f^{\oplus}_{P\times\mathbb{R}}(pre_1(t_1))^{+\tau_1+\tau_2-\tau_1} \overset{\rightrightarrows}{=}^{(3)} pre_2(f_T(t_1))^{+\tau_2} \end{aligned}$$

Hence, $f_T(t_1)$ is activated under (S_2, τ_2).

Thus, a firing step $(TN_2, M_2, \tau_2) \xrightarrow{(f_T(t_1),S_2,\tau_2)} (TN_2, M_2', \tau_2)$ exists.

f is also timed P/T state morphism $f : (TN_1, M_1') \to (TN_2, M_2')$**:**

For the morphism condition, we have to show that $f^{\oplus}_{P\times\mathbb{R}}(M_1')^{+(\tau_2-\tau_1)} \overset{\leftarrow}{\leq} M_2'$, i.e. the morphism condition is also true for the follower markings.

From the definition of firing steps, we obtain the computation of the follower marking

$$M_2' = M_2 \ominus S_2 \oplus post_2(f_T(t_1))^{+\tau_2}$$

Then, vie the timed P/T state morphism condition, we obtain

$$\begin{aligned} &M_2 \ominus S_2 \oplus post_2(f_T(t_1))^{+\tau_2} \\ \overset{\rightarrow}{\geq}\ & f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\tau_2-\tau_1} \ominus (f^{\oplus}_{P\times\mathbb{R}}(S_1)^{+\tau_2-\tau_1} \downarrow M_2^*) \oplus post_2(f_T(t_1))^{+\tau_2}. \end{aligned}$$

Via the timed P/T-morphism condition and the definition of the projection of selections $\downarrow$ follows

$$\begin{aligned} &f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\tau_2-\tau_1} \ominus (f^{\oplus}_{P\times\mathbb{R}}(S_1)^{+\tau_2-\tau_1} \downarrow M_2^*) \oplus post_2(f_T(t_1))^{+\tau_2} \\ \overset{\rightrightarrows}{=}\ & f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\tau_2-\tau_1} \ominus (f^{\oplus}_{P\times\mathbb{R}}(S_1)^{+\tau_2-\tau_1} \downarrow M_2^*) \oplus f^{\oplus}_{P\times\mathbb{R}}(pre_1(t_1))^{+\tau_2} \\ \overset{\rightrightarrows}{=}\ & f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\tau_2-\tau_1} \ominus f^{\oplus}_{P\times\mathbb{R}}(S_1)^{+\tau_2-\tau_1} \oplus f^{\oplus}_{P\times\mathbb{R}}(pre_1(t_1))^{+\tau_2} \end{aligned}$$

Then, via successive application of Lemma 6.6 and 6.7, we obtain

$$\begin{aligned}
&f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\tau_2-\tau_1} \ominus f^{\oplus}_{P\times\mathbb{R}}(S_1)^{+\tau_2-\tau_1} \oplus f^{\oplus}_{P\times\mathbb{R}}(post_1(t_1))^{+\tau_2} \\
\overset{\rightarrow}{=}\ & f^{\oplus}_{P\times\mathbb{R}}(M_1)^{+\tau_2-\tau_1} \ominus (f^{\oplus}_{P\times\mathbb{R}}(S_1 \oplus post_1(t_1))^{+\tau_2})^{+\tau_2-\tau_1} \\
\overset{\rightarrow}{=}\ & f^{\oplus}_{P\times\mathbb{R}}(M_1 \ominus S_1 \oplus post_1(t_1)^{+\tau_1})^{+\tau_2-\tau_1} = f^{\oplus}_{P\times\mathbb{R}}(M_1')^{+\tau_2-\tau_1}
\end{aligned}$$

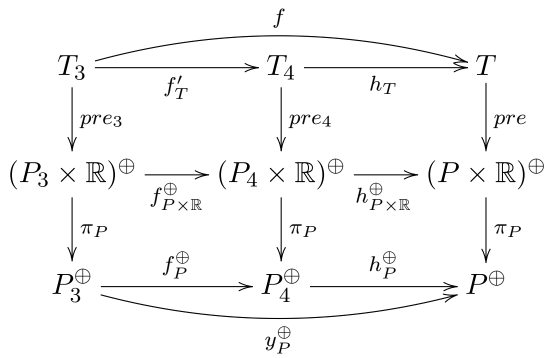
Thus, $f^{\oplus}_{P\times\mathbb{R}}(M_1')^{+(\tau_2-\tau_1)} \overset{\leftarrow}{\leq} M_2'$. Therefore, f can be considered a P/T-state morphism $f : (TN_1, M_1', \tau_1) \to (TN_2, M_2', \tau_2)$.

□

B.9 Proof of Fact 7.2 (Gluing of Timed P/T Nets is Pushout)

Proof. **Universal property:** Given timed P/T net $TN = (P, T, pre, post)$ with morphisms $x : TN_2 \to TN, x = (x_P, x_T)$ and $y : TN_3 \to TN, y = (y_P, y_T)$, so that $x \circ f = y \circ g$. Then, there exists a unique morphism $h : TN_4 \to TN$, so that $h \circ g' = x$ and $h \circ f' = y$.

- Existence: Since h is induced by the pushout construction in **Sets**, it remains to be shown that h is a well-defined timed P/T morphism.



For the locations of pre, we have:

  - Case 1: $\exists t' \in T_3 : f_T'(t') = t$.
    To show: $\pi^{\oplus}_P(pre(h_T(t))) = \pi^{\oplus}_P(h^{\oplus}_{P\times\mathbb{R}}(pre_4(t)))$.

$$\begin{aligned}
\pi^{\oplus}_P(pre(h_T(t))) &= \pi^{\oplus}_P(pre(h_T(f_T'(t')))) \\
&= \pi^{\oplus}_P(pre(y_T(t'))) = \pi^{\oplus}_P(y^{\oplus}_{P\times\mathbb{R}}(pre_3(t'))) \\
&= y^{\oplus}_{P\times\mathbb{R}}(\pi^{\oplus}_P(pre_3(t'))) = (h_P \circ f_P')^{\oplus}(\pi^{\oplus}_P(pre_3(t'))) \\
&= h^{\oplus}_p(f'^{\oplus}_P(\pi^{\oplus}_P(pre_3(t')))) \\
&= h^{\oplus}_P(\pi^{\oplus}_P(f'^{\oplus}_{P\times\mathbb{R}}(pre_3(t')))) = h^{\oplus}_P(\pi^{\oplus}_P(pre_4(f_T'(t')))) \\
&= h^{\oplus}_P(\pi^{\oplus}_P(pre_4(t))) = \pi^{\oplus}_P(h^{\oplus}_{P\times\mathbb{R}}(pre_4(t)))
\end{aligned}$$

  - Case 2: $\nexists t^* \in T_3 : f_T'(t^*) = t \wedge \exists t' \in T_2 : g_T'(t') = t$.
    To show: $\pi^{\oplus}_P(pre(h_T(t))) = \pi^{\oplus}_P(h^{\oplus}_{P\times\mathbb{R}}(pre_4(t)))$.

$$\begin{aligned}
\pi_P^{\oplus}(pre(h_T(t))) &= \pi_P^{\oplus}(pre(h_T(g'_T(t'))))\\
&= \pi_P^{\oplus}(pre(x_T(t'))) = \pi_P^{\oplus}(x_{P\times\mathbb{R}}^{\oplus}(pre_3(t')))\\
&= x_{P\times\mathbb{R}}^{\oplus}(\pi_P^{\oplus}(pre_3(t'))) = (h_P \circ g'_P)^{\oplus}(\pi_P^{\oplus}(pre_3(t')))\\
&= h_p^{\oplus}(g'^{\oplus}_P(\pi_P^{\oplus}(pre_3(t'))))\\
&= h_P^{\oplus}(\pi_P^{\oplus}(g'^{\oplus}_{P\times\mathbb{R}}(pre_3(t')))) = h_P^{\oplus}(\pi_P^{\oplus}(pre_4(g'_T(t'))))\\
&= h_P^{\oplus}(\pi_P^{\oplus}(pre_4(t))) = \pi_P^{\oplus}(h_{P\times\mathbb{R}}^{\oplus}(pre_4(t)))
\end{aligned}$$

For the locations of post, the proof is analogous.

Next, we show that the morphism condition is satisfied, i.e. we have to show that $\forall p \in P_4 : pre(h_T(t))[p] \geq h_{P\times\mathbb{R}}^{\oplus}(pre_4(t))[p]$, and $\forall p \in P_4 : post(h_T(t))[p] \leq h_{P\times\mathbb{R}}^{\oplus}(post_4(t))[p]$

– Case 1: $\exists t' \in T_3 : f'_T(t') = t$, i.e. $h_T(t) = h_T(f'_T(t')) = y_T(t')$.

$$\begin{aligned}
pre(h_T(t))[p] &= pre(y_T(t'))[p] \geq y_{P\times\mathbb{R}}^{\oplus}(pre_3(t'))[p]\\
&= (h_P \circ f'_P)_{\times\mathbb{R}}^{\oplus}(pre_3(t'))[p]\\
&= h_{P\times\mathbb{R}}^{\oplus}(f'^{\oplus}_{P\times\mathbb{R}}(pre_3(t')))[p]
\end{aligned}$$

Via the definition of pre_4, we obtain
$h_{P\times\mathbb{R}}^{\oplus}(f'^{\oplus}_{P\times\mathbb{R}}(pre_3(t')))[p] = h_{P\times\mathbb{R}}^{\oplus}(pre_4(f'_T(t')))[p] = h_{P\times\mathbb{R}}^{\oplus}(pre_4(t))[p]$.

– Case 2: $\nexists t' \in T_3 : f'_T(t') = t$.
By construction of T_4 as pushout of T_2 and T_3 it follows that there exists $t^* \in T_2$ with $g'_T(t^*) = t$, i.e. $h_T(t) = h_T(g'_T(t^*)) = x_T(t^*)$. Then,

$$\begin{aligned}
pre(h_T(t))[p] &= pre(x_T(t^*))[p] \geq x_{P\times\mathbb{R}}^{\oplus}(pre_3(t^*))[p]\\
&= (h_P \circ g'_P)_{\times\mathbb{R}}^{\oplus}(pre_3(t^*))[p]\\
&= h_{P\times\mathbb{R}}^{\oplus}(g'^{\oplus}_{P\times\mathbb{R}}(pre_3(t^*)))[p]
\end{aligned}$$

Via the definition of pre_4, we obtain
$h_{P\times\mathbb{R}}^{\oplus}(g'^{\oplus}_{P\times\mathbb{R}}(pre_3(t^*)))[p] = h_{P\times\mathbb{R}}^{\oplus}(pre_4(g'_T(t^*)))[p] = h_{P\times\mathbb{R}}^{\oplus}(pre_4(t))[p]$.

For post, the proof works analogously.

- Uniqueness: Assume there is $h' \neq h$ with $h' \circ g' = x$ and $h' \circ f' = y$. Since the pushout is constructed componentwise in **Sets**, there is a unique morphism for both the place and transition components, h_P and h_T, i.e. $h = (h_P, h_T)$ and $h' = (h_P, h_T)$.

□

B.10 Proof of Fact 7.15 (Monomorphisms and Isomorphisms of Timed P/T Nets)

Proof. **Monomorphisms.** First, we consider f_P and f_T being monomorphisms, and show that then also f is a monomorphism.

Let $g, h : TN_0 \to TN_1$ be timed P/T morphisms with $f \circ g = f \circ h$. Then we have

$$f_P \circ g_P = (f \circ g)_P = (f \circ h)_P = f_P \circ g_P$$

implying $g_P = h_P$ due to the fact that f_P is a monomorphism. Analogously, $f_T \circ g_T = f_T \circ h_T$ implies $g_T = h_T$ because also f_T is a monomorphism. Hence, we have $g = (g_P, g_T) = (h_P, h_T) = h$ which means that f is a monomorphism.

Now, let f be a monomorphism. We have to show that also f_P and f_T are monomorphisms.

Let $g_P, h_P : P_0 \to P_1$ be functions with $f_P \circ g_P = f_P \circ h_P$. We define a timed P/T net $TN_0 = (P_0, T_0, pre_0, post_0)$ with $T_0 = \emptyset$, and pre_0 and $post_0$ being empty functions. Then, by defining $g = (g_P, g_T)$ and $h = (h_P, h_T)$ with empty functions g_T and h_T, we have that $g, h : TN_0 \to TN_1$ are timed P/T morphisms, since there is no $t \in T_0$ which could violate the required condition. Moreover, we have that $f_T \circ g_T$ and $f_T \circ h_T$ both are empty functions which means that $f_T \circ g_T = f_T \circ h_T$. Thus, we have $f \circ g = f \circ h$, implying $g = h$. Hence, we also have $g_P = h_P$ which means that f_P is a monomorphism.

Finally, let $g_T, h_T : T_0 \to T_1$ be functions with $f_T \circ g_T = f_T \circ h_T$. We define a timed P/T net $TN_0 = (P_0, T_0, pre_0, post_0)$ with

- $P_0 = \{p \in P_1 \mid \exists t \in T_0 : p \leq \pi_P^{\oplus}(pre_1(g_T(t)))$ or $p \leq \pi_P^{\oplus}(post_1(g_T(t))),$
- $pre_0(t) = pre_1(g_T(t))$, and
- $post_0(t) = post_1(g_T(t))$

Obviously, pre_0 and $post_0$ are well-defined, because the definition of P_0 ensures that all places occurring in $pre_1(g_T(t))$ or $post_1(g_T(t))$ are elements of P_0. Further, we define morphisms $g = (g_P, g_T)$ and $h = (h_P, h_T)$ with g_P and h_P being inclusions. We show that $g, h : TN_0 \to TN_1$ are well-defined timed P/T morphisms. Let $t \in T_0$. We have

$$f_{P\times\mathbb{R}}^{\oplus} \circ g_{P\times\mathbb{R}}^{\oplus} \circ pre_0(t) = f_{P\times\mathbb{R}}^{\oplus} \circ g_{P\times\mathbb{R}}^{\oplus} \circ pre_1(g_T(t)) = f_{P\times\mathbb{R}}^{\oplus} \circ pre_1 \circ g_T(t)$$

and

$$\begin{aligned} f_{P\times\mathbb{R}}^{\oplus} \circ h_{P\times\mathbb{R}}^{\oplus} \circ pre_0(t) &= f_{P\times\mathbb{R}}^{\oplus} \circ h_{P\times\mathbb{R}}^{\oplus}(pre_1(g_T(t))) = f_{P\times\mathbb{R}}^{\oplus} \circ pre_1(g_T(t)) \\ &= pre_2 \circ f_T \circ g_T(t) = pre_2 \circ f_T \circ h_T(t) \\ &= f_{P\times\mathbb{R}}^{\oplus} pre_1 \circ h_T(t) \end{aligned}$$

As shown above, f being a monomorphism implies that also f_P is a monomorphism. So f_P is injective which also holds for $f_{P\times\mathbb{R}}^{\oplus}$. Thus, by monomorphism $f_{P\times\mathbb{R}}^{\oplus}$ in **Sets** we obtain by the equations above that $g_{P\times\mathbb{R}}^{\oplus} \circ pre_0(t) = pre_1 \circ g_T(t)$ and $h_{P\times\mathbb{R}}^{\oplus} \circ pre_0(t) = pre_1 \circ h_T(t)$. Hence, g and h are (time-strict) timed P/T morphisms.

So, since g_P and h_T both are inclusions, it follows that $g_P = h_P$ which especially means that $f_P \circ g_P = f_P \circ h_P$. Thus, we also have $f \circ g = f \circ h$ which by the fact that f is a monomorphism implies that $g = h$, and therefore $g_T = h_T$. Hence, f_T is a monomorphism.

Isomorphisms. First, let f_P and f_T be isomorphisms and f time-strict injective. We show that f is an isomorphism in **TPTNets**. By isomorphisms f_P and f_T in **Sets** there are functions $g_P : P_2 \to P_1$, $g_T : T_2 \to T_1$ such that g_P and f_P, and g_T and f_T are inverse isomorphisms. We define $g = (g_P, g_T)$ and show that g is a timed P/T morphism. Using the fact that f is time-strict, we have

$$\begin{aligned} pre_1 \circ g_T &= id^{\oplus}_{P_1 \times \mathbb{R}} \circ pre_1 \circ g_T = (g_P \circ f_P)^{\oplus}_{\times \mathbb{R}} \circ pre_1 \circ g_T \\ &= g^{\oplus}_{P \times \mathbb{R}} \circ f^{\oplus}_{P \times \mathbb{R}} \circ pre_1 \circ g_T = g^{\oplus}_{P \times \mathbb{R}} \circ pre_2 \circ f_T \circ g_T \\ &= g^{\oplus}_{P \times \mathbb{R}} \circ pre_2 \circ id_{T_2} = g^{\oplus}_{P \times \mathbb{R}} \circ pre_2 \end{aligned}$$

and, analogously, $post_1 \circ g_T = g^{\oplus}_{P \times \mathbb{R}} \circ post_2$. Hence, g is a (time-strict) timed P/T morphism. Finally, we have

$$g \circ f = (g_P, g_T) \circ (f_P, f_T) = (g_P \circ f_P, g_T \circ f_T) = (id_{P_1}, id_{T_1}) = id_{TN_1}$$

and analogously we obtain $f \circ g = id_{TN_2}$ which means that f and g are inverse isomorphisms in **TPTNets**.

Now, let f be an isomorphism in **TPTNets**. We show that f_P and f_T are isomorphic functions, and that f is time-strict. From f being an isomorphism, it follows that there is an inverse isomorphism $g = (g_P, g_T) : TN_2 \to TN_1$. Then, since commutativity of timed P/T morphisms implies commutativity of underlying functions, it follows immediately that f_P and g_P, and f_T and g_T are mutually inverse isomorphic functions. So, it remains to show that f is time-strict, i. e. that we have $pre_2 \circ f_T(t) = f^{\oplus}_{P \times \mathbb{R}} \circ pre_1(t)$ and $post_2 \circ f_T(t) = f^{\oplus}_{P \times \mathbb{R}} \circ post_1(t)$. By the fact that f is timed P/T morphism, we already have that $pre_2 \circ f_T(t) \overset{\leftarrow}{\leqq} f^{\oplus}_{P \times \mathbb{R}} \circ pre_1(t)$ and $post_2 \circ f_T(t) \overset{\rightarrow}{\leqq} f^{\oplus}_{P \times \mathbb{R}} \circ post_1(t)$. Moreover, by timed P/T morphism g which is inverse to f, we obtain

$$\begin{aligned} pre_2 \circ f_T(t) &= id^{\otimes}_{P_2 \times \mathbb{R}} \circ pre_2 \circ f_T(t) = (f_P \circ g_P)^{\otimes}_{\times \mathbb{R}} \circ pre_2 \circ f_T(t) \\ &= f^{\oplus}_{P \times \mathbb{R}} \circ g^{\oplus}_{P \times \mathbb{R}} \circ pre_2 \circ f_T(t) \overset{\rightarrow}{\leqq} f^{\oplus}_{P \times \mathbb{R}} \circ pre_1 \circ g_T \circ f_T(t) \\ &= f^{\oplus}_{P \times \mathbb{R}} \circ pre_1 \circ id_{T_1}(t) = f^{\oplus}_{P \times \mathbb{R}} \circ pre_1(t) \end{aligned}$$

Thus, since the location-strict delay relation $\overset{\leftarrow}{\leqq}$ is a partial order, it follows that $pre_2 \circ f_T(t) = f^{\oplus}_{P \times \mathbb{R}} \circ pre_1(t)$. The proof for $post_2 \circ f_T(t) = f^{\oplus}_{P \times \mathbb{R}} \circ post_1(t)$ works analogously. Hence, f is time-strict.

□

B.11 Proof of Fact 7.16 (Closure-Properties of Time-Strict Injective Morphisms)

Proof. **Composition.** Since injective functions are closed under composition, we have that the components of $g \circ f$ are injective, and thus, also $g \circ f$ is injective. It remains to show that $g \circ f$ is time-strict. Using the fact that f and g are time-strict, we get:

$$\begin{aligned} pre_3 \circ (g \circ f)_T &= pre_3 \circ g_T \circ f_T = g^{\oplus}_{P \times \mathbb{R}} \circ pre_2 \circ f_T \\ &= g^{\oplus}_{P \times \mathbb{R}} \circ f^{\oplus}_{P \times \mathbb{R}} \circ pre_1 \end{aligned}$$

$$\begin{aligned} post_3 \circ (g \circ f)_T &= post_3 \circ g_T \circ f_T = g^{\oplus}_{P\times\mathbb{R}} \circ post_2 \circ f_T \\ &= g^{\oplus}_{P\times\mathbb{R}} \circ f^{\oplus}_{P\times\mathbb{R}} \circ post_1 \end{aligned}$$

Hence, $g \circ f$ is time-strict and injective.

Decomposition. $g \circ f$ and g being injective means that $(g \circ f)_P = g_P \circ f_P$, g_P, $g_T \circ f_T$ and g_T are injective. So, by decomposition of injective functions we obtain that f_P and f_T are injective, and hence also f is injective. It remains to show that f is time-strict. Since $g \circ f$ and g are time-strict, we have:

$$\begin{aligned} g^{\oplus}_{P\times\mathbb{R}} \circ pre_2 \circ f_T &= pre_3 \circ g_T \circ f_T = pre_3 \circ (g \circ f)_T \\ &= (g \circ f)^{\oplus}_{P\times\mathbb{R}} \circ pre_1 = g^{\oplus}_{P\times\mathbb{R}} \circ f^{\oplus}_{P\times\mathbb{R}} \circ pre_1 \end{aligned}$$

Due to injectivity of g_P there is also $g^{\oplus}_{P\times\mathbb{R}}$ injective and thus it is a monomorphism in **Sets**. Hence, the equation above implies $pre_2 \circ f_T = f^{\oplus}_{P\times\mathbb{R}} \circ pre_1$. The proof for the post domain works analogously.

Isomorphism. By Fact 7.15 we know that for a timed P/T morphism f being an isomorphism means that f_P and f_T are isomorphisms in **Sets**, i. e. they are bijective functions, and f is time-strict. Since bijectivity implies injectivity, we have that all isomorphisms in **TPTNets** are time-strict injective.

□

B.12 Proof of Theorem 7.17 (Timed P/T Nets Are $\mathcal{M}$-Adhesive)

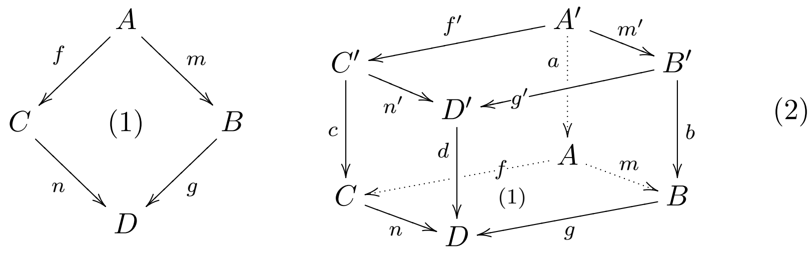
Proof. We have to show that $(\mathbf{TPTNets}, \mathcal{M}_{strict})$ satisfies the conditions of $\mathcal{M}$-adhesive categories in Definition 7.14. First, the class $\mathcal{M}_{strict}$ is a class of monomorphisms since by Fact 7.15 injective morphisms (i. e. morphisms with injective components which are monomorphisms in **Sets**) are monomorphisms in **TPTNets**. The class $\mathcal{M}_{strict}$ of all time-strict injective morphisms is closed under composition, decomposition and isomorphisms as shown in Fact 7.16.

From Fact 7.2 it follows that the category **TPTNets** has pushouts along $\mathcal{M}_{strict}$-morphisms which can be constructed as gluings of timed P/T nets as defined in Definition 7.1. Moreover, from Fact 7.12 it follows that the category **TPTNets** has pullbacks along $\mathcal{M}_{strict}$-morphisms which can be constructed as restrictions of timed P/T nets as defined in Definition 7.11.

Further, by Fact 7.4 $\mathcal{M}_{strict}$-morphisms are closed under pushouts and by Corollary 7.13 $\mathcal{M}_{strict}$-morphisms are also closed under pullbacks. It remains to show that the vertical VK property holds. So, we consider a pushout (1) as shown in Figure 54a with $m \in \mathcal{M}_{strict}$ and a cube (2) as shown in Figure 54b with (1) in the bottom, all vertical morphisms $a, b, c, d \in \mathcal{M}_{strict}$, and pullbacks in the back faces.

By construction of pushouts and pullbacks as gluings and restrictions, respectively, we also have corresponding pushouts and pullbacks in the P- and T-components, i. e. we have that the bottoms of the cubes (3) and (4) in **Sets**, shown in Figure 55, are pushouts, and the back faces are pullbacks.

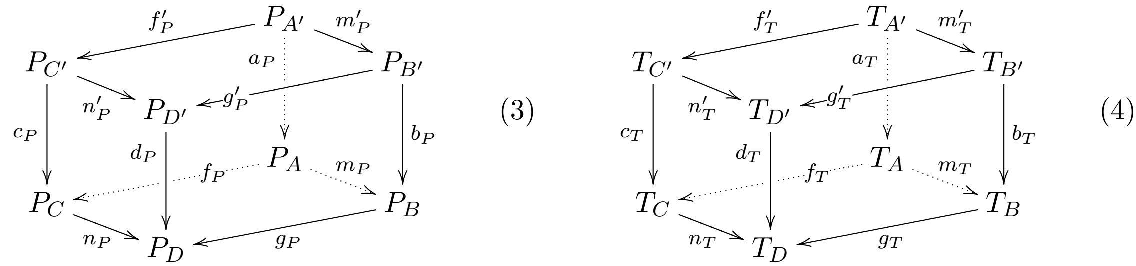
Top face pushout implies front faces pullbacks. Let the top face of the cube (2) be a pushout. Then we also have that the top faces of the cubes (3) and (4) are pushouts. In [EEPT06] it is shown that the category $(\mathbf{Sets}, \mathcal{M}_{inj})$ with the class $\mathcal{M}_{inj}$ of all



(a) $\mathcal{M}$-VK square

(b) VK cube

Figure 54: $\mathcal{M}$-VK square and VK cube



(a) VK cube of places

(b) VK cube of transitions

Figure 55: VK cubes of places and transitions

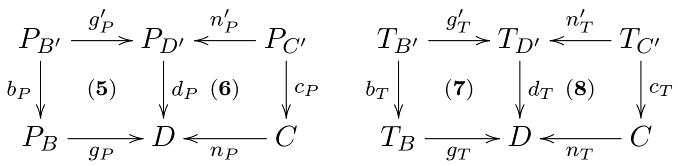
injective functions is $\mathcal{M}$-adhesive. Moreover, we have that $m_P, m_T \in \mathcal{M}_{inj}$, and all vertical morphisms $a_P, b_P, c_P, d_P, a_T, b_T, c_T, d_T \in \mathcal{M}_{inj}$. So, the vertical VK property implies that the front faces of cubes (3) and (4) are pullbacks in **Sets**, i. e. we have pullbacks (5)-(8) in Figure 56a in **Sets**.

Now, we construct the pullbacks (9) and (10) in **TPTNets** along $\mathcal{M}_{strict}$-morphism d, shown in Figure 56b. Since pullbacks along time-strict injective morphisms can be constructed as restrictions, according to Definition 7.11 we have pullbacks (11)-(14) in **Sets**, also shown in Figure 56b.

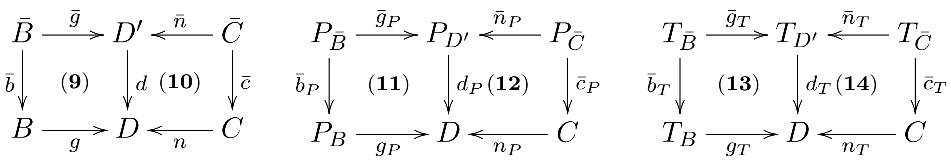
Then by uniqueness of pullbacks up to isomorphism, there is an isomorphisms $i_P : P_{B'} \to P_{\bar{B}}$ with $\bar{b}_P \circ i_P = b_P$ and $\bar{g}_P \circ i_P = g'_P$ by pullbacks (5) and (11), an isomorphism $i_T : T_{B'} \to T_{\bar{B}}$ with $\bar{b}_T \circ i_T = b_T$ and $\bar{g}_T \circ i_T = g'_T$ by pullbacks (7) and (13). Analogously, due to pullbacks (6) and (12), and (8) and (14), there are isomorphisms $j_P : P_{C'} \to P_C$ and $j_T : T_{C'} \to T_C$ with $\bar{c}_P \circ j_P = c_P$, $\bar{n}_P \circ j_P = n'_P$, $\bar{c}_T \circ j_T = c_T$, and $\bar{n}_T \circ j_T = n'_T$.

Moreover, by closure of $\mathcal{M}_{strict}$-morphisms under pullbacks, from $d \in \mathcal{M}_{strict}$ it follows that also $\bar{b}, \bar{c} \in \mathcal{M}_{strict}$. So, we have a morphism $b : B' \to B$ and a time-strict injective morphism $\bar{b} : \bar{B} \to B$ with $\bar{b}_P \circ i_P = b_P$ and $\bar{b}_T \circ i_T = b_T$ which by Lemma 6.10 implies that $i = (i_P, i_T)$ is a timed P/T morphism. Analogously, morphism $c : C' \to C$ and time-strict injective morphism $\bar{c} : \bar{C} \to C$ with $\bar{c}_P \circ j_P = c_P$ and $\bar{c}_T \circ j_T = c_T$ implies that $j = (j_P, j_T)$ is a timed P/T morphism.

Further, commutativity of the P- and T-components implies commutativity of the



(a) Pullbacks (5)-(8) in **Sets**



(b) Pullbacks (9), (10) in **TPTNets**, and pullbacks (11)-(14) in **Sets**

Figure 56: Pullbacks in **TPTNets** and **Sets**

corresponding timed P/T morphisms, i.e. we have $\bar{b} \circ i = b$ and $\bar{c} \circ j = c$. By closure of $\mathcal{M}_{strict}$-morphisms under decomposition and $\bar{b}, b, \bar{c}, c \in \mathcal{M}_{strict}$ it follows that $i, j \in \mathcal{M}_{strict}$. Hence, $i : B' \to \bar{B}$ and $j : C' to \bar{C}$ are time-strict injective morphisms with isomorphic components which by Fact 7.15 impliest that i and j are isomorphisms in **TPTNets**. Finally, due to uniqueness of pullbacks it follows that the front faces of cube (2) in Figure 54b are pullbacks in **TPTNets**.

Front faces pullbacks imply top face pushout. Now let the front faces of the cube (2) in Figure 54b be pullbacks. Then, considering again the cubes in Figure 55, we have pushouts in the bottoms and all side faces are pullbacks which implies that the top faces are pushouts by VK property in $(\mathbf{Sets}, \mathcal{M}_{inj})$.

By $m \in \mathcal{M}_{strict}$ and closure of $\mathcal{M}_{strict}$-morphisms under pullbacks, we have that also $m' \in \mathcal{M}_{strict}$, allowing us to construct the pushout (15) in **TPTNets** as gluing of timed P/T nets, shown in Figure 57, implying pushouts (16) and (17) in **Sets** according to Definition 7.1.

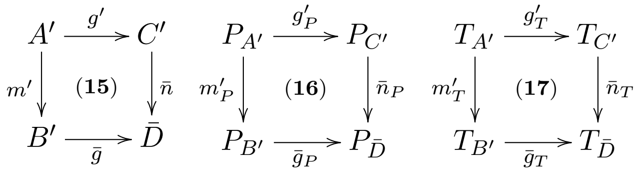


Figure 57: Pushout (15) in **TPTNets**, and pushouts (16),(17) in **Sets**

Then, by commutativity of the top face of cube (2) in Figure 54b due to the universal property of pushouts there is a unique timed P/T morphism $\bar{d} : \bar{D} \to D'$ with

$$\bar{d} \circ \bar{g} = g' \quad \text{and} \quad \bar{d} \circ \bar{n} = n'.$$

Note that we also have corresponding commutativity of the components which means that $\bar{d}_P$ and $\bar{d}_T$ are also the unique functions induced by pushouts (16) and (17) in **Sets**. Moreover, due to pushout (16) and the pushout in the top of cube (3) by uniqueness of pushouts it follows that $\bar{d}_P$ is an isomorphism. Analogously, by pushout (17) and the pushout in the top of cube (4) we have that $\bar{d}_T$ is an isomorphism. In order to show that also $\bar{d}$ is an isomorphism, according to Fact 7.15 it remains to show that $\bar{d}$ is time-strict. i.e. that for all $t \in T_{\bar{D}}$ we have $pre_{D'} \circ \bar{d}_T(t) = \bar{d}^{\oplus}_{P\times\mathbb{R}} \circ pre_{\bar{D}}(t)$ and $post_{D'} \circ \bar{d}_T(t) = \bar{d}^{\oplus}_{P\times\mathbb{R}} \circ post_{\bar{D}}(t)$.

Let $t \in T_{\bar{D}}$. By construction of pushout (15) as gluing according to Definition 7.1, we can distinguish the following two cases:

Case 1. There is $t^* \in T_{C'}$ with $\bar{n}_T(t^*) = t$.

Then by Definition 7.1 we have $pre_{\bar{D}}(t) = \bar{n}^{\oplus}_{P\times\mathbb{R}}(pre_{C'}(t^*))$. Moreover, $m \in \mathcal{M}_{strict}$ by closure under pushouts implies $n \in \mathcal{M}_{strict}$. This in turn implies $n' \in \mathcal{M}_{strict}$ by closure under pullbacks. Thus, we have:

$$\begin{aligned}\bar{d}^{\oplus}_{P\times\mathbb{R}} \circ pre_{\bar{D}}(t) &= \bar{d}^{\oplus}_{P\times\mathbb{R}} \circ \bar{n}^{\oplus}_{P\times\mathbb{R}} \circ pre_{C'}(t^*) = (\bar{d}_P \circ \bar{n}_P)^{\oplus}_{\times\mathbb{R}} \circ pre_{C'}(t^*)\\ &= n'^{\oplus}_{P\times\mathbb{R}} \circ pre_{C'}(t^*) = pre_{D'} \circ n'_T(t^*)\\ &= pre_{D'} \circ \bar{d}_T \circ \bar{n}_T(t^*) = pre_{D'} \circ \bar{d}_T(t)\end{aligned}$$

Case 2. There is no $t^* \in T_{C'}$ with $\bar{n}_T(t^*) = t$.

By uniqueness of pushouts, we can w.l.o.g. assume that also the pushout in the bottom of cube (2) in Figure 54b is constructed as a gluing of timed P/T nets as defined in Definition 7.1.

Since there is no $t^* \in T_{C'}$ with $\bar{n}_T(t^*) = t$, by $n'_T = \bar{d}_T \circ \bar{n}_T$ and injective $\bar{d}_T$ it follows that there is also no $t^* \in T_{C'}$ with $n'_T(t^*) = \bar{d}_T(t)$. So due to the pullback in the right front of cube (4) in Figure 55b there is also no $t^* \in T_C$ with $n_T(t^*) = d_T \circ \bar{d}_T(t)$. Hence, according to gluing D of B and C over A, by Definition 7.1 there is $\bar{t} \in T_B$ with $g_T(\bar{t}) = d_T \circ \bar{d}_T(t)$ and $pre_D \circ d_T \circ \bar{d}_T(t) = g^{\oplus}_{P\times\mathbb{R}} \circ pre_B(\bar{t})$. Further, by the pullback in the left front of cube (4) in Figure 55b there is $t' \in T_{B'}$ with $b_T(t') = \bar{t}$ and $g'_T(t') = \bar{d}_T(t)$. Then by $\bar{d}_T(\bar{g}_T(t')) = g'_T(t') = \bar{d}_T(t)$ and injective $\bar{d}_T$ we obtain that $\bar{g}_T(t') = t$. Thus, according to Definition 7.1 by the fact that $\bar{D}$ is a gluing of B' and C', we have that $pre_{\bar{D}}(t) = \bar{g}^{\oplus}_{P\times\mathbb{R}} \circ pre_{B'}(t')$.

So, using the fact that $b, d \in \mathcal{M}_{strict}$ are time-strict, we obtain

$$\begin{aligned}d^{\oplus}_{P\times\mathbb{R}} \circ \bar{d}^{\oplus}_{P\times\mathbb{R}} \circ pre_{\bar{D}}(t) &= d^{\oplus}_{P\times\mathbb{R}} \circ \bar{d}^{\oplus}_{P\times\mathbb{R}} \circ \bar{g}^{\oplus}_{P\times\mathbb{R}} \circ pre_{B'}(t')\\ &= d^{\oplus}_{P\times\mathbb{R}} \circ (\bar{d}_P \circ \bar{g}_P)^{\oplus}_{\times\mathbb{R}} \circ pre_{B'}(t')\\ &= d^{\oplus}_{P\times\mathbb{R}} \circ g'^{\oplus}_{P\times\mathbb{R}} \circ pre_{B'}(t')\\ &= (d_P \circ g'_P)^{\oplus}_{\times\mathbb{R}} \circ pre_{B'}(t')\\ &= (g_P \circ b_P)^{\oplus}_{\times\mathbb{R}} \circ pre_{B'}(t')\\ &= g^{\oplus}_{P\times\mathbb{R}} \circ b^{\oplus}_{P\times\mathbb{R}} \circ pre_{B'}(t')\\ &= g^{\oplus}_{P\times\mathbb{R}} \circ pre_B \circ b_T(t')\\ &= g^{\oplus}_{P\times\mathbb{R}} \circ pre_B(\bar{t})\\ &= pre_D \circ d_T \circ \bar{d}_T(t)\\ &= d^{\oplus}_{P\times\mathbb{R}} \circ pre_{D'} \circ \bar{d}_T(t)\end{aligned}$$

So we have $d^{\oplus}_{P\times\mathbb{R}} \circ \bar{d}^{\oplus}_{P\times\mathbb{R}} \circ pre_{\bar{D}}(t) = d^{\oplus}_{P\times\mathbb{R}} \circ pre_{D'} \circ \bar{d}_T(t)$ which especially holds for the case 1 above, and therefore it holds for all $t \in T_{\bar{D}}$. Thus, we have

$d^{\oplus}_{P\times\mathbb{R}} \circ \bar{d}^{\oplus}_{P\times\mathbb{R}} \circ pre_{\bar{D}} = d^{\oplus}_{P\times\mathbb{R}} \circ pre_{D'} \circ \bar{d}_T$. Since $d \in \mathcal{M}_{strict}$ is injective, also $d^{\oplus}_{P\times\mathbb{R}}$ is injective and hence it is a monomorphism in **Sets**. Thus, we have $\bar{d}^{\oplus}_{P\times\mathbb{R}} \circ pre_{\bar{D}} = pre_{D'} \circ \bar{d}_T$. The proof for the post domains works analogously.

So we have that $\bar{d}$ is time-strict, and its components are isomorphisms which by Fact 7.15 implies that it is an isomorphism in **TPTNets**. Hence, by uniqueness of pushouts up to isomorphism, we obtain that the top face of cube (2) in Figure 54b is a pushout.

□

References

[BD91] Bernard Berthomieu and Michael Diaz. Modeling and verification of time dependent systems using time petri nets. *IEEE Transactions on Software Engineering, Vol. 17, No. 3, March 1991*, 1991.

[BH07] MengChu Zhou Branislav Hruz. *Modeling and Control of Discrete-event Dynamic Systems: with Petri Nets and Other Tools.* Springer, 2007.

[CCCS92] Javier Campos, Giovanni Chiola, Jos M. Colom, and Manuel Silva. Properties and performance bounds for timed marked graphs. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 39:386–401, 1992.

[EEPT06] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation.* EATCS Monographs in Theor. Comp. Science. Springer Verlag, 2006.

[EGH10] Hartmut Ehrig, Ulrike Golas, and Frank Hermann. Categorical Frameworks for Graph Transformation and HLR Systems based on the DPO Approach. *Bulletin of the EATCS*, 102:111–121, 2010.

[EHKP91a] H. Ehrig, A. Habel, H.-J. Kreowski, and F. Parisi-Presicce. From graph grammars to high level replacement systems. In *4th Int. Workshop on Graph Grammars and their Application to Computer Science*, volume 532 of *Lecture Notes in Computer Science*, pages 269–291. Springer Verlag, 1991.

[EHKP91b] H. Ehrig, A. Habel, H.-J. Kreowski, and F. Parisi-Presicce. Parallelism and concurrency in high-level replacement systems. *Math. Struct. in Comp. Science*, 1:361–404, 1991.

[EHP+02] Hartmut Ehrig, Kathrin Hoffmann, Julia Padberg, Paolo Baldan, and Reiko Heckel. Formal and natural computing. chapter High-level net processes, pages 191–219. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[EHPP06] H. Ehrig, A. Habel, J. Padberg, and U. Prange. Adhesive High-Level Replacement Systems: A New Categorical Framework for Graph Transformation. *Fundamenta Informaticae*, 74(1):1–29, 2006.

[Ehr04] H. Ehrig. Behaviour and Instantiation of High-Level Petri Net Processes. *Fundamenta Informaticae*, 64:1–37, 2004.

[EMC+01] H. Ehrig, B. Mahr, F. Cornelius, M. Grosse-Rhode, P. Zeitz, G. Schröter, and K. Robering. *Mathematisch Strukturelle Grundlagen der Informatik, 2. überarbeitete Auflage.* Springer, 2001.

[ER97] H. Ehrig and W. Reisig. An Algebraic View on Petri Nets. *Bulletin of the EATCS*, pages 52–58, February 1997.

[GLMR05] Guillaume Gardey, Didier Lime, Morgan Magnin, and Olivier H. Roux. Romo: A tool for analyzing time petri nets. pages 418–423, 2005.

[Jen97] Kurt Jensen. A brief introduction to coloured petri nets. In *TACAS*, pages 203–208, 1997.

[JK09] Kurt Jensen and Lars Michael Kristensen. *Coloured Petri Nets - Modelling and Validation of Concurrent Systems.* Springer, 2009.

[JKW07] Kurt Jensen, Lars Michael Kristensen, and Lisa Wells. Coloured petri nets and cpn tools for modelling and validation of concurrent systems. In *International Journal on Software Tools for Technology Transfer*, page 2007, 2007.

[LS04] S. Lack and P. Sobociński. Adhesive Categories. In *Proc. FOSSACS 2004*, volume 2987 of *LNCS*, pages 273–288. Springer, 2004.

[MGE+10] Tony Modica, Karsten Gabriel, Hartmut Ehrig, Kathrin Hoffmann, Sarkaft Shareef, Claudia Ermel, Ulrike Golas, Frank Hermann, and Enrico Biermann. Low- and High-Level Petri Nets with Individual Tokens. Technical Report 2009/13, Technische Universität Berlin, 2010. `http://www.eecs.tu-berlin.de/menue/forschung/forschungsberichte/2009`.

[MGH11] Tony Modica, Karsten Gabriel, and Kathrin Hoffmann. Formalization of Petri Nets with Individual Tokens as Basis for DPO Net Transformations. In H. Ehrig, C. Ermel, and K. Hoffmann, editors, *Proc. of 4th Workshop on Petri Nets and Graph Transformation (PNGT)*, volume 40. European Association of Software Science and Technology, 2011.

[MM90] J. Meseguer and U. Montanari. Petri nets are monoids. *Information and Computation*, 88(2):105–155s, 1990.

[PER95] J. Padberg, H. Ehrig, and L. Ribeiro. Algebraic high-level net transformation systems. *Mathematical Structures in Computer Science*, 5:217–256, 1995.

[Pet62] Carl Adam Petri. *Kommunikation mit Automaten.* Bonn: Institut fr Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962.

[PPE+05] J. Padberg, U. Prange, H. Ehrig, C. Ermel, and K. Hoffmann. Skript petrinetze. pages 13–14, 52, 62ff, 2005.

[Rei85] W. Reisig. *Petrinetze, eine Einfhrung.* Springer Verlag Berlin, 1985.

[Rei91] Wolfgang Reisig. Petri nets and algebraic specifications. *Theoretical Computer Science*, 80(1):1–34, March 1991.

[Roz97] G. Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation. Vol 1: Foundations.* World Scientific, 1997.