

Coloured Petri Nets Extended with Place Capacities, Test Arcs and Inhibitor Arcs

Søren Christensen
Niels Damgaard Hansen
Computer Science Department, Aarhus University
Ny Munkegade, Bldg. 540
DK-8000 Aarhus C, Denmark
Phone: +45 86 12 71 88
Telefax: +45 86 13 57 25
Telex: 64767 aausci dk
E-mail: schristensen@daimi.aau.dk, ndh@daimi.aau.dk

Abstract

In this paper we show how to extend Coloured Petri Nets (CP-nets), with three new modelling primitives—place capacities, test arcs and inhibitor arcs. The new modelling primitives are introduced to improve the possibilities of creating models that are on the one hand compact and comprehensive and on the other hand easy to develop, understand and analyse. A number of different place capacity and inhibitor concepts have been suggested earlier, e.g., integer and multi-set capacities and zero-testing and threshold inhibitors. These concepts can all be described as special cases of the more general place capacity and inhibitor concepts defined in this paper.

We give an informal description of the new concepts and show how the concepts can be formally defined and integrated in the Petri net framework—keeping the basic properties of CP-nets. In contrast to a number of the previously suggested extensions to CP-nets the new modelling primitives preserve the concurrency properties of CP-nets. We show how CP-nets with place capacities, test arcs and inhibitor arcs can be transformed into behaviourally equivalent CP-nets without these primitives. From this we conclude that the basic properties of CP-nets are preserved and that the theory developed for CP-nets can be applied to the extended CP-nets. This means that, e.g., the generalisations of analysis methods of CP-nets to cover the new modelling primitives are straightforward.

The reader is assumed to be familiar with the notion of CP-nets.

Introduction

The development of high-level Petri Nets, CP-nets [Jen91], [Jen92] and Predicate/Transition nets [Gen86], during the late 70's and the 80's has been a substantial step in the direction of breaking the limitations of Petri Nets for modelling large and complex systems. We have, nevertheless, during our CP-net modelling, recognised that there exist a number of problems that are hard to capture by means of the existing modelling primitives. This often results in unnecessarily complex models or models which do not satisfactorily describe the modelled systems.

This is the motivation for the introduction of three new modelling primitives: place capacities, test arcs and inhibitor arcs. The primitives are general in the sense that they are equally useful for modelling in a number of different application areas—not just tailored for use in one specific area. Place capacities, test arcs and inhibitor arcs facilitate the creation of compact, comprehensive but still easily understandable models. It should be noted that the inclusion of these new modelling primitives does not, in theory, increase the computational power of CP-nets. All CP-nets including place capacities, test arcs and inhibitor arcs can be transformed into CP-nets without these primitives, but the nets will often be more difficult to read. We show how CP-nets with place capacities, test arcs and inhibitor arcs can be transformed into behaviourally equivalent CP-nets without these primitives. It is hereby ensured that the theory developed for CP-nets can be applied to the extended CP-nets. This is especially valuable when the analysis methods of CP-nets are generalised to cover extended CP-nets.

The discussion of place capacities and inhibitor arcs is based on work presented in [Bil88]. We show how the work presented in [Bil88] can be generalised and improved to overcome a number of the problems noted there.

Our paper is organised as follows. First place capacities, test arcs and inhibitor arcs are described one at a time. Each modelling primitive is presented informally before it is formally defined. It is possible to read the informal descriptions without reading the formal definitions, but it is not recommended to read the formal definitions without reading the informal introductions.

In this paper we have left out the technical details of some of the proofs. The detailed proofs can be found in [CD92].

1. Informal Introduction to Place Capacities

In this section we propose to extend CP-nets with a general place capacity construct. The approach is based on ideas originating from Kurt Jensen and briefly mentioned in [Bil88]. First we informally describe the generalised place capacity concept. Then we discuss how it relates to a number of previously proposed concepts. The formal definition of the generalised place capacity concept can be found in section 2.

Previously proposed place capacity concepts are different generalisations of the place capacity construct of P/T-nets to cover high level nets, see, e.g., [GLT79], [Res85] or [Dev89]. As they have all proved to be useful from a practical point of view for different modelling purposes, we want to define a place capacity concept for CP-nets general enough to cover all of them.

The basic idea behind our approach is described in the following. A small example illustrating our place capacity concept is given below the description. We propose to express place capacity as a combination of:

- A *capacity colour set* associated with each capacity place.

- A **capacity projection**, which is a linear function mapping markings of the capacity place into multi-sets over the capacity colour set. We use the term projection to indicate that this function is often used to disregard some of the information associated with the marking of the capacity place. The result of mapping a multi-set, ms , over the colour set into a multi-set over the capacity colour set by means of the capacity projection is called the **volume** of ms . Throughout this paper we use the term volume of an arc expression to denote the volume of the multi-set of tokens determined by the arc expression.
- A **capacity multi-set** restricting the acceptable markings of the capacity place. The capacity multi-set is a multi-set over the capacity colour set. A marking of a capacity place is legal if the volume of the marking does not exceed the capacity multi-set.

In Fig. 1.1 we show a capacity stating that a place p , holding pairs, can be marked with at most two tokens with identical first components. The colour set of p is the product of two colour sets, U and V . The place capacity of p is specified as the triple: $(U, Proj1, 2*U)$, where U is the capacity colour set, $Proj1$ is the capacity projection mapping pairs onto their first component and $2*U$ is the capacity multi-set consisting of 2 elements of each colour in U .

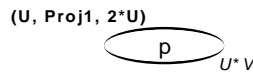


Fig 1.1: Place capacity expressing an upper limit on identical first components

In the following we show how three earlier proposed capacity concepts: *integer capacity*, *multi-set capacity* and *multi-set capacity with unbounded colours* [Bil88], can be expressed in terms of our general capacity concept.

Integer capacity, sometimes also called *total capacity*, can be represented as a single integer specifying the upper limit of the number of tokens on the place regardless of token colours. In our general framework we handle this as follows: the capacity colour set is a singleton set, e.g., $E = \{e\}$. The capacity projection is the function, Ig , ignoring the colour of the tokens. This means that Ig maps any colour into e , e.g., $Ig(3a + 2c) = 5e$, where a and c are colours of the colour set U of the capacity place. Finally the capacity multi-set is $k \cdot e$ where k is the integer determining the upper limit of the number of tokens.

From a modelling point of view, the new place capacity concept may seem to create an unnecessary overhead: you now have to specify both the capacity colour set and the capacity projection in addition to the original integer capacity. To reduce this overhead we suggest to use a shorthand notation for some of the most commonly used capacity restrictions. In the following figures the general notation is illustrated in the left-hand side whereas a shorthand notation is shown in the right-hand side.

Fig 1.2 illustrates how the integer capacity of a place p can be expressed in terms of the general place capacity concept.



Figure 1.2: Integer capacity (left: general notation, right: shorthand notation)

Multi-set capacity expresses place capacity in terms of a multi-set over the colour set of the place. This multi-set denotes the upper limit of tokens of specific colours. Multi-set capacity can, in the same way as integer capacities, easily be expressed by means of the generalised place capacity concept. The capacity colour set of a place is identical to the colour set of the place. The capacity projection is the identity function, and the capacity

multi-set is identical to the original multi-set capacity. This is illustrated in Fig 1.3 together with a shorthand notation of it.



Figure 1.3: Multi-set capacity (left: general notation, right: shorthand notation)

Multi-set capacities with unbounded colours are identical to multi-set capacities except that the multi-set is allowed to contain infinite elements [Bil88]. In terms of our general capacity concept multi-set capacities with unbounded colours can be expressed as follows. The capacity colour set of a place is identical to the colour set of the place. The capacity projection is the identity function except that unbounded colours are discarded, i.e., mapped into the empty multi-set. In this paper we denote by Id^* the identity function omitting the colours having a zero coefficient in the associated multi-set parameter (in $[]$ brackets), e.g., $Id^*[2^a+7^b](4^b + 3^c) = 4^b$. Finally, the capacity multi-set is the multi-set, where unbounded colours are discarded. Please note that the functions Id and Id^* are identical when the parameter multi-set of Id^* contains all members of the colour set. Fig. 1.4 shows a multi-set capacity with unbounded colours.



Figure 1.4: Multi-set capacity with unbounded colours (left: general notation, right: shorthand notation)

Above we have only considered simple place capacity restrictions. The general concept of place capacity also allows us to specify more elaborated capacities, e.g., only allowing lists of a certain length or pairs where the first component differs from the second.

In order to formally define the behaviour of CP-nets with place capacities, we extend the enabling rule of CP-nets to handle place capacities, but we do not change the occurrence rule. Adding a capacity restriction to a place constrains the enabling of the input transitions of the place. Informally, a step is enabled in CP-nets with place capacities if the following conditions are satisfied:

- (i) The step is enabled in the traditional CP-net sense—that is, there are enough tokens of the right colours on the input places.
- (ii) The marking of capacity places does not violate the specified capacity restrictions when the tokens produced in the step are added.

Intuitively, the second condition seems too restrictive, because you would have expected to be allowed to subtract the tokens consumed in the step before testing the capacity restriction. This would, however, violate the so-called diamond rule of Petri nets stating that steps can be split into an arbitrary sequence of substeps. The diamond rule is illustrated in Fig. 1.5: transitions A and B are concurrently enabled in marking M with bindings b_A and b_B . If A occurs with binding b_A , we arrive at marking M_a where B is enabled with binding b_B , and vice versa. If A and B occur concurrently with bindings b_A and b_B respectively, we arrive immediately at marking M_{ab} .

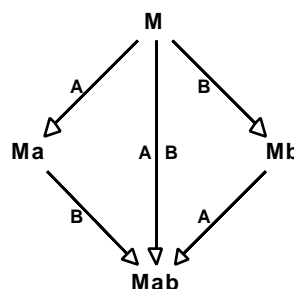


Figure 1.5. The diamond rule of Petri nets

The small net in Fig. 1.6 illustrates why it is necessary to use the enabling rule outlined above. Let us assume that the place p has integer capacity 1 and it is initially marked with one token. Transition A adds a token to p and B consumes a token from p .

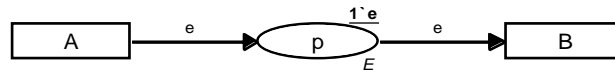


Figure 1.6: Net illustrating the enabling rule

If we use the obvious enabling rule, where we subtract the tokens consumed in the step when we calculate the enabling, this would mean that A and B are concurrently enabled, B can occur before A , but A cannot occur before B , thus violating the diamond rule as illustrated in Fig. 1.7.

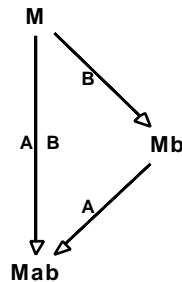


Figure 1.7: Diamond rule diagram when using obvious enabling rule

If we instead use the enabling rule (ii), outlined above, the diamond rule is preserved since our enabling rule prevents A and B from being concurrently enabled without each of them being enabled alone. We then obtain the diagram shown in Fig. 1.8.

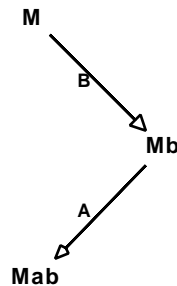


Figure 1.8: Diamond rule diagram when using our restricted enabling rule

As mentioned in the introduction, it is valuable to know that a CP-net with the new modelling primitives can be transformed into a behaviourally equivalent CP-net without the primitives. The idea behind the transformation of capacities is to add a complementary place for each capacity place and then omit the capacity restriction without hereby changing the behaviour of the model. The complementary place p' of a capacity place p can be viewed as modelling the "free space" available on the capacity place. Fig. 1.9 illustrates the transformation.

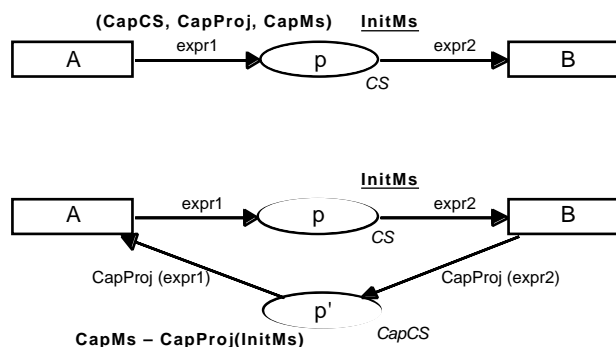


Figure 1.9: Transformation into behaviourally equivalent CP-net without place capacities

The colour set of p' is equal to the capacity colour set of p . The initial marking of p' is the multi-set capacity of p subtracted the volume of the initial marking of p . For each arc, a , connected to p we have an arc a' connecting the transition of a to p' . If a is an input arc of p , a' is an output arc of p' and vice versa. The arc expression of a' is the volume of the arc expression of a , i.e., if a multi-set m is added to p by means of the arc a , the volume of m is removed from p' , and vice versa.

The behaviour is not changed by the transformation because the arcs to the complementary place specify the same restrictions to the ordinary enabling rule as the enabling condition (ii) above. In Fig. 1.10 we have shown the CP-net obtained by transforming the net of Fig. 1.6.

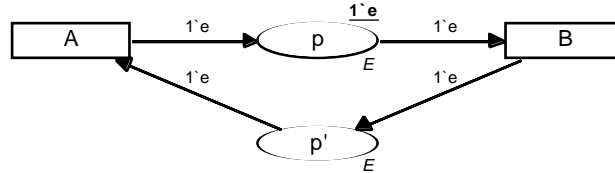


Figure 1.10: Behaviourally equivalent CP-net of the net shown in Fig. 1.6

The formal definition of CP-nets with capacities and the transformation of CP-nets with place capacity into behaviourally equivalent CP-nets can be found in section 2.

2. Formal Definition of CP-nets with Place Capacities

In this section we formalise how the place capacity concepts introduced in the previous section can be added to CP-nets. The notation used in this chapter is consistent with the one used in [Jen91] and [Jen92]. First we recall the definition and notation for multi-sets.

A **multi-set** m , over a non-empty set S , is a function $m : S \rightarrow \mathbb{N}$. The non-negative integer $m(s) \in \mathbb{N}$ is the number of appearances of the element s in the multi-set m . We usually represent the multi-set m by a formal sum:

$$\sum_{s \in S} m(s) \cdot s$$

By S_{MS} we denote the set of all multi-sets over S . The non-negative integers $\{m(s) \mid s \in S\}$ are called the coefficients of the multi-set m , and the number of appearances of s , $m(s)$, is called the **coefficient** of s . An element $s \in S$ is said to **belong** to the multi-set m iff $m(s) > 0$ and we then write $s \in m$.

2.1 The Definition of CP-nets with Place Capacities

We recall the definition of non-hierarchical CP-nets:

Definition 2.1 ([Jen92], Def. 2.5)

A non-hierarchical **CP-net** is a tuple $CPN = (C, P, T, A, N, C, G, E, I)$ satisfying the requirements below:

- (i) C is a finite set of non-empty types, called **colour sets**.
- (ii) P is a finite set of **places**.
- (iii) T is a finite set of **transitions**.
- (iv) A is a finite set of **arcs** such that:
 - $P \cap T = P \cup A = T \cup A = \emptyset$.
- (v) N is a **node** function. It is defined from A into $P \times T \cup T \times P$.
- (vi) C is a **colour** function. It is defined from P into C .

Continues

- (vii) G is a **guard** function. It is defined from T into expressions such that:
 - $t \in T: [\text{Type}(G(t)) = \mathbb{B} \quad \text{Type}(\text{Var}(G(t)))]$.
- (viii) E is an **arc expression** function. It is defined from A into expressions such that:
 - $a \in A: [\text{Type}(E(a)) = C(p(a))_{MS} \quad \text{Type}(\text{Var}(E(a)))]$
where $p(a)$ is the place of $N(a)$.
- (ix) I is an **initialization** function. It is defined from P into closed expressions such that:
 - $p \in P: [\text{Type}(I(p)) = C(p)_{MS}]$.

A detailed explanation of the requirements of this definition can be found in [Jen91] and [Jen92]. For an arc, a , we use $s(a)$, $d(a)$, $p(a)$ and $t(a)$ to denote the source, destination, place and transition respectively. $A(x)$ denotes the set of all arcs connected to the node x . Moreover, we use $\text{Var}(\text{expr})$ to denote the variables appearing in an expression expr and we use $\text{Type}(v)$ and $\text{Type}(\text{expr})$ to denote the type of a variable v and an expression expr , respectively. For more information about the use of types, expressions and variables in CP-nets see [Jen91] or [Jen92].

We now formally define CP-nets with place capacities. Some explanation is given below the formal definition. It is recommended to read this explanation in parallel with the definition.

Definition 2.2

A CP-net with place capacities is a tuple $\text{CPN}_C = (\text{CPN}, \text{CS})$ satisfying the requirements below:

- (i) CPN is a non-hierarchical CP-net: $(\Sigma, P, T, A, N, C, G, E, I)$.
- (ii) CS is a **capacity specification**: $(\text{CN}, \text{CC}, \text{CP}, \text{CM})$.
- (iii) $\text{CN} \subseteq P$ is a set of **capacity nodes**.
- (iv) CC is a **capacity colour** function. It is defined from CN into Σ .
- (v) CP is a **capacity projection** function. It is defined from CN into linear functions such that:
 - $p \in \text{CN}: \text{CP}(p) \in [C(p)_{MS} \quad \text{CC}(p)_{MS}]_{\mathbb{L}}$.
- (vi) CM is a **capacity multi-set** function. It is defined from CN into closed expressions such that:
 - $p \in \text{CP}: [\text{Type}(\text{CM}(p)) = \text{CC}(p)_{MS} \quad \text{CP}(p)(I(p)) \quad \text{CM}(p)]$.

- (i) CPN is a CP-net, cf. Def. 2.1.
- (ii) A capacity specification is defined as a four tuple satisfying (iii) - (vi).
- (iii) CN denotes the subset of places with capacity restriction. If a place p belongs to CN we say that p is a capacity place. In Fig. 1.1 we have a single capacity place p .
- (iv) For each capacity place we specify a capacity colour set. The capacity colour set of p in Fig. 1.1 is U .
- (v) The capacity projection maps multi-sets over the colour set of the place to multi-sets over the capacity colour set. We demand that CP is linear. We say that a multi-set ms over the colour set of a capacity place p has **volume** $\text{CP}(p)(ms)$. Analogously for an arc a we say that the volume of the arc expression of a is the volume of the multi-set determined by the arc expression, i.e., $\text{CP}(p(a))(E(a))$. The projection function of p in Fig. 1.1 is $Proj_1$. If the marking of p is $1^{\setminus(a,x)} + 1^{\setminus(a,y)} + 2^{\setminus(b,x)}$ the volume of p is $2^{\setminus a} + 2^{\setminus b}$.
- (vi) The capacity multi-set of a place is a multi-set over the capacity colour set of the place. We demand that the volume of the initial marking of p is less than or equal to the capacity multi-set of p . In Fig. 1.1 we have a capacity multi-set: $2 * U$.

2.2 Behaviour of CP-nets with Place Capacities

Having defined the static structure of CP-nets with place capacities we are now ready to consider their behaviour. Adding a capacity restriction to a given place constrains the enabling of the input transitions of the place, but does not change the effect of an occurrence.

Our definitions of bindings, $B(t)$, token elements, TE, binding elements, BE, markings, \mathbb{M} , and steps, \mathbb{Y} , are identical to the definitions given in [Jen92], which except for technical details are identical to those of [Jen91]. However, the enabling rule of CP-nets must be extended to take place capacities into account.

Definition 2.3

A step $Y \in \mathbb{Y}$ is **enabled** in a marking $M \in \mathbb{M}$ iff the following properties are satisfied:

- (i) $\forall p \in P: \sum_{t \in T} (t, p) \cdot Y_t \leq E(p, t) \wedge M(p) \leq C(p)$.
- (ii) $\forall p \in CN: CP(p)(M(p) + \sum_{t \in T} (t, p) \cdot Y_t) \leq CM(p)$.

If Y fulfils (ii) it is said to be **capacity enabled**.

- (i) This is the enabling rule of CP-nets, See [Jen91] Def. 2.6 or [Jen92] Def. 2.8.
- (ii) The capacity enabling rule ensures that all reachable markings fulfil the restriction imposed by the place capacities, i.e., the volume of the marking of each capacity place must be less than or equal to the capacity multi-set.

This rule is comparable to the usual enabling rule ensuring that the marking of a place exceeds the multi-set of tokens consumed by an enabled step without taking advantage of the tokens added in the step. In the capacity enabling we also include the volume of the tokens added, but we do not subtract the volume of the tokens removed as discussed earlier.

2.3 Behaviourally Equivalent CP-net

All CP-nets with place capacities can be transformed into behaviourally equivalent CP-nets. The transformation is informally described in section 1.

Definition 2.4

Let a CP-net with place capacities $CPN_C = (CPN, CS)$ be given with $CPN = (\Sigma, P, T, A, N, C, G, E, I)$ and $CS = (CN, CC, CP, CM)$. Then we define the **equivalent CP-net** to be $CPN^* = (\Sigma^*, P^*, T^*, A^*, N^*, C^*, G^*, E^*, I^*)$ where:

- (i) $\Sigma^* = \Sigma$.
- (ii) $P^* = P \cup P'$ where $P' = \{p' \mid p \in CN\}$.
- (iii) $T^* = T$.
- (iv) $A^* = A \cup A'$ where $A' = \{a' \mid a \in A(CN)\}$.
- (v) $N^*(a) = \sum_{t \in T} (a, t) \cdot I_t$, iff $a \in A$, $N^*(a') = \sum_{t \in T} (a', t) \cdot I_t$, iff $a' \in A'$, $N^*(b) = \sum_{t \in T} (b, t) \cdot I_t$, iff $b \in P$, $N^*(b') = \sum_{t \in T} (b', t) \cdot I_t$, iff $b' \in P'$.
- (vi) $C^*(p) = \{C(p), \text{ iff } p \in P, CC(q), \text{ iff } p=q' \in P'\}$.
- (vii) $G^* = G$.
- (viii) $E^*(a) = \{E(a), \text{ iff } a \in A, CP(p(b))(E(b)), \text{ iff } a=b' \in A'\}$. *Continues*
- (ix) $I^*(p) = \{I(p), \text{ iff } p \in P, CM(q) - CP(q)(I(q)), \text{ iff } p=q' \in P'\}$.

- (i) The set of colour sets is unchanged.

- (ii) We extend the set of places to include a complementary place of each capacity place. We use single prime to indicate a complementary place, i.e., the complementary place of p is denoted p' . We assume P' to be disjoint from other net elements.
- (iii) The set of transitions is unchanged.
- (iv) The set of arcs is extended to include arcs connected to the complementary places. For each arc a connected to a capacity place we have an arc, a' , connected to the complementary place. We assume A' to be disjoint from all other net elements.
- (v) The node function is changed to match (iv). A complementary arc a' connects a complementary place with the transition $t(a)$. If a is an input arc of p , a' will be an output arc of p' and vice versa.
- (vi) The colour function is extended to handle the complementary places.
- (vii) The guard function is unchanged.
- (viii) The arc expressions are changed to match (iv). The arc expression of an arc connected to a complementary place is the volume of the arc expression of the arc connected to the original place.
- (ix) The initialisation expressions are extended to handle the complementary places. The initial marking of a complementary place is the multi-set difference between the capacity multi-set and the volume of the initial marking of the original place.

Before we show that the transformation specified in Def. 2.4 preserves the behaviour of the CP-nets, we show a lemma that allows us to restrict the set of markings without neglecting any reachable marking.

Lemma 2.5

Let CPN be a CP-net with complementary places corresponding to the capacity specification $CS = (CN, CC, CP, CM)$.

Then we have the following property:

$$p \in CN \quad M \in [M_0]: M(p') = CM(p) - CP(p)(M(p)).$$

Proof: The proof can be derived from the previous definitions in the following way: the property is true in the initial marking due to the transformation in Def. 2.4 (ix). If a binding changes the marking of a capacity place the arcs to the complementary place ensure that the marking of the complementary place is updated accordingly: from Def. 2.4(iv) we know that an arc a , connected to a capacity place will have a corresponding arc a' . From Def. 2.4(v) we know that if an input arc of p then a' is an output arc of p' and vice versa. Finally Def. 2.4(viii) specifies that the arc expression of a' is the volume of the arc expression of a .

We are now ready to show that the transformation specified in Def. 2.4 preserves the behaviour of the CP-nets. We show that there is a correspondence between markings and steps of the original CP-net with place capacities and markings and steps of the transformed CP-net.

\mathbb{M} denotes the set of all markings and \mathbb{Y} denotes the set of all bindings, see ([Jen92], Def. 2.13). We use $M \upharpoonright Q$ to denote function restriction of M to the subset of places specified by Q . All concepts with a star refer to CPN^* , while those without refer to CPN_C . We denote the subset of all markings \mathbb{M}^* fulfilling the capacity restriction of lemma 2.5 by \mathbb{M}^+ .

Theorem 2.6

Let CPN_C be a CP-net with place capacities and let CPN^* be the equivalent CP-net. Then we have the following properties:

- (i) $M = (M^* | P) \quad M_0 = (M_0^* | P).$
- (ii) $Y = Y^*.$
- (iii) $M_1, M_2 \in M^+, \quad Y \in Y: (M_1 | P) [Y \rangle_{CPN_C} (M_2 | P) \quad M_1 [Y \rangle_{CPN^*} M_2.$

Proof: A detailed proof can be found in [CD92].

3. Informal Introduction to Test Arcs

When modelling with CP-nets you often have to model a situation similar to a traditional database system where multiple users share some common data. A straightforward CP-net description of this situation is shown in Fig. 3.1, where the actual database is modelled as a token on the place Data Base.

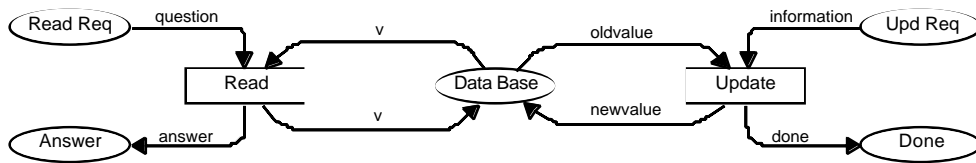


Figure 3.1: Small database example

Users are often allowed to read the data simultaneously, but only one at a time is allowed to update the data—often called multiple-read/single-write. Hereby data integrity is ensured.

Unfortunately, the model of Fig. 3.1 suffers from the lack of concurrency between simultaneous occurrences of the Read transition. Due to the enabling rule of CP-nets, an occurrence step can only contain one instance of either the Read transition or the Update transition, because each occurrence of the transitions monopolises the token, hereby preventing simultaneous access. In the case of the Update transition, this is exactly what we want to model, but the Read transitions are not able to access the data simultaneously. We are not able to model the wanted concurrency—at least not in the straightforward way shown above. If we have an upper limit for the number of simultaneous reads we could, of course, create sufficient tokens for all of these and then let each Read access only a single token, while Update changes all tokens. This solution unfortunately increases the complexity of the net inscriptions and the model contains a limit to the number of simultaneous reads.

As illustrated above, this lack of appropriate modelling primitives has often resulted in descriptions with either reduced concurrency or increased complexity of the net structure and/or the net inscriptions. We therefore propose to extend CP-nets with a new kind of arcs called test arcs. In contrast to ordinary arcs, several test arcs are allowed to access the same tokens in the same occurrence step, but not in the same step as ordinary arcs access the tokens. Test arcs cannot change the marking of a place.

Besides being a convenient modelling primitive, test arcs also simplify the transformation of CP-nets with inhibitor arcs into CP-nets, as we shall see in sections 5 and 6.

We have chosen to draw test arcs as connectors with a cross bar at both ends of the connector instead of arrowheads, because they are easy to draw and they can easily be distinguished from both source and destination of ordinary arcs. The small database example of Fig. 3.1 is then changed into the model shown in Fig. 3.2. The model in Fig 3.2 now expresses the wanted functionality and concurrency. Several instances of the Read transition may occur in the same occurrence step, but not concurrent with the Up-

date transition since, due to the ordinary arc, it still needs to get exclusive access to the token.

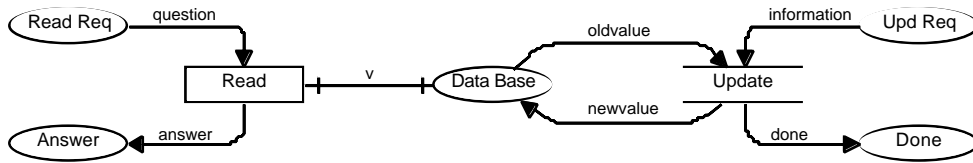


Figure 3.2: Small database example using test arcs

In the following section we formally define how CP-nets can be extended with test arcs. We also describe how a model with test arcs can be formally transformed into a behaviourally equivalent CP-net without test arcs. The idea behind the transformation is illustrated in Fig. 3.3. For each token in the original net we generate an infinite set of tokens in the equivalent CP-net by using the cross product with an infinite colour set Z . A test arc is then transformed into two ordinary arcs with opposite directions, which produce and consume only one single token, while ordinary arcs consume all tokens corresponding to the original token, i.e., an infinite set of tokens. In Fig. 3.3 the \times operator is used to denote the multi-set cross product and z denotes a variable of type Z .

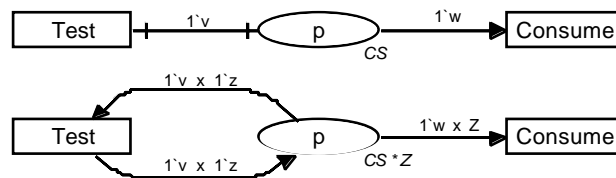


Figure 3.3: Transformation of test arcs

This means, that if the place p of the original net with test arcs is marked with, e.g., 2^a , the corresponding place in the transformed net is marked with $2^a(z_1) + 2^a(z_2) + 2^a(z_3) + \dots$, where z_1, z_2, z_3, \dots are the colours of the infinite colour set Z . The Test transition of the transformed net consumes and produces only one token, e.g., $1^a(z_7)$, whereas Consume removes $1^a(z_1) + 1^a(z_2) + 1^a(z_3) + \dots$. Having infinitely many different tokens in the marking of a place is not a theoretical problem, it corresponds to a PTnet with infinitely many places each having a finite marking.

4. Formal Definition of CP-nets with Test Arcs

In this section we show how test arcs are formally defined. In Def. 4.1 we give the structural definition of test arcs. In Def. 4.2 we define the additions to the enabling rule of CP-nets. Finally, in Def. 4.3, Lemma 4.4 and Theorem 4.5 we define the transformation of CP-nets with test arcs into behaviourally equivalent CP-nets.

4.1 The Definition of CP-nets with Test Arcs

Definition 4.1

A CP-net with test arcs is a tuple $CPN_T = (CPN, TS)$ satisfying the requirements below:

- (i) CPN is a non-hierarchical CP-net: (P, T, A, N, C, G, E, I) .
- (ii) TS is a **test arc specification**: (A_T, N_T, E_T) .
- (iii) A_T is a set of **test arcs**, such that: $A_T \cap (P \cup T \cup A) = \emptyset$.

Continues

- (iv) N_T is a **test node** function. It is defined from A_T into $P \times T$.
- (v) E_T is the **test expression** function. It is defined from A_T into expressions such that:
 - $a \in A_T: [\text{Type}(E_T(a)) = C(p(a))_{MS} \quad \text{Type}(\text{Var}(E_T(a)))]$
where $p(a)$ is the place of $N_T(a)$.

- (i) CPN is a CP-net, see Def. 2.1.
- (ii) TS is a triple, defined as follows:
- (iii) A_T is the set of test arcs. All test arcs are disjoint from other net elements.
- (iv) The test node function maps each test arc into a pair, where the first element is a place and the second a transition. Even though we use $P \times T$ it is important to remember that test arcs have no direction.
- (v) The type of the test expression of a test arc, a , must match the colour set of the place of a . All variables of a test expression must be of known colour sets. This is exactly the same demand as for ordinary arc expressions, Def. 2.1 (viii).

We extend the p and t functions so that they also handle test arcs. This can be done by means of the N_T function.

4.2 Behaviour of CP-nets with Test Arcs

Test arcs only influence the enabling of bindings, i.e., not the actual tokens moved when an enabled binding occurs. This means that we only need to specify the new enabling rule, and keep the occurrence rule of CP-nets unchanged.

Definition 4.2

A step $Y \in \mathcal{Y}$ is **enabled** in a marking $M \in \mathcal{M}$ iff the following properties are satisfied:

- (i) $\forall p \in P: \exists i \in \text{su}((t, b) \in Y, E(p, t) \langle b \rangle) \quad M(p)$.
- (ii) $\forall a \in A_T: \exists (t(a), b) \in Y: E_T(a) \langle b \rangle \quad M(p(a)) \quad \neg \exists i \in \text{su}((t', b') \in Y, E(p(a), t') \langle b' \rangle)$.

If Y fulfils (ii) it is said to be **test arc enabled**.

- (i) This is the enabling rule of CP-nets, See [Jen91] Def. 2.6 or [Jen92] Def. 2.8.
- (ii) For each test arc we demand that the required multi-set of tokens can be accessed, after the tokens used in the step have been consumed but before the tokens produced in the step have been added.

4.3 Behaviourally Equivalent CP-net

All CP-nets with test arcs can be transformed into behaviourally equivalent CP-nets. The idea behind the transformation was illustrated in section 3.

Definition 4.3

Let a CP-net with test arcs $CPN_T = (CPN, TS)$ be given with $CPN = (\mathcal{C}, P, T, A, N, C, G, E, I)$ and $TS = (A_T, N_T, E_T)$. Then we define the **equivalent CP-net** to be $CPN^* = (\mathcal{C}^*, P^*, T^*, A^*, N^*, C^*, G^*, E^*, I^*)$ where:

- (i) $\mathcal{C}^* = \{U \times \{0, 1, \dots, |U| - 1\} \mid U \in \mathcal{C}\} \cup \{Z\}$, where Z is an infinite colour set.
- (ii) $P^* = P$.
- (iii) $T^* = T$.
- (iv) $A^* = A \cup (A_T \times \text{Dir})$, where $\text{Dir} = \{\text{In}, \text{Out}\}$.

Continues

- (v) $C^*(a) = \{c \mid (a, \text{In}) \in A_T \times \text{Dir}, (t(a), p(a)), \text{iff } a = (a', \text{In}) \in A_T \times \text{Dir}, (t(a'), p(a')), \text{iff } a = (a', \text{Out}) \in A_T \times \text{Dir}\}$
- (vi) $C^*(p) = C(p) \times Z$
- (vii) $G^* = G$
- (viii) $E^*(a) = \{E(a), \text{iff } a \in A, E_T(a) \neq \emptyset, \text{iff } a = (a', d) \in A_T \times \text{Dir}\}$
where v_Z is a variable of type Z . v_Z must be unique for each test arc a .
- (ix) $I^*(p) = I(p) \times Z$

- (i) The set of colour sets is extended to include the colour sets determined by the cross product of each original colour set and a new infinite colour set called Z . Z itself is also included in the set of colour sets.
- (ii) The set of places is unchanged.
- (iii) The set of transitions is unchanged.
- (iv) The set of arcs is extended by arcs corresponding to the test arcs. Each test arc is substituted by a pair of ordinary arcs.
- (v) The node function is extended to handle the arcs corresponding to the test arcs. For each test arc, a' , we have an arc (a', In) having $p(a')$ as the source and $t(a')$ as the destination, and an arc (a', Out) having $t(a')$ as the source and $p(a')$ as the destination.
- (vi) The colour set of each place is the cross product of the old colour set and the new infinite colour set Z .
- (vii) The guard function is unchanged.
- (viii) The arc expressions are changed to match (vi). For the ordinary arcs we specify that the expressions must take a token for each colour in Z , i.e., an infinite set of tokens. The expressions corresponding to the test arcs will take only a single token. The tokens consumed by an arc (a', In) will have the same value as the tokens produced by (a', Out) .
- (ix) The initialization expression is changed to fit (vi).

Before we show that the transformation specified in Def. 4.3 preserves the behaviour of the CP-nets, we show a lemma that allows us to restrict the set of markings without neglecting any reachable marking. We show that for any reachable marking of a place we have a full set of Z tokens for each colour c , where (c, z) is a member of the marking.

Lemma 4.4

Let $CPN = (P, T, A, N, C, G, E, I)$ be a CP-net corresponding to the translation of a test arc specification.

Then we have the following property:

$$\forall p \in P, M \in [M_0], (c, z) \in M(p): M(p)(c, z) \geq 1 \iff c \in Z \implies M(p)(c, z) = 1$$

Proof: From Def. 4.3(vi) we know that the colour set of a place p has the structure $C \times Z$. The lemma holds for M_0 due to the transformation specified in Def. 4.3(ix). From Def. 4.3(viii) we have that each arc corresponding to an ordinary arc moves a full set of Z tokens, while the arcs corresponding to a test arc do not change the marking. This means that all reachable markings will fulfil lemma 4.4.

We are now ready to show that the transformation specified in Def. 4.3 preserves the behaviour. \mathbb{M} denotes the set of all markings and \mathbb{Y} denotes the set of all bindings, see ([Jen92], Def. 2.13). We use $M \times Z$ to denote the marking, which for each place p , has the tokens: $M(p) \times Z$. We use $Y \setminus V_Z$ to denote the restriction of a binding element $Y = (t, b)$ where variables of colour set Z are discarded, i.e., $Y \setminus V_Z = (t, b \setminus V_Z)$. All concepts with a

* refer to CPN^* , while those without refer to CPN_T . M^+ denotes the subset of all markings, M^* , fulfilling the restriction of lemma 4.4.

Theorem 4.5

Let CPN_T be a CP-net with test arcs and let CPN^* be the equivalent CP-net. Then we have the following properties:

(i) $M^+ = \{M \times \mid M \in M\} \quad M_0^* = M_0 \times .$

(ii) $Y = \{Y^* \setminus V_Z \mid Y^* \in Y^*\}.$

(iii) $M_1, M_2 \in M, \quad Y^* \in Y^*:$

$$M_1 \times [Y^* \rangle_{CPN^*} M_2 \times \quad M_1 [Y^* \setminus V_Z \rangle_{CPN_T} M_2.$$

$$M_1, M_2 \in M, \quad Y \in Y:$$

$$M_1 [Y \rangle_{CPN_T} M_2 \quad Y^* \in Y^*: Y = Y^* \setminus V_Z \quad M_1 \times [Y^* \rangle_{CPN^*} M_2 \times .$$

Proof: A detailed proof can be found in [CD92].

5. Informal Introduction to Inhibitor Arcs

The convenience of inhibitor arcs providing the ability to test for the absence of tokens has been known in the Petri net area for a long time. As in the case of place capacities, a number of different inhibitor concepts have been proposed, reflecting the fact that they are useful for different modelling purposes—see, e.g., [Bil88] or [CDF91]. Nevertheless, the concepts seem to be generalisations of the P/T inhibitor concepts to cover high-level Petri nets, e.g., zero-testing inhibitor arcs and threshold inhibitor arcs. A transition t connected to a place p by a *zero-testing inhibitor arc* can only occur if the marking of p is empty. If t and p are connected by a *threshold inhibitor arc* t can only occur if the marking of p is less than or equal to a specified multi-set called the threshold.

In this paper we improve and generalise the inhibitor arcs defined in [Bil88] in several ways. This section informally describes by means of small examples both the reasons why we find it necessary to change the concepts and the actual changes we propose. The generalised inhibitor arcs are formally defined in section 6.

Please note that we throughout this paper have chosen to draw inhibitor arcs as connectors with a circle at both ends because this notation ensures that inhibitor arcs are easily distinguished from both source and destination of ordinary arcs. Furthermore, this notation indicates that inhibitor arcs define an undirected relationship between places and transitions.

In [Bil88] the enabling rule defined to cover inhibitor arcs states that a multi-set of bindings is enabled if there are enough tokens on the input places, there is enough volume left in the capacity places and the inhibitor thresholds are not exceeded. Unfortunately this enabling rule does not preserve the diamond rule.

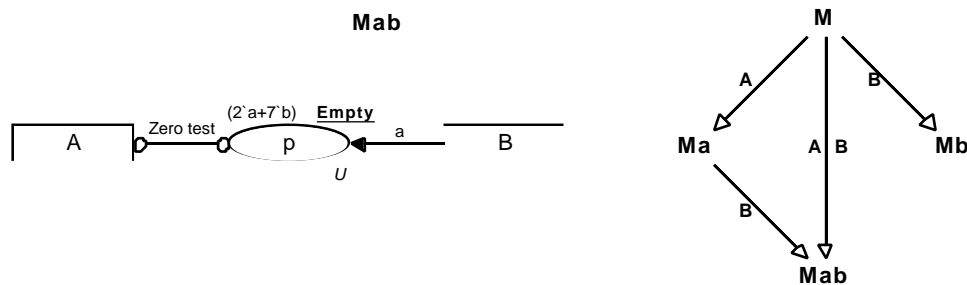


Figure 5.1: CP-net with inhibitor arcs violating the diamond rule

Let us assume that the inhibitor arc of Fig 5.1 is a zero-testing inhibitor arc, and the place p is unmarked. Then A and B are concurrently enabled according to the above en-

abling rule. If B occurs before A, A is no longer enabled, and the diamond rule is not fulfilled—an edge of the diamond is missing as illustrated in the diamond rule diagram of Fig. 5.1. To ensure that the diamond rule is preserved, we strengthen the enabling rule concerning inhibitors arcs. We demand that the threshold restrictions are valid for the current marking plus the tokens produced in the step. Hereby the diamond rule is satisfied since the concurrency causing the problem is transformed into a conflict—see Fig. 5.2.

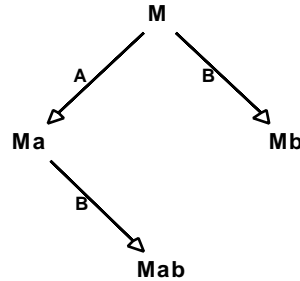


Figure 5.2: Diamond rule diagram for changed enabling rule

One may, of course, argue that this enabling rule is too restrictive from a modelling point of view, because models are hereby left out. It is, however, our belief that the diamond rule is such an essential and integral part of the Petri net framework and necessary in order to be able to understand the behaviour of CP-nets, that one must not deviate from it.

The next change we propose concerns the generality of the inhibitor concept. From our point of view there is no reason why the class of inhibitor arcs should be limited to zero-testing inhibitors and threshold inhibitors. We therefore propose to generalise the inhibitor concept along the line of the generalisation of the place capacity concept. We introduce a new generalised concept covering both zero-testing and threshold inhibitors. In order to be able to transform a CP-net with inhibitor arcs to a behaviourally equivalent CP-net in a straightforward way, we demand—like [Bil88]—that the place of an inhibitor arc must be a capacity place. The reason for this demand will be clear later when we discuss the transformation.

We suggest to define the inhibitor condition as an ordinary arc expression, except it must evaluate to a multi-set over the capacity colour set of the place the inhibitor arc is connected to. The inhibitor arc expression then expresses the threshold that must not be exceeded by the volume of the marking of the place.

In Fig. 5.3 we show how zero-testing and threshold inhibitor arcs are easily expressed by means of the new inhibitor concept. The place p of Fig. 5.3 is a capacity place with multi-set capacity $2^a + 7^b + 1^c$. Since the capacity projection is equal to the identity function the volume of the marking of p is equal to the marking of p . The threshold inhibitor is then expressed by defining the inhibitor arc expression equal to the threshold multi-set, i.e., $1^a + 2^b$. This means that the Threshold transition is only enabled if the marking of p is less than or equal to $1^a + 2^b$.

The zero-testing inhibitor is obtained by associating the empty multi-set, with the inhibitor arc. This means that the transition Zero-test is only enabled if the marking of p is the empty multi-set.

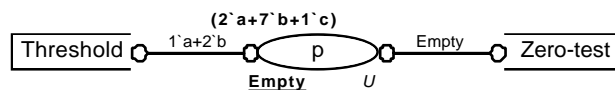


Figure 5.3: Zero-testing and threshold inhibitors expressed by means of inhibitor arc expressions

In [Bil88] it is pointed out several times that it is important to be able to transform proposed extensions of CP-nets into behaviourally equivalent CP-nets without the extensions. This is an obvious way of ensuring that the basic properties of CP-nets are pre-

served. Unfortunately, the transformation of inhibitor arcs proposed in [Bil88] does not preserve the entire behaviour. Only the single step behaviour—also called interleaving behaviour is preserved. This means that the behaviours of the two models are only equivalent as long as we consider steps without concurrency. The author is well aware of this problem, and it is suggested to be subject for further work.

There are two reasons for the difference in the behaviour of the two models in [Bil88]: the problem with the enabling rule illustrated above and the selected transformation. In [Bil88] the basic idea is to transform an inhibitor arc a , connected to a place p , into two ordinary arcs with opposite directions connected to the complementary place p' . The arc expressions of the arcs connected to p' are the capacity of p minus the arc expression of a . The transformation of the zero-testing inhibitor of Fig 5.1 is illustrated in Fig. 5.4.

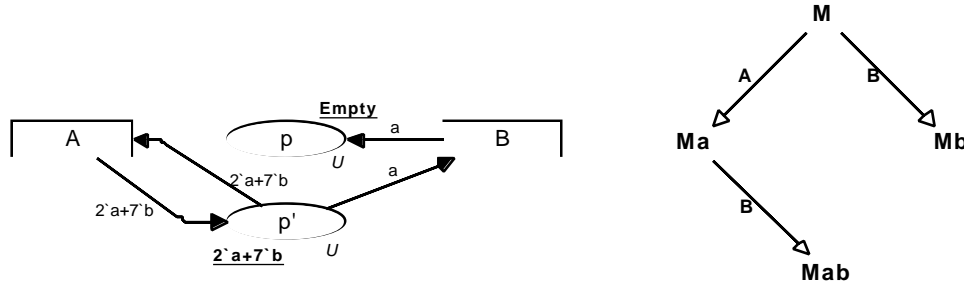


Figure 5.4: Transformation of zero-testing inhibitor arc and associated diamond rule

It is obvious from Fig. 5.1 and 5.4 that this transformation may limit the amount of concurrency. If p is unmarked the transitions A and B are concurrently enabled in the model of Fig. 5.1, but not in the transformed model in Fig 5.4. Since A and B concurrently access the same tokens a conflict arises between the transitions—a conflict that did not exist in the original model because of the enabling rule chosen in [Bil88]. If two transitions are connected to the same place by means of inhibitor arcs, the transformation may also cause the two transitions to be in conflict, because they—when transformed to ordinary arcs—try to access the same tokens in the same step. Fig. 5.1 and 5.4 also illustrate that the difference in behaviour results in different diamond rule diagrams.

Our solution to this problem is straightforward. We propose to use our strengthened enabling rule and to use test arcs in the transformation instead of two ordinary arcs with opposite directions. Hereby we avoid the problems described above. We propose to transform an inhibitor arc between a transition A and a place p into a test arc between A and the complementary place p' hereby ensuring that the concurrency of the model is preserved. Note that the arc expression of an inhibitor arc evaluates to a multi-set over the capacity colour set. The transformation of inhibitor arcs is illustrated in Fig. 5.5.

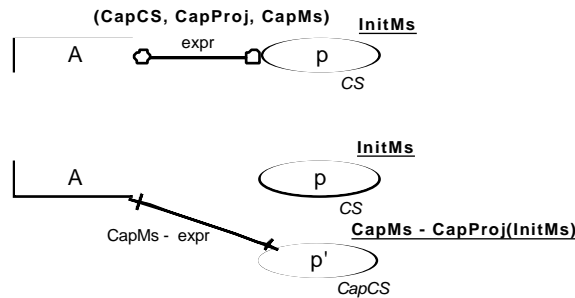


Figure 5.5: Transformation of an inhibitor arc to test arc connected to the complementary place

Fig 5.5 also illustrates why we demand that the place of an inhibitor arc must be a capacity place—otherwise it would be impossible to compute the initial marking of the complementary place and the test arc inscription.

Please note that our transformation—due to the use of test arcs—preserves the concurrency between two transitions connected to the same place by inhibitor arcs. This is illustrated in Fig. 5.6 showing the transformation of Fig. 5.3.

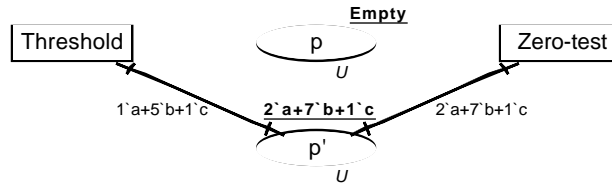


Figure 5.6: Transformation of Fig. 5.3

6. Formal Definition of CP-nets with Inhibitor Arcs

In contrast to the sections on place capacities and test arcs this section does not show how CP-nets with inhibitor arcs can be transformed directly into CP-nets. Instead we show how the same behaviour can be expressed using complementary places and test arcs as illustrated above.

6.1 Definition of CP-nets with Inhibitor Arcs

Since we demand that inhibitor arcs are connected to capacity places, inhibitor arcs are introduced into a framework where place capacity already is defined.

Definition 6.1

A CP-net with inhibitor arcs is a tuple $CPN_{CI} = (CPN, CS, IS)$ satisfying the requirements below:

- (i) CPN is a non-hierarchical CP-net: $(\ , P, T, A, N, C, G, E, I)$.
- (ii) CS is a capacity specification: (CN, CC, CP, CM) .
- (iii) IS is an **inhibitor arc specification**: (A_I, N_I, E_I) .
- (iv) A_I is a set of **inhibitor arcs**, such that: $A_I \cap (P \cup T \cup A) = \emptyset$.
- (v) N_I is the **inhibitor node** function. It is defined from A_I into $CN \times T$.
- (vi) E_I is the **inhibitor expression** function. It is defined from A_I into expressions such that:
 - $a \in A_I : [\text{Type}(E_I(a)) = CC(p(a))_{MS} \quad \text{Type}(\text{Var}(E_I(a))) \quad]$
where $p(a)$ is the capacity place of $N_I(a)$.

- (i) CPN is a CP-net, see Def. 2.1.
- (ii) CS is a capacity specification, see Def. 2.2.
- (iii) IS is an inhibitor arc specification, defined as a triple with the following elements:
- (iv) A_I is the set of inhibitor arcs. Inhibitor arcs are disjoint from other net elements.
- (v) The inhibitor node function maps each inhibitor arc into a pair, where the first element is a capacity place and the second a transition. Even though we use $CN \times T$ it is important to remember that inhibitor arcs have no direction.
- (vi) The type of the expression of an inhibitor arc, a , must match the capacity colour set of the place of a . All variables of an inhibitor arc expression must be of known colour sets.

We extend the p and t functions so that they also handle inhibitor arcs. This can be done by means of the N_I function.

6.2 Behaviour of CP-nets with Inhibitor Arcs

The behaviour of CP-nets with inhibitor arcs is described as follows:

Definition 6.2

A step $Y \in \mathcal{Y}$ is **enabled** in a marking $M \in \mathcal{M}$ iff the following properties are satisfied:

- (i) $\forall p \in P: \forall i \in \text{su}((t,b) \in Y, E(p,t) < b >) \quad M(p)$.
- (ii) $\forall p \in \text{CN}: \text{CP}(p)(M(p) + \forall i \in \text{su}((t,b) \in Y, E(t,p) < b >)) \leq \text{CM}(p)$.
- (iii) $\forall a \in A_I: \forall (t(a),b) \in Y: E_I(a) < b > \leq \text{CP}(p(a))(M(p(a)) + \forall i \in \text{su}((t',b') \in Y, E(t',p(a)) < b' >))$.

If Y fulfils (iii) it is said to be **inhibitor arc enabled**.

- (i) This is the enabling rule of CP-nets, See [Jen91] Def. 2.6 or [Jen92] Def. 2.8.
- (ii) The capacity enabling rule is defined in Def. 2.3.
- (iii) For each inhibitor arc we demand that the threshold expression evaluates to a multi-set which is larger than the volume of the marking together with the volume of the added tokens.

6.3 Behaviourally Equivalent CP-net

In sections 2.3 and 4.3 we have shown how to transform CP-nets with place capacities and CP-nets with test arcs into behaviourally equivalent CP-nets. We could specify a similar transformation for CP-nets with inhibitor arcs, but instead we take the easy way out. We show how to transform inhibitor arcs into test arcs connected to complementary places. This means that we must do the following: by means of Def. 2.3, we transform a CP-net with place capacities and inhibitor arcs into a CP-net with complementary places and inhibitor arcs. Then we use Def. 6.3, see below, to transform the inhibitor arcs into test arcs connected to the complementary places. Finally, we use Def. 4.3 to transform a CP-net with test arcs into an ordinary CP-net.

Definition 6.3

Let a CP-net with inhibitor arcs and complementary places $\text{CPN}_I = (\text{CPN}, \text{IS})$ be the result of the transformation of a CP-net having the capacity specification $\text{CS} = (\text{CN}, \text{CC}, \text{CP}, \text{CM})$. We use the following notation: $\text{CPN} = (\mathcal{P}, P, P', T, A, A', N, C, G, E, I)$ and $\text{IS} = (A_I, N_I, E_I)$. We define the equivalent CP-net with test arcs to be $\text{CPN}_T^* = (\text{CPN}^*, \text{TS})$ where $\text{TS} = (A_T, N_T, E_T)$ where:

- (i) $\text{CPN}^* = \text{CPN}$.
- (ii) $A_T = A_I$.
- (iii) $\tau(a) = (p', t)$ where $i(a) = (p, t)$.
- (iv) $E_T(a) = \text{CM}(p(a)) - E_I(a)$.

- (i) The basic net definition is unchanged.
- (ii) Each inhibitor arc is substituted by a test arc.
- (iii) The test node function maps a test arc into the pair consisting of the complementary place and the transition of the inhibitor arc.
- (iv) The expression of the test arc is the multi-set difference between the capacity multi-set and the inhibitor expression of the original arc.

Theorem 6.4

Let CPN_I be a CP-net with inhibitor arcs and complementary places and let CPN_T^* be the equivalent CP-net with test arcs. Then we have the following properties:

- (i) $M = M^* \quad M_0 = M_0^*$
- (ii) $Y = Y^*$.
- (iii) $M_1, M_2 \in M, \quad Y \in Y: M_1 [Y]_{CPN_I} M_2 \quad M_1 [Y]_{CPN_T^*} M_2$.

Proof: A detailed proof can be found in [CD92].

In the previous sections we have given an informal introduction to the notion of place capacity, test arcs and inhibitor arcs. We have also shown how these concepts can be formally defined in such a way that the extended CP-nets can be transformed into behaviourally equivalent CP-nets without extensions. For reasons of simplicity we have introduced the primitives one at a time. When modelling you would probably often want to use all three primitives. In the following section we have therefore shown how the three concepts can be combined into a single net formalism called CP-nets with place capacities, test arcs and inhibitor arcs.

7. CP-nets with place capacities, test arcs and inhibitor arcs

The three extensions described in the previous sections can easily be combined into a single model making it possible to model systems using place capacities, test arcs as well as inhibitor arcs.

Using the transformation defined in Def. 2.4, 4.3 and 6.3 we have shown, that a CP-net with place capacities, test arcs or inhibitor arcs can be transformed into a behaviourally equivalent CP-net. Since we from section 6.3 know that the transformation of inhibitor arcs uses both complementary places and test arcs, the straightforward order of transformation of a CP-net with both place capacities, test arcs and inhibitor arcs is: first place capacities are transformed into complementary places using Def. 4.2. Then we use Def. 6.3 to transform the inhibitor arcs into test arcs connected to the complementary places. Finally we use Def. 4.3 to transform the test arcs either remaining from the original net or resulting from the transformation of the inhibitor arcs into a CP-net.

8. Conclusion

In this paper we have extended CP-nets to include three new modelling primitives: place capacities, test arcs and inhibitor arcs, hereby improving the possibilities to create compact and understandable descriptions reflecting the modelled system in a straightforward way.

Our place capacity concept may be viewed as a generalisation of the previously proposed place capacity concepts: integer capacity, multi-set capacity and multi-set capacity with unbounded colours. In addition to these a number of other useful capacity restrictions can be formulated by means of our generalised concept.

Test arcs are—in contrast to ordinary arcs—able to access the same tokens simultaneously, but test arcs are not allowed to change the marking. Often the modelled system allows concurrent access to common data provided that the data is not changed. Without test arcs or similar modelling primitives this situation is difficult to model and therefore often results in models with either reduced concurrency or increased complexity in net structure and/or net inscriptions.

The definition of inhibitor arcs is also a generalisation of previously proposed inhibitor arc primitives. We associate an inhibitor arc expression with each inhibitor arc. The inhibitor arc expression is written like an ordinary arc expression. The result of

the evaluation of the inhibitor arc expression is denoted the threshold value and the associated transition is disabled if the threshold value is exceeded. A number of useful inhibitor conditions including zero-testing inhibitors and threshold inhibitors are easily expressed by means of our generalised inhibitor concept. To preserve the diamond rule—stating that steps can be split up into substeps—we need to strengthen the enabling rule known from previously proposed inhibitor concepts. We demand that the thresholds are not exceeded for the current marking and with the tokens produced in the step.

We have given an informal description of the new concepts, a formal definition and shown how they fit nicely into the Petri net framework. Both the static and dynamic properties of CP-nets with place capacities, test arcs and inhibitor arcs are formally defined. In order to ensure that the extensions do not violate the basic properties of CP-nets we have shown how CP-nets with place capacities, test arcs and inhibitor arcs can be transformed into behaviourally equivalent CP-nets. The possibility to transform a CP-net with the three new primitives to a behaviourally equivalent CP-net allows us to relate properties of the extended CP-nets to properties of the CP-nets. This is, e.g., useful when the analysis methods of CP-nets are generalised to cover extended CP-nets.

It should be noted, that place capacities, test arcs and inhibitor arcs from a theoretical point of view do not increase the computational power of CP-nets. It is, however, widely accepted that place capacities and inhibitor arcs are valuable modelling primitives. We furthermore believe that the inclusion of test arcs results in models reflecting the concurrency of the modelled system to a larger extent.

Acknowledgement

We would like to thank Kurt Jensen for valuable discussions and Peter Huber, Kjeld Høyer Mortensen, Mogens Nielsen and Laura Petrucci for many helpful comments on earlier versions of this paper.

References

- [Bil88] J. Billington: **Extending Coloured Petri Nets**. University of Cambridge, Computer Laboratory, Technical Report No. 148, October 1988. (Ph.D. Thesis).
- [CD92] S. Christensen, N. Damgaard Hansen: **Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs**. Daimi PB-398, ISSN 0105-8517, May 1992. Available as: Daimi PB-398, ISSN 0105-8517, May 1992.
- [CDF91] G. Chiola, S. Donatelli and G. Franceschinis. **Priorities, Inhibitor Arcs and Concurrency in P/T Nets**. Proceedings of the 12th International Conference on Application and Theory of Petri Nets, Aarhus 1991, 182-205.
- [Dev89] R. Devillers: **The Semantics of Capacities in P/T Nets**. In: G. Rozenberg (ed.): *Advances in Petri Nets 1989*. Lecture Notes in Computer Science, vol. 424. Springer-Verlag, 1989, pp. 128-150.
- [Gen86] H.J. Genrich: **Predicate/Transition nets**. In: W. Brauer, W. Reisig and G. Rozenberg (eds.): *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986 Part I*, Lecture Notes in Computer Science vol. 254, Springer-Verlag 1987, 207-247.
- [GLT79] H.J. Genrich, K. Lautenbach and P.S. Thiagarajan: **Elements of general net theory**. In: W. Brauer (ed.): *Net theory and applications. Proceedings of the Advanced Course on General Net Theory of Processes and Systems, Hamburg 1979*, Lecture Notes in Computer Science vol. 84, Springer-Verlag 1980, 21-163.
- [Jen91] K. Jensen: **Coloured Petri nets: A high level language for system design and analysis**. In: G. Rozenberg (ed.): *Advances in Petri Nets 1990*. Lecture Notes in Computer Science, vol. 383. Springer-Verlag, 1990, pp. 342-416.
- [Jen92] K. Jensen: **Coloured Petri nets. Basic concepts, analysis methods and practical use. Volume 1: Basic concepts**. EATCS monographs on Theoretical Computer Science, Springer-Verlag 1992.
- [Res85] W. Reisig: **Petri Nets. An introduction**. EATCS Monographs on Theoretical Computer Science, Vol. 4, ISBN 3-540-13723-8 or 0-387-13723-8, Springer-Verlag, 1985.

Content

Introduction.....	2
1. Informal Introduction to Place Capacities	2
2. Formal Definition of CP-nets with Place Capacities	6
2.1 The Definition of CP-nets with Place Capacities.....	6
2.2 Behaviour of CP-nets with Place Capacities.....	8
2.3 Behaviourally Equivalent CP-net.....	8
3. Informal Introduction to Test Arcs	10
4. Formal Definition of CP-nets with Test Arcs	11
4.1 The Definition of CP-nets with Test Arcs.....	11
4.2 Behaviour of CP-nets with Test Arcs.....	12
4.3 Behaviourally Equivalent CP-net.....	12
5. Informal Introduction to Inhibitor Arcs.....	14
6. Formal Definition of CP-nets with Inhibitor Arcs.....	17
6.1 Definition of CP-nets with Inhibitor Arcs.....	17
6.2 Behaviour of CP-nets with Inhibitor Arcs	18
6.3 Behaviourally Equivalent CP-net.....	18
7. CP-nets with place capacities, test arcs and inhibitor arcs.....	19
8. Conclusion.....	19
Acknowledgement.....	20
References	20