

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : GLSD 4/M2/2024



Mémoire

Présenté pour obtenir le diplôme de master académique en
Informatique

Parcours : Génie Logiciel et Systèmes Distribués (GLSD)

Titre

Reconfiguration Dynamique dans les Réseaux de Petri Colorés pour la Modélisation et L'analyse de Systèmes Flexibles

Présenté par
Kerfaoui Kheira

Soutenu le 11/06/2024 devant le jury composé de :

Guerrouf Faycal

MAA

Président

Tigane Samir

MAA

Rapporteur

Laanani Sadek

MAA

Examineur

Année universitaire 2023-2024

Abstract

Petri nets remain a powerful and efficient tool for modeling concurrent systems, but their limited ability to handle reconfigurable systems in their basic version has led to the development of extended versions. These extended versions offer solutions to overcome the initial limitations, enabling more efficient modeling and analysis of complex, dynamic systems. To overcome this weakness, several versions of extended Petri nets have been proposed : colored Petri nets, reconfigurable Petri nets and random Petri nets. These networks offer greater flexibility and the ability to model changes in system structure and behavior over time. Thanks to these improvements, analysts and engineers can obtain more accurate and realistic models of the systems they are working on. In this study, we present the definition of colored Petri net morphism and the application of Transformation rules in colored Petri nets whose structure can be dynamically modified. Rule-based modification of the network structure makes it easy to model complex and dynamic structures, such as reconfigurable manufacturing systems. The transformation approach is based on an algebraic transformation approach known as DPO (Double -PushOut).

Keywords : Petri nets, Colored Petri nets, Graphical Transformation Systems, Reconfigurable Petri nets.

Résumé

Les réseaux de Petri restent un outil puissant et efficace pour modéliser les systèmes concurrents, mais leur capacité limitée à gérer les systèmes reconfigurables dans leur version de base a conduit au développement de versions étendues. Ces versions étendues offrent des solutions pour surmonter les limitations initiales, ce qui permet de modéliser et d'analyser plus efficacement des systèmes dynamiques et complexes. Pour pallier cette faiblesse, plusieurs versions de réseaux de Petri étendus ont été proposées : les réseaux de Petri colorés, les réseaux de Petri reconfigurables et les réseaux de Petri aléatoires. Ces réseaux offrent une plus grande flexibilité et la possibilité de modéliser les changements dans la structure et le comportement des systèmes au fil du temps. Grâce à ces améliorations, les analystes et les ingénieurs peuvent obtenir des modèles plus précis et plus réalistes des systèmes sur lesquels ils travaillent.

Nous présentons dans cette étude la définition de morphisme des réseaux de Petri colorés et l'application de règle de Transformation dans les réseaux de Petri colorés dont la structure peut être modifiée de manière dynamique. La modification de la structure du réseau à l'aide de règles permet de modéliser facilement des structures complexes et dynamiques, telles que les systèmes de fabrication reconfigurables. L'approche de la transformation est basée sur une approche de transformation algébrique connue sous le nom de DPO (Double -PushOut).

Mots-clés : Réseaux de Petri, Réseaux de Petri Colorés, Systèmes de Transformation Graphique, Réseaux de Petri reconfigurables.

Remerciements

Avant tout, je merci ALLAH qui m'a aidé pour finir ce travail.

Je remercie toute ma famille où'ils ont été vraiment un grand soutien.

Je tiens à remercier mon encadreur Dr.Tigane Samir, pour sa méthode de travail ,son orientation ,ces conseils qui m'ont été d'une grande aide.

Je remercie enfin l'ensemble des membres du jury.

Dédicaces

À mes chers parents, pour leur amour inconditionnel, leur soutien et leurs sacrifices. Vous avez toujours cru en moi et m'avez guidé tout au long de ce voyage.

À mes sœurs (Karima, Messaouda) , pour leur patience, leur encouragement et leur affection. Vous avez été mes premières amies et mes plus grandes alliées.

À mon frère (Messaoud), pour sa camaraderie, ses conseils et son inspiration. Tu es un modèle de persévérance et de dévouement.

Merci à chacun de vous pour avoir rendu cette réussite possible. Cette étape importante de ma vie vous est dédiée.

Table des matières

Abstract	i
Résumé	ii
Remerciements	iii
Dédicaces	iv
1 Introduction générale	1
I L'état de l'art	6
2 Réseaux de Petri Colorés	7
2.1 Introduction	8
2.2 Notations	8
2.3 Réseaux de Petri	10
2.3.1 Formelle d'un Réseau de Petri	11
2.4 Modélisation graphique :[GUE11]	12
2.4.1 Places, Transitions et Arcs	12
2.4.2 Marquages	13
2.4.3 Franchissement de transitions	15
2.5 Propriétés des réseaux de Petri	17

2.6	Graphe des marquages et de couverture :[MAR]	19
2.6.1	Arbre et graphe des marquages accessibles	19
2.6.2	Arbre et graphe de couverture	20
2.7	Réseaux de Petri colorés	21
2.7.1	Formelle d'un Réseau de Petri coloré	22
2.8	Conclusion	23
3	Reconfiguration dans les Réseaux de Petri	25
3.1	Introduction	26
3.1.1	Systèmes de Transformation Graphique	27
3.1.2	Approche à Double-Pushout pour les réseaux de Petri	27
3.2	Systèmes de Réécriture du Réseau	31
3.3	Réseaux de Petri Reconfigurables	32
3.4	Conclusion	35
4	Reconfiguration dans les Réseaux de Petri Colorés	36
4.1	Introduction	37
4.2	Réseaux de Petri Colorés Reconfigurables	37
4.2.1	Définition des RecRdPCs	37
4.3	Exemple illustratif	43
4.4	Conclusion	48
5	Conclusion et Perspectives	49
	Bibliographie	51

Table des figures

2.1	Réseau de Petri.	11
2.2	Les matrices I , O et C	12
2.3	Places, transitions et arcs.	13
2.4	Exemple de Réseau de Petri.	14
2.5	Franchissement de transition.	16
2.6	Réseau bornés, non bornés.	17
2.7	Réseau vivant.	18
2.8	t_2 est quasi vivante.	18
2.9	M_1 est un blocage.	19
2.10	Réseau réinitialisable.	19
2.11	Arbre de marquage	20
2.12	Arbre de couverture	21
2.13	Réseau de Petri Coloré.	23
3.1	Morphismes RdP.	28
3.2	Diagramme DPO.	31
3.3	SRR règle r_0	33
3.4	Reconfiguration après application de r	34

4.1	Un morphisme RdPC valide.	39
4.2	Un morphisme RdPC invalide.	39
4.3	Diagramme DPO.	43
4.4	La nomenclature.	44
4.5	Le modèle RdPC pour le système à sa première configuration. . .	46
4.6	Le modèle RdPC pour la configuration du système C_1 lorsque la machine M3 tombe en panne.	47

Chapitre 1

Introduction générale

Introduction générale

Dans cette introduction, nous commençons par présenter le contexte de ce travail, à savoir Modélisation formelle basée sur des réseaux de Petri colorés à structure dynamique.

Nous nous concentrons ensuite sur la motivation de ce travail, en définissant la Problématique et les objectifs.

Enfin, nous terminons en décrivant le manuscrit d'organisation.

Contexte

À l'heure actuelle, chaque discipline des sciences de l'ingénieur vise à employer le langage de modélisation formelle pour décrire, analyser ou diagnostiquer les systèmes, dans le but de garantir leur fiabilité. L'une de ces sciences, le génie logiciel, possède une variété de langages de modélisation formelle : les réseaux de Petri, les automates, le pi-calcul, etc. Les réseaux de Petri sont utilisés dans cette famille pour représenter et étudier les systèmes. Les réseaux de Petri possèdent une catégorie qui peut être utilisée pour décrire la reconfiguration des systèmes, connue sous le nom de réseaux de Petri reconfigurables. Ces approches reposent sur l'approche algébrique Double-Pushout (DPO), qui a été développée dans le domaine de la grammaire des graphes et de la transformation des graphes.

Les réseaux de Petri reconfigurables et leurs types (haut niveau ou bas niveau), qui se composent d'un réseau de Petri et d'un ensemble de règles qui peuvent le modifier, se distinguent par la possibilité de distinguer différents niveaux de changement. Ils offrent un formalisme puissant et intuitif pour représenter des systèmes logiciels

ou matériels dynamiques qui sont exécutés dans des infrastructures en constante évolution. Ces infrastructures sont en constante évolution car elles sont également susceptibles de changer et car elles sont compatibles avec différentes applications qui peuvent partager certaines ressources. De plus en plus fréquents, ces systèmes logiciels ou matériels dynamiques présentent des difficultés à gérer. Pour modéliser et simuler des systèmes dynamiques, il est essentiel d'avoir un outil qui facilite la modélisation, la simulation et l'analyse des réseaux de Petri reconfigurables.

Problématique

La définition du morphisme dans les réseaux de Petri colorés et l'application des règles de transformation posent une problématique complexe et cruciale pour la modélisation précise des systèmes. Un morphisme doit préserver la structure et les propriétés des réseaux de Petri colorés tout en respectant les particularités des jetons colorés, assurant ainsi une correspondance fidèle entre le réseau source et le réseau cible. Parallèlement, les règles de transformation, qui déterminent comment les transitions peuvent se produire et comment les jetons se déplacent et se transforment au sein du réseau, doivent être appliquées de manière cohérente pour garantir la précision et la robustesse du modèle. La formalisation de ces concepts est essentielle pour exploiter pleinement les capacités des réseaux de Petri colorés, mais elle soulève des questions sur la manière de maintenir l'intégrité structurelle et comportementale du système lors des transformations. Ainsi, une problématique clé réside dans la recherche d'une méthode rigoureuse et systématique pour définir les morphismes et appliquer les règles de transformation tout en conservant les caractéristiques fondamentales des réseaux de Petri colorés.

Motivation et objectifs

La motivation principale de cette recherche est de fournir une compréhension approfondie et des outils efficaces pour la modélisation et l'analyse des systèmes complexes à l'aide des réseaux de Petri colorés. Le premier objectif est d'identifier et de formaliser le concept de morphisme adapté aux réseaux de Petri colorés, garantissant une correspondance structurelle et comportementale précise entre différents réseaux. Ensuite, il est essentiel de proposer des règles de transformation spécifiques à ces réseaux, prenant en compte la complexité ajoutée par les jetons colorés et leurs interactions. Ces règles doivent permettre des transitions et des mouvements de jetons qui préservent l'intégrité et les propriétés du système modélisé. Enfin, le développement d'un cadre théorique pour la reconfiguration des réseaux de Petri colorés est crucial. Ce cadre doit définir clairement les concepts et les méthodes nécessaires pour appliquer les morphismes et les règles de transformations de manière systématique et rigoureuse. L'objectif ultime est de faciliter la reconfiguration et l'adaptation des modèles basés sur les réseaux de Petri colorés, permettant ainsi une modélisation flexible et précise des systèmes dynamiques et complexes.

l'organisation

Ce travail se compose d'une première partie est organisé comme suit :

Chapitre 2 décrit et donne les concepts de base des réseaux de Petri et définit les réseaux de Petri colorés.

Chapitre 3 présente la relation entre Systèmes de transformation graphique et les

réseaux de Petri et donne un aperçu des réseaux de Petri reconfigurables.

Chapiter 4 Il fournit un cadre théorique pour les réseaux de Petri colorés reconfigurables.

Chapiter 5 Les résultats de l'étude permettent de créer un outil qui simplifie la modélisation, la simulation et l'analyse des réseaux de Petri colorés reconfigurables à Le futur.

Première partie

L'état de l'art

Chapitre 2

Réseaux de Petri Colorés

2.1 Introduction

Les réseaux de Petri, dès leur conception en 1962 par Carl Adam Petri, sont un outil très adapté à l'étude des systèmes à événements discrets en raison de leur capacité de modélisation et de leurs caractéristiques mathématiques. Les modèles obtenus, qui représentent de manière graphique la structure du système étudié, offrent la possibilité d'analyser les propriétés et donc de valider leurs spécifications[Atl12]. Principalement, les RdPs ont été employés afin de représenter des systèmes où certains événements peuvent se produire simultanément avec des contraintes sur la concordance, la priorité ou la fréquence de leurs événements[Pet77]. Les réseaux de Petri ont évolué vers les réseaux de Petri colorés, ce qui a élargi leur utilisation et les a rendu plus adaptés à la modélisation de systèmes complexes avec des données et des ressources spécifiques.

Les réseaux de Petri colorés constituent une catégorie captivante de réseaux de Petri de haut niveau. Leur particularité réside dans la teinte de chaque jeton. Ils proposent un modèle graphique parfait pour représenter les options, tout en créant des structures compactes et facilement reproduisables. Toutefois, leur vérification officielle comporte des contraintes, en raison de leur complexité [SOA23].

2.2 Notations

Dans ce travail, nous utilisons certaines notations pour définir les éléments des "Réseaux de Petri Colorés". Soit P est égal au domaine de la fonction V , qui représente l'ensemble des places. V est une fonction de domaine associée à P , et pour chaque p dans P , V_p est l'ensemble des valeurs ou des couleurs associées à p

(c'est-à-dire que V_p n'est pas vide).

Cela signifie que T est égal au domaine de la fonction I , qui est également égal au domaine de la fonction O , qui est également égal au domaine de la fonction F , et ces fonctions représentent toutes les ensembles des transitions.

La fonction I permet de déterminer le sous-ensemble de places d'entrée de chaque transition. Si $t \in T$ et $p \in I(t)$ alors p est une place d'entrée de t avec la multiplicité $I(t)(p)$. On peut considérer cette multiplicité comme le poids de l'arc reliant la place d'entrée p et la transition t . Une transition $t \in T$ est activée si chacune des places d'entrée contient au moins le nombre de jetons spécifié, c'est-à-dire que pour tout $p \in I(t)$: il y a au moins $I(t)(p)$ jetons dans p .

La fonction O permet de déterminer le sous-ensemble de places de sortie de chaque transition. Si $t \in T$ et $p \in O(t)$ alors p est une place de sortie de t avec la multiplicité $O(t)(p)$. On peut considérer cette multiplicité comme le poids de l'arc reliant la place de sortie p et la transition t . Une transition $t \in T$ est activée si chacune des places de sortie contient au moins le nombre de jetons spécifié, c'est-à-dire que pour tout $p \in O(t)$: il y a au moins $O(t)(p)$ jetons dans p .

Les opérations sur les ensembles sont désignées de la manière habituelle. Si P est un ensemble, $\#P$ est le nombre d'éléments de P et $\mathcal{P}(P)$ est l'ensemble des puissances de P (l'ensemble de tous les sous-ensembles de P).

L'ensemble de tous les jetons colorés possibles est défini par CT , telle que : $p \in P$ et $v \in V_p$.

Pour $\mathcal{B}(CT)$ est l'ensemble de tous les bags (multi-ensembles) possibles sur CT . Si CT est un ensemble, $\#CT$ est le nombre d'éléments de CT .

La fonction de transition F est définie pour chaque transition t dans T comme

suit : F_t appartient à $\mathcal{B}(CT) \rightarrow \mathcal{B}(CT)$, telle que :

- Le domaine de la fonction F_t est l'ensemble $\{c \in \mathcal{B}(CT) \mid \forall p \in P, (\sum_{v \in V_p} c(\langle p, v \rangle)) = I_t(p)\}$.
- Le domaine de la fonction F_t est également l'ensemble $\{c \in \mathcal{B}(CT) \mid \forall p \in P, ((\sum_{v \in V_p} c(\langle p, v \rangle)) = O_t(p))\}$.

$\mathcal{B}(CT) \rightarrow \mathcal{B}(CT)$:est une fonction partielle définie sur une partie d'un multi-ensemble.

Enfin, le marquage initial est noté M^0 .

$\text{dom}(F)$ dénote le domaine de la fonction F .

$\text{codom}(F)$ dénote le codomaine de la fonction F .

2.3 Réseaux de Petri

Un réseau de Petri, également connu sous le nom de réseau place/transition ou réseau P/T, est l'une des nombreuses représentations mathématiques des systèmes distribués discrets. En tant que langage de modélisation, il représente graphiquement la structure d'un système distribué sous la forme d'un graphe bipartite dirigé avec des annotations. Les réseaux de Petri ont été inventés en 1962 par Carl Adam Petri dans sa thèse de doctorat. Un réseau de Petri se compose de places, de transitions et d'arcs dirigés. Les arcs relient les places aux transitions, et non pas les places aux places ou les transitions aux transitions. Les places d'entrée d'une transition sont ceux qui sont connectés à la transition par un arc, tandis que les places de sortie sont ceux vers lesquels les arcs mènent à partir de la transition. Les places peuvent contenir un nombre quelconque de jetons. La répartition des jetons sur

les places d'un réseau s'appelle marquage. Les transitions opèrent sur les jetons d'entrée par un processus appelé "tir". Une transition est activée si elle peut être déclenchée, c'est-à-dire s'il y a des jetons dans chaque place d'entrée. Lorsqu'une transition se déclenche, elle consomme les jetons de ses places d'entrée, effectue une tâche de traitement et place un nombre déterminé de jetons dans chacun de ses places de sortie[Mur89].

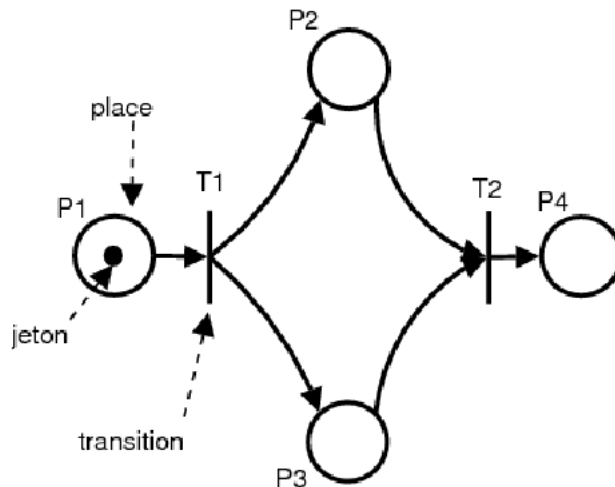


FIGURE 2.1 – Réseau de Petri.

2.3.1 Formelle d'un Réseau de Petri

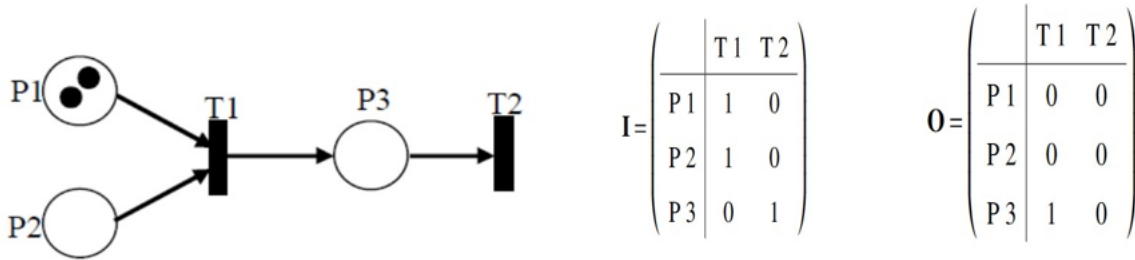
Définition 1 *Un réseau de Petri.* est défini par un 4-uplet, $RdP = \langle P, T, I, O \rangle$ où [Dar18] :

- P : Ensemble de places du réseau.
- T : Ensemble de transitions du réseau.
- $I : P \times T \rightarrow \mathbb{N}$ est la fonction input.
- $O : P \times T \rightarrow \mathbb{N}$ est la fonction output.

Définition 2 Les matrices I , O et C . sont définies comme suit [Bek09] :

- $I(p_i, t_j)$ est une matrice qui représente le poids de l'arc reliant p_i à t_j .
- $O(p_i, t_j)$ est une matrice qui représente le poids de l'arc reliant t_j à p_i .
- C est la matrice des incidences et est définie par :

$$C = O(p_i, t_j) - I(p_i, t_j).$$



$$C = \begin{pmatrix} -1 & 0 \\ -1 & 0 \\ 1 & -1 \end{pmatrix}$$

FIGURE 2.2 – Les matrices I , O et C .

2.4 Modélisation graphique :[GUE11]

2.4.1 Places, Transitions et Arcs

Un réseau de Petri (RdP) est un outil formel pour la modélisation et l'analyse des systèmes discrets. Il se compose de trois éléments fondamentaux :

1. Places :

Représentées graphiquement par des cercles. Stockent des jetons, symbolisés par des points noirs. Le nombre de jetons dans une place représente l'état actuel du système par rapport à cette ressource.

2. Transitions :

Représentées graphiquement par des rectangles ou des barres. Modélisent les événements ou actions qui modifient l'état du système. Activées par la présence de jetons dans les places d'entrée. Consomment des jetons des places d'entrée et en produisent pour les places de sortie.

3. Arcs :

Relient les places aux transitions et vice versa, mais jamais deux places ou deux transitions. Indiquent le flux de jetons entre les places et les transitions. Un arc peut être associé à un poids, indiquant le nombre de jetons requis pour activer la transition.

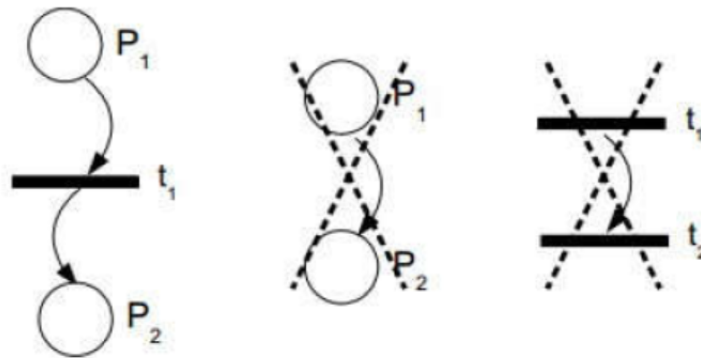


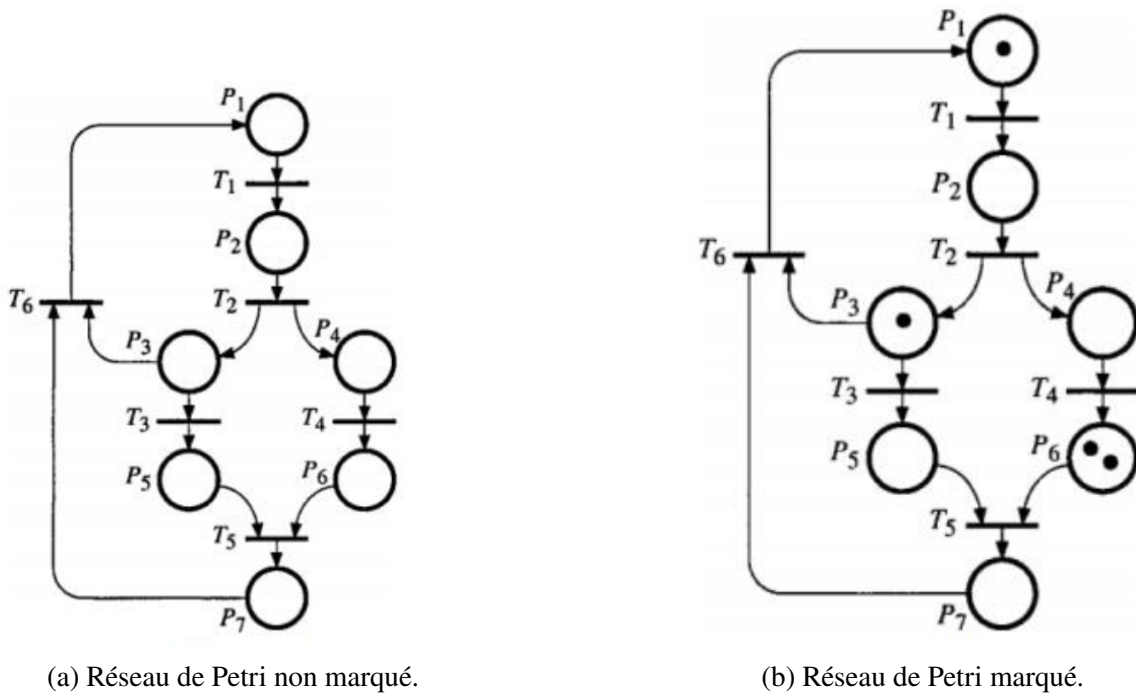
FIGURE 2.3 – Places, transitions et arcs.

2.4.2 Marquages

Soit $N = \langle P, T, I, O \rangle$ un réseau de Petri sans marquage initial. Un réseau de Petri marqué avec un marquage initial M^0 est noté $Rdp = \langle N, M^0 \rangle$. Le marquage M^0 est

un vecteur de dimension $|P|$, où $|P|$ est le nombre de places du réseau, et chaque composante M_i représente le nombre de jetons initialement présents dans la place P_i .

Exemple. Figure 2.4a : Le réseau de Petri représenté ne possède aucun jeton dans ses places. Il s'agit donc d'un réseau de Petri non marqué. Figure 2.4b : Le réseau de Petri représenté possède des jetons dans ses places. La distribution des jetons est donnée par le vecteur $M = (1, 0, 1, 0, 0, 2, 0)$. Il s'agit donc d'un réseau de Petri marqué avec un marquage initial $M^0 = (1, 0, 1, 0, 0, 2, 0)$.



(a) Réseau de Petri non marqué.

(b) Réseau de Petri marqué.

FIGURE 2.4 – Exemple de Réseau de Petri.

2.4.3 Franchissement de transitions

Une transition t est dite franchissable (sensibilisée) pour un marquage M si et seulement si la condition suivante est vérifiée pour toutes les places p du réseau de Petri :

$$M(p) \geq I(p, t)$$

où :

$M(p)$ représente le nombre de jetons dans la place p pour le marquage M . $I(p, t)$ représente le poids de l'arc reliant la place p à la transition t .

1. Notations du franchissement :

Le franchissement de la transition t peut être exprimé par les notations suivantes :

$$M \geq I(., t)$$

2. Conséquence du franchissement :

Si la transition t est franchissable pour le marquage M , son franchissement conduit à un nouveau marquage M' défini par :

$$M'(p) = M(p) - I(p, t) + O(p, t)$$

pour toutes les places p du réseau de Petri.

où :

$O(p, t)$ représente le poids de l'arc reliant la transition t à la place p .

3. Interprétation du franchissement :

Le franchissement d'une transition t dans un réseau de Petri représente une action indivisible qui consiste à :

Retirer un nombre de jetons égal au poids de l'arc reliant chaque place amont p à la transition t . Ajouter un nombre de jetons égal au poids de l'arc reliant la transition t à chaque place aval p' .

4. Condition nécessaire au franchissement :

La notion de franchissabilité exprime une condition nécessaire à l'exécution d'une action dans un réseau de Petri. Cette condition garantit que les ressources requises par la transition sont disponibles avant son déclenchement.

5. Exemple de franchissement :

illustre les règles de franchissement et de sensibilisation des transitions en utilisant la réaction chimique $2H_2 + O_2 \rightarrow 2H_2O$. Dans cet exemple, la transition t est franchissable pour le marquage initial car il y a deux jetons dans chacune des places d'entrée (H_2 et O_2). Le franchissement de la transition t entraîne la consommation de deux jetons de H_2 et d'un jeton de O_2 , et la production de deux jetons de H_2O . Le nouveau marquage reflète ces changements.

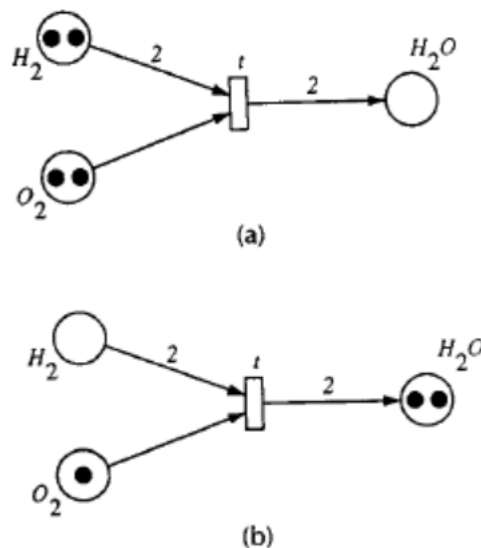


FIGURE 2.5 – Franchissement de transition.

2.5 Propriétés des réseaux de Petri

Les RdP présentent un riche ensemble de propriétés intéressantes pour modéliser les systèmes manufacturiers et le partage des ressources. Nous présenterons dans la suite les propriétés qui nous seront utiles dans le cadre de ce travail [Rez16] :

1. **RdP Borné** : Une place $p_i \in P$ dans un Réseau de Petri est dite k -bornée, avec $k \in \mathbb{N}$, pour un marquage initial M^0 donné, si $M(p_i) \leq k$ pour tout état accessible $M \in M$. Le Réseau de Petri est appelé borné si toutes ses places sont bornées.

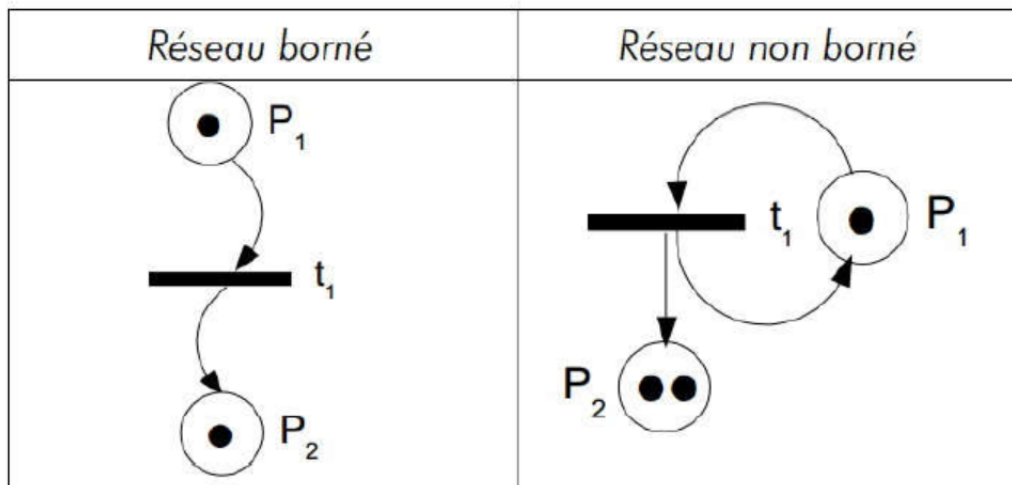


FIGURE 2.6 – Réseau bornés, non bornés.

2. **Vivacité** : Un Réseau de Petri est vivant si chacune de ses transitions est vivante, c'est-à-dire que, quel que soit le marquage M , il existe toujours un marquage atteignable à partir de M qui permet de franchir K .

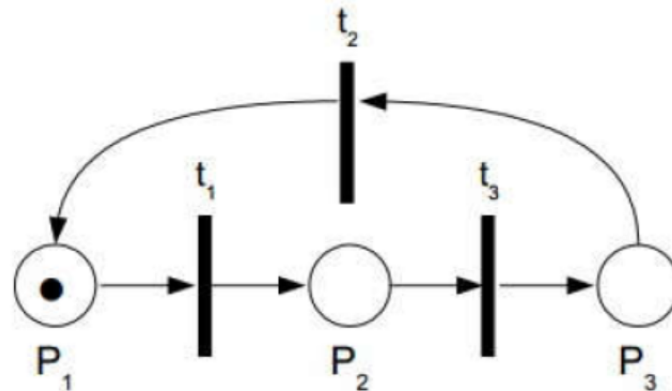
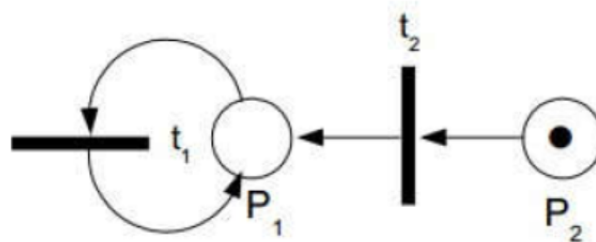


FIGURE 2.7 – Réseau vivant.

3. **Quasi-vivacité** : La quasi-vivacité établit une propriété moins contraignante que la vivacité. Alors que la vivacité exige que la transition soit franchissable à partir de tout marquage, la quasi-vivacité demande simplement l'existence d'une séquence de transition permettant de franchir t_j , depuis le seul marquage initial. Un réseau dans lequel toutes les transitions sont quasi-vivantes est dit quasi-vivant. On peut donc dire simplement qu'un tel réseau ne comporte pas de branches mortes pour le marquage initial, il existe toujours au moins un moyen de franchir chaque transition en partant de M^0 .

FIGURE 2.8 – t_2 est quasi vivante.

4. **Blocage d'un RdP** : Un blocage est un marquage tel qu'aucune transition ne peut être validée.

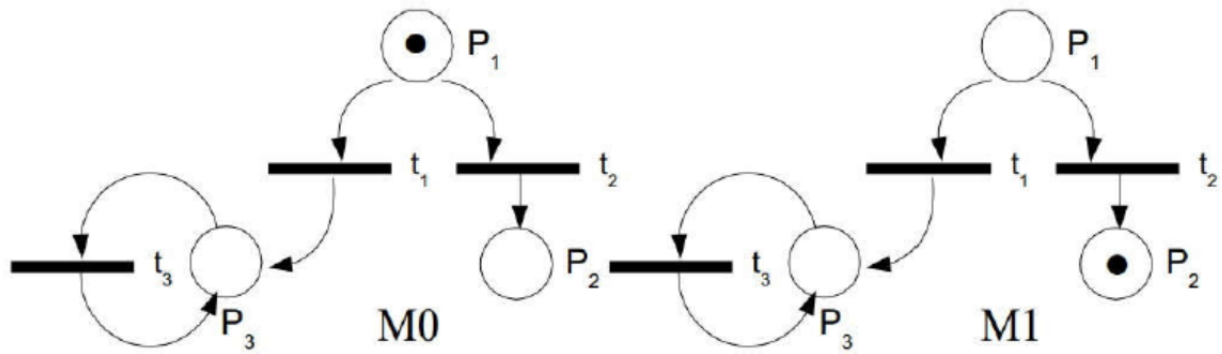


FIGURE 2.9 – M1 est un blocage.

5. **RdP Réversible** : Un Réseau de Petri est réversible si, à partir de n'importe quel état atteignable M, il existe une séquence franchissable qui permet de revenir à M^0 .

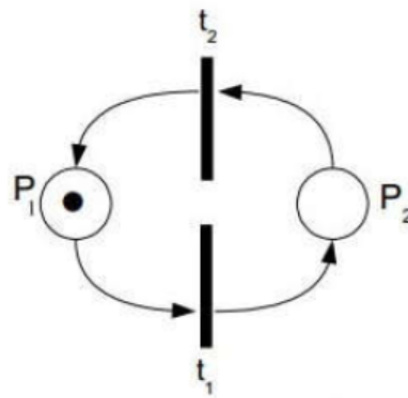


FIGURE 2.10 – Réseau réinitialisable.

2.6 Graphe des marquages et de couverture :[MAR]

2.6.1 Arbre et graphe des marquages accessibles

Première méthode : Construire le graphe des marquages accessibles de manière exhaustive. Cette méthode est très simple.

Les nœuds du graphe correspondent aux différents marquages possibles à partir de M_0 . On commence avec le seul marquage initial, puis on construit progressivement les arcs et les nœuds. Pour chaque nouveau marquage, on détermine l'ensemble des transitions possibles, et pour chaque transition de cet ensemble, on ajoute un arc menant au nouveau marquage. Si un marquage existe déjà dans le graphe, on se contente d'ajouter l'arc.

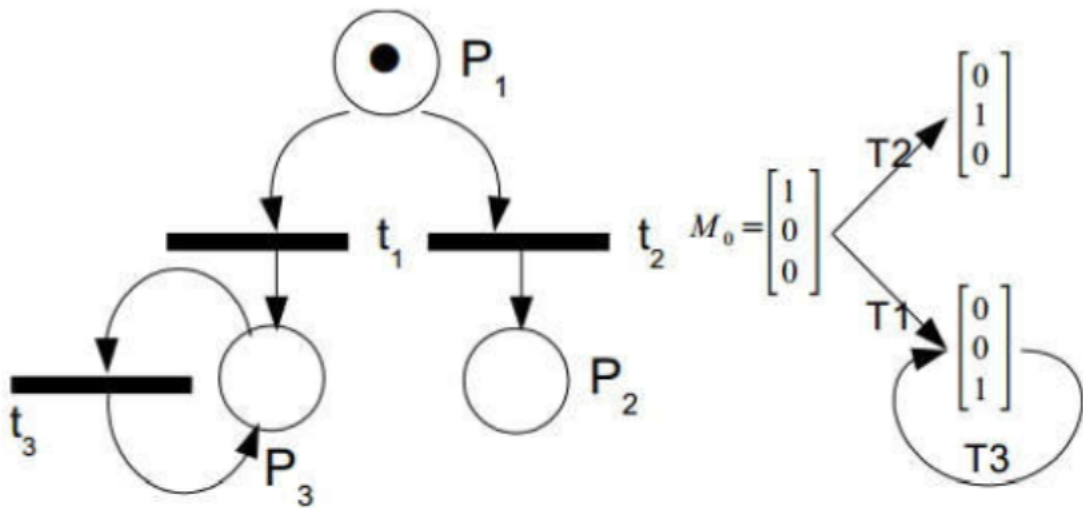


FIGURE 2.11 – Arbre de marquage

2.6.2 Arbre et graphe de couverture

L'inconvénient de la méthode précédente est majeur : tous les réseaux ne possèdent pas un graphe des marquages accessibles fini.

Ainsi, une méthode alternative a été proposée. Le mécanisme de construction est similaire à celui du graphe des marquages accessibles, mais chaque nouveau marquage (nœud du graphe) ajouté est vérifié pour s'assurer qu'il n'est pas supérieur à un marquage déjà existant sur au moins une séquence entre M^0 et le nouveau marquage.

Si tel est le cas, tous les marquages de places supérieurs sont remplacés par le symbole ω , qui indique que la place en question peut contenir un nombre infini de jetons (elle est donc non bornée).

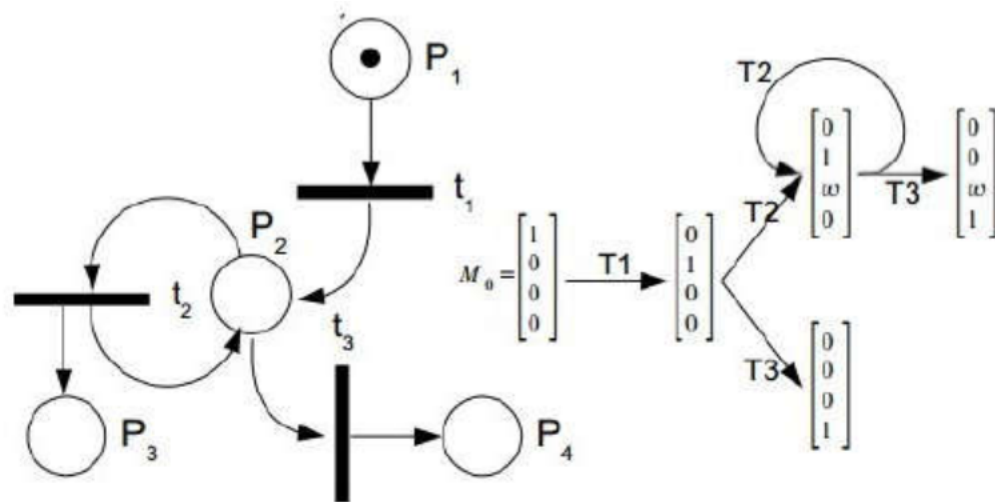


FIGURE 2.12 – Arbre de couverture

Dans ce qui suit, les places non bornées restent invariablement non bornées, quelles que soient les transitions effectuées; ainsi, le symbole ω persiste en tout temps.

Cette méthode génère le graphe de couverture, qui est toujours fini.

Tout comme avec le graphe des marquages accessibles, l'observation du graphe de couverture permet d'extraire un certain nombre de propriétés sur le réseau de Petri.

2.7 Réseaux de Petri colorés

Un modèle de réseau de Petri appelé réseau de Petri coloré (RdPC) a été introduit afin de résoudre le problème de complexité des réseaux de Petri (un modèle RdP d'un système donné a tendance à devenir trop complexe, que ce soit la taille du

modèle ou l'analyse, même pour des systèmes de taille modeste). Le principe d'un modèle RdPC est d'étendre la notion de jeton avec une couleur, par commodité nous utilisons la notation de (c) (la couleur du jeton). Ça permet de décrire des processus similaires de manière uniforme et succincte sans perdre la capacité de les distinguer.

2.7.1 Formelle d'un Réseau de Petri coloré

Définition 3 *Un RdPC est défini par 8-uplet, $RdPC = (P, V, T, I, O, CT, F, M^0)$ avec [VdA92] :*

- $P = \text{dom}(V)$, l'ensemble des places.
- V est une fonction de domaine P , pour tout $p \in P$,
 V_p est l'ensemble des valeurs ou des couleurs de p ($V_p \neq \emptyset$).
- $T = \text{dom}(I) = \text{dom}(O) = \text{dom}(F)$, l'ensemble des transitions.
- $I \in T \rightarrow \mathcal{P}(P)$, les places d'entrée d'une transition et leurs poids.
- $O \in T \rightarrow \mathcal{P}(P)$, les places de sortie d'une transition.
- $CT = \{(p, v) \mid p \in P \wedge v \in V_p\}$, l'ensemble de tous les jetons colorés possibles
 .
- F est la fonction de transition, pour tout $t \in T$, $F_t \in B(CT) \rightarrow B(CT)$, telle que :

$$\text{dom}(F_t) = \{c \in \mathcal{B}(CT) \mid \forall p \in P \left(\sum_{v \in V_p} c(\langle p, v \rangle) \right) = I_t(p)\}$$

$$\text{dom}(F_t) = \{c \in \mathcal{B}(CT) \mid \forall p \in P \left(\sum_{v \in V_p} c(\langle p, v \rangle) \right) = O_t(p)\}.$$

- M^0 un marquage initial.

Exemple. Dans cet exemple, la place initial (Ready To React) contient trois jetons H et deux jetons O, qui sont étiquetés en haut à droite de la place. L'opérateur ++ est utilisé pour représenter une opération d'union de plusieurs ensembles. Le poids sur l'arc signifie que X sont nécessaires pour que la transition (Molecules React) soit activée et déclenchée. Une fois la transition déclenchée, les jetons sont extraits de et deux jetons W, ou eau, sont placés dans la place (Resulting Compound).

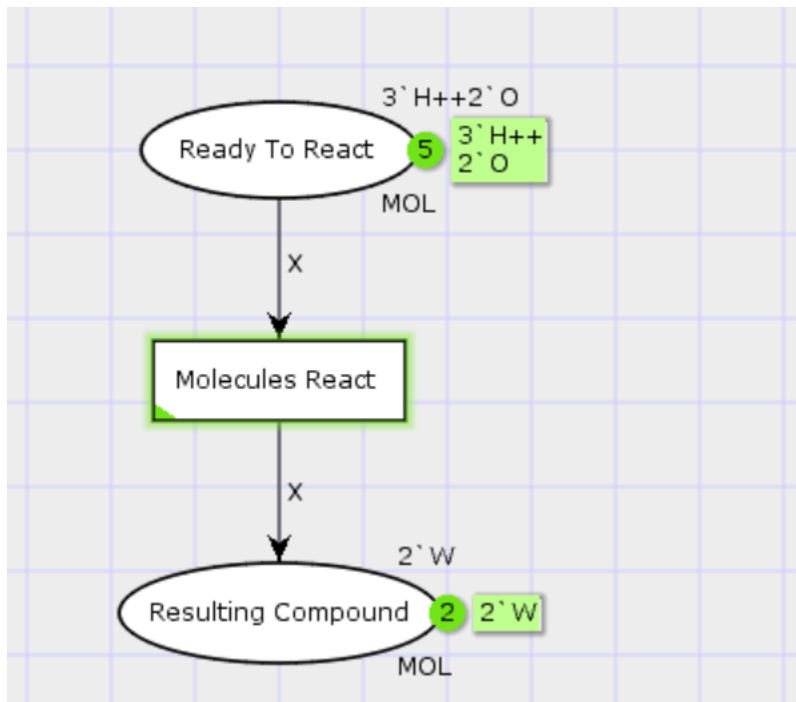


FIGURE 2.13 – Réseau de Petri Coloré.

2.8 Conclusion

Les concepts et techniques introduits dans ce chapitre, tels que la modélisation graphique, le franchissement, les propriétés, et le graphe des marquages et de couverture sont essentiels pour comprendre et utiliser les réseaux de Petri. Les extensions apportées par les réseaux de Petri colorés permettent d'aborder des systèmes

plus sophistiqués avec des exigences de modélisation plus élevées. Dans le chapitre suivant, nous nous intéressons à La reconfiguration dans les réseaux de Petri.

Chapitre 3

Reconfiguration dans les Réseaux de Petri

3.1 Introduction

Les réseaux de Petri reconfigurables, qui se composent d'un réseau de Petri et d'un ensemble de règles qui peuvent le modifier, ont la particularité de pouvoir différencier les différents niveaux de changement. Ils offrent des modèles puissants et intuitifs pour représenter des systèmes logiciels ou matériels en mouvement qui sont opérationnels dans des infrastructures en mouvement. Ces infrastructures sont en constante évolution car elles peuvent également évoluer et accueillir différentes applications qui peuvent partager certaines ressources. De tels systèmes logiciels ou matériels en constante évolution sont de plus en plus fréquents, mais leur gestion est complexe. La représentation de leurs processus et des changements de système au sein d'un même modèle sont indispensables pour la modélisation et la simulation de systèmes à structures dynamiques. Étant donné que le type de réseau de Petri sous-jacent peut être différent (telles que les réseaux de transition/-place, les réseaux d'objets, les réseaux temporisés et/ou stochastiques ou les réseaux de haut niveau), cette méthode peut être perçue comme une famille de techniques de modélisation formelle. Les réseaux reconfigurables de Petri sont une version des systèmes de transformation abstraits développés dans la théorie des catégories. Il s'agit d'identifier les catégories qui permettent des transformations en double-pushout : il ne faut donc que des descriptions schématiques. Cela présente l'avantage d'une théorie approfondie qui fournit de nombreux résultats relativement à la partie transformation.[PK18]

3.1.1 Systèmes de Transformation Graphique

Les systèmes de Transformation de Graphes (STGs) sont des outils puissants pour modéliser l'évolution des structures complexes. Ils ajoutent une dimension dynamique aux graphes en permettant la réécriture de règles pour représenter les changements structuraux. Les STGs trouvent des applications variées en informatique, notamment dans la modélisation de systèmes concurrents et distribués. Un STG comprend un graphe initial G_0 et un ensemble de règles de réécriture w pour reconfigurer le système. Chaque règle comporte une partie gauche (L) et une partie droite (R). Lorsque L est trouvé dans G, la règle est applicable, remplaçant L par R. Certains STG incluent un graphique d'interface pour illustrer les modifications et détailler les éléments à conserver.[Roz97]

3.1.2 Approche à Double-Pushout pour les réseaux de Petri

L'approche DPO s'appuie sur des morphismes graphiques pour établir des correspondances entre les transitions et les places de deux diagrammes. Si une place P est mappée à une place P' , il doit nécessairement exister une correspondance entre leurs pré-réglages et post-réglages respectifs, de manière analogue à la correspondance établie entre les transitions.

Définition 4 Morphisme RdP. Soit $G = (P_G, T_G, F_G, M_G^0)$ et $H = (P_H, T_H, F_H, M_H^0)$ deux RdPs. Un morphisme de réseau de Petri $\phi : G \rightarrow H$ est une paire de mappages $\phi_P : P_G \rightarrow P_H$ et $\phi_T : T_G \rightarrow T_H$, où $\forall p \in P_G$ et $\forall t \in T_G$, les conditions suivantes sont remplies [TIG20] :

$$— F_G(p, t) = F_H(\phi_P(p), \phi_T(t)) ,$$

- $F_G(t, p) = F_H(\phi_T(t), \phi_P(p))$,
- $M_G^0(p) \leq M_H^0(\phi_P(p))$.

Exemple. Considérons le RdPs indiqué dans Figure 3.1

Une occurrence de RdP L_0 est située dans H_0 de telle sorte que P'_0, T'_1, P'_1 et t'_2 dans L_0 sont mis en correspondance avec P_0, t_1, P_1 et t_2 dans H_0 . respectivement. D'autre part, le place P'_0 dans L_1 ne peut pas être mis en correspondance avec P_0 dans H_1 , puisque $ML_1(P'_0) \not\leq MH_1(P_0)$. De même, le place P'_1 et la transition de L_2 ne peuvent pas être mappés à la place P_1 et à la transition t_2 de H_2 , puisque $FL_2(P'_1, t'_2) \neq FH_2(P_1, t_2)$.

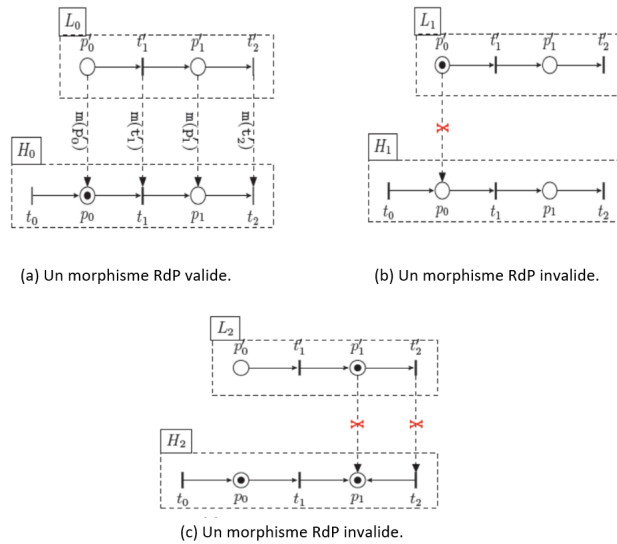


FIGURE 3.1 – Morphismes RdP.

Définition 5 Règle de Transformation. Les transformations dans DPO sont basées sur des règles w , ce qui signifie que chaque règle de transformation graphique comprend trois graphes L, I et R $w = \left(L \xleftarrow{\phi_l} I \xrightarrow{\phi_r} R \right)$; et deux morphismes de graphes, ϕ_l et ϕ_r , tels que [HEM05, EEGH15, TIG20] :

- L est désigné comme le côté gauche (une structure à supprimer),

- I comme l'interface commune,
- R comme le côté droit (une structure à insérer),
- $\phi_l : I \rightarrow L$ est un morphisme de graphes qui fait passer I à L ,
- $\phi_r : I \rightarrow R$ est un morphisme de graphes qui fait passer I à R .

En supposant une règle $w = \left(L \xleftarrow{\phi_l} I \xrightarrow{\phi_r} R \right)$, les composants de L qui n'appartiennent pas à l'image de ϕ_l sont dits obsolètes, tandis que les éléments de R qui n'appartiennent pas à l'image de ϕ_r sont dits frais. En appliquant la formule de manière informelle : $w = \left(L \xleftarrow{\phi_l} I \xrightarrow{\phi_r} R \right)$. L'ajout de certains éléments du côté droit R d'un graphe G à la place de ceux du côté gauche L est connu sous le nom d'ajout de R à un graphe G . pour commencer à appliquer w , nous localisons une correspondance de L dans G via un morphisme m . Lorsque les morphismes m , ϕ_l , et ϕ_r sont injectifs, nous éliminons les images des nœuds qui ne sont plus utilisés dans L (créant un réseau connu sous le nom de contexte) et nous ajoutons de nouveaux éléments de R . Cette transformation est représentée par $G \Rightarrow H$ par w et m , où H est le graphe résultant. Les termes "frais" et "obsolète" désignent respectivement les parties nouvellement ajoutées et supprimées.

Soit RdP $H = (P_H, T_H, F_H, M_H^0)$ défini comme suit :

1. $P_H = P_C \uplus (P_R \setminus P_I)$.
2. $T_H = T_C \uplus (T_R \setminus T_I)$.

3. F_H et M_H^0 sont donnés par les équations 3.1 et 3.2, respectivement

$$F_H(v, w) = \begin{cases} F_C(v, w) & \text{si } t \in T_C, \\ F_R(v, w) & \text{si } t \in T_R, \\ 0 & \text{autrement.} \end{cases} \quad (3.1)$$

$$M_H^0(p) = \begin{cases} M_C^0(p) & \text{si } p \in P_C \setminus P_R, \\ M_R^0(p) & \text{si } p \in P_R \setminus P_C, \\ M_C^0(p) + M_R^0(p) - M_I^0(p) & \text{autrement.} \end{cases} \quad (3.2)$$

Exemple.[TIG20] Une illustration d'une transformation RdP basée sur la DPO peut être trouvée à la figure 3.2.

En fait, Une transformation DPO se compose de deux constructions de collage de graphes, souvent appelées "pushouts". Le premier pushout (illustré par la boîte (1) de la figure 3.2) établit intuitivement l'existence d'un graphe (inconnu) C (appelé contexte) qui peut être collé avec L sur une interface commune I , le graphe résultant étant G , représenté par $G = C +_I L$. On peut appliquer une telle règle à G si ce graphe existe. En faisant adhérer C et R sur une interface commune I , le deuxième pushout (représenté par la boîte (2) dans la figure 3.2) calcule le graphe résultant H , qui est représenté par $H = C +_I R$.

Prise en compte de la règle illustrée à la figure 3.2. Une occurrence de L est localisée dans G par un morphisme RdP où $m(P0) = P0, m(t_1) = t_1, m(P1) = P1, et m(t_2) = t_2$. Alors, un réseau contextuel C est défini en supprimant les images des nœuds obsolètes de L (qui sont t_1, t_2) du réseau G . Ensuite, nous insérons de nouveaux nœuds de R (qui sont t_3) et des nœuds de C dans H . Enfin, la fonction de flux et le

marquage du nouveau réseau H sont calculés conformément aux équations 3.1 et 3.2, respectivement. Par exemple, (i) la connexion entre t_0 et p_0 dans H est égale à la connexion entre eux dans C , puisque t_0 et p_0 appartiennent tous deux au réseau C , (ii) la connexion entre t_3 et p_1 dans H est égale à la connexion entre eux dans R , puisque t_3 et p_1 appartiennent au réseau R ,
 (iii) $M_H^0(p) = M_C^0(p) + M_R^0(p) - M_I^0(p) = 1, etc.$

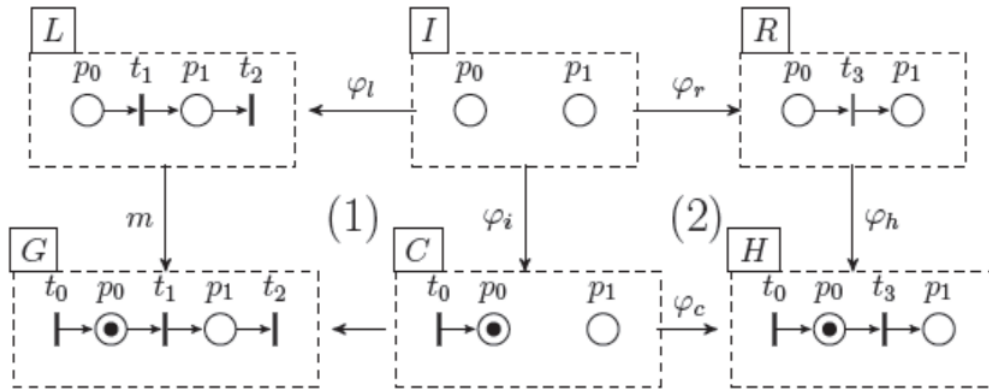


FIGURE 3.2 – Diagramme DPO.

3.2 Systèmes de Réécriture du Réseau

Avant d'appliquer une configuration, il est nécessaire de trouver un sous-réseau G' isomorphe au côté gauche de la règle. Afin d'activer la règle, il est nécessaire de vérifier qu'un élément x' ne fait partie du contexte que s'il est intégré à l'interface d'entrée de la règle, et vice versa, s'il est intégré à l'interface de sortie. Au cours de la réécriture, les jetons sont transférés de G' à R et le sous-réseau G' est remplacé par le côté droit R [BLO03].

Définition 6 *Systèmes de Réécriture du Réseau.* Un système de réécriture du réseau (SRR) est une structure $N = (R, (G_0, M_0))$, où $R = \{r_0, \dots, r_h\}$ est un ensemble

fini de règles de réécriture et (G_0, M_0) est un réseau de Petri marqué.

Définition 7 Règle de Réécriture. Une règle de réécriture $r \in R$ est une structure $r = \langle L, R, \bullet r, r \bullet \rangle$, où [LO04b] :

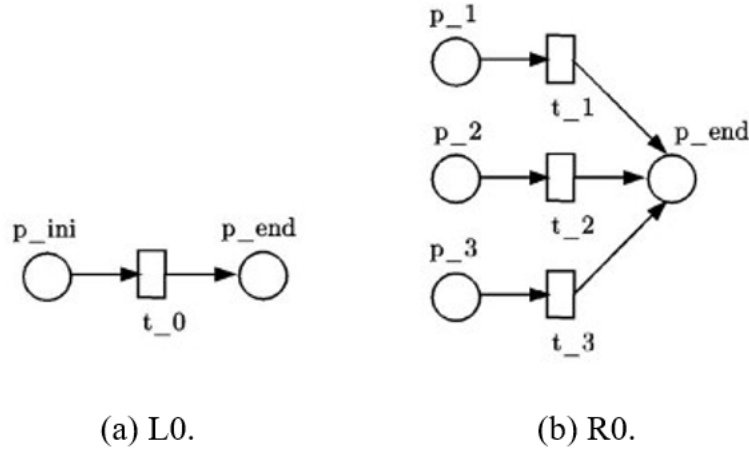
1. $L = \langle P_L, T_L, F_L \rangle$ et $R = \langle P_R, T_R, F_R \rangle$ sont des réseaux de Petri appelés respectivement le côté gauche et le côté droit de r , respectivement.
2. $r \subseteq (P_L \times P_R) \cup (T_L \times T_R)$ appelée relation de transfert de r , est une relation binaire reliant les places de L aux places de R et les transitions de L aux transitions de R ,
3. $\bullet r$ et $r \bullet$ sont des sous-relations de la relation de transfert appelées respectivement relation d'interface d'entrée et relation d'interface de sortie.

Exemple. Considérons la règle SRR r_0 décrite dans la Figure 3.3. Leurs L et R ne sont pas marqués RdP, respectivement. Ses relations d'interface concernent la transition $t-0$ aux transitions $t-1$, $t-2$ et $t-3$; et la place $p-ini$ à $p-1$, $p-2$ et $p-3$; place $p-end$ à $p-end$.

La relation de transfert est donnée par $r = \{(\{p-ini\}, \{p-1, p-2, p-3\}), (\{t-0\}, \{t-1, t-2, t-3\}), (\{p-end\}, \{p-end\})\}$ et les relations d'interface d'entrée et de sortie sont $\bullet r = \{(\{p-ini\}, \{p-1, p-2, p-3\})\}$ et $r \bullet = \{(\{p-end\}, \{p-end\})\}$, respectivement.

3.3 Réseaux de Petri Reconfigurables

Les réseaux de Petri qui peuvent être reconfigurés sont suggérés dans [LO04b] afin de faciliter la reconfiguration dans RdP. Il s'agit d'un ensemble de règles de réécriture qui ne modifient que les relations de flux (c'est-à-dire que les places et les

FIGURE 3.3 – SRR règle r_0 .

transitions sont toujours inchangées après l'application d'une règle). Afin de superviser la mise en œuvre des règles, les réseaux de Petri reconfigurables sont élargis aux réseaux de Petri reconfigurables contrôlés [LO04a], où l'activation d'une règle de réécriture est conditionnée par le marquage du réseau.

Définition 8 Réseaux de Petri reconfigurables. Une structure $N = \langle P, T, R, \gamma_0 \rangle$ est un MC-RPN [LO04a], où :

1. $P = \{p_0, \dots, p_n\}$ est un ensemble non vide et fini de positions,
2. $T = \{t_0, \dots, t_m\}$ est un ensemble non vide et fini de transitions,
3. $P \cap T = \emptyset$,
4. $R = \{r_0, \dots, r_k\}$ est un ensemble fini de règles de réécriture,
5. γ_0 est la configuration initiale du N .

Définition 9 Règles dans les RdPs. Une règle $r \in R$ est une structure $r = \langle D, \bullet r, r \bullet, C, M \rangle$, telle que :

1. $D \subseteq P$,

2. $\bullet r : (D \times T) \cup (T \times D) \rightarrow \mathbb{N}$ décrit les relations d'écoulement des places dans D avant d'appliquer r ,
3. $r \bullet : (D \times T) \cup (T \times D) \rightarrow \mathbb{N}$ décrit les relations d'écoulement des places dans D après application de r ,
4. $C \subseteq D$ est l'ensemble des places de contrôle,
5. M est le marquage minimum requis des places de C afin que la règle puisse être appliquée.

Exemple. La figure 3.4a montre un exemple d'RRdP N . Cet RRdP est défini comme $N = \langle P, T, R, \gamma_0 \rangle$, où [TIG20] :

- (1) $P = \{p_0, p_1\}$, $T = \{t_0, t_1, t_2\}$ et γ_0 comme le montre la figure 3.4a,
- (2) $R = \{r_0 = \langle D, \bullet r, r \bullet, C, M \rangle\}$ tel que :
 - $D = P$, $C = \{p_0\}$, $M(p_0) \geq 3$,
 - $\bullet r(t_i, p_i) = 1$ pour $i = 0, 1$, $r(p_j, t_{j+1}) = 1$ pour $j = 0, 1$ et, sinon, $\bullet r(.,.) = 0$,
 - $r \bullet(t_0, p_0) = 1$, $r \bullet(p_0, t_2) = 1$ et, sinon, $r \bullet(.,.) = 0$.

Les deux fonctions $\bullet r$ et $r \bullet$ peuvent être désignées simplement par : $p_0(t_0 - t_1) + p_1(t_1 - t_2) \Delta p_0(t_0 - t_2) + p_1(\emptyset)$. La figure 3.4b illustre la configuration obtenue après application de la règle r .

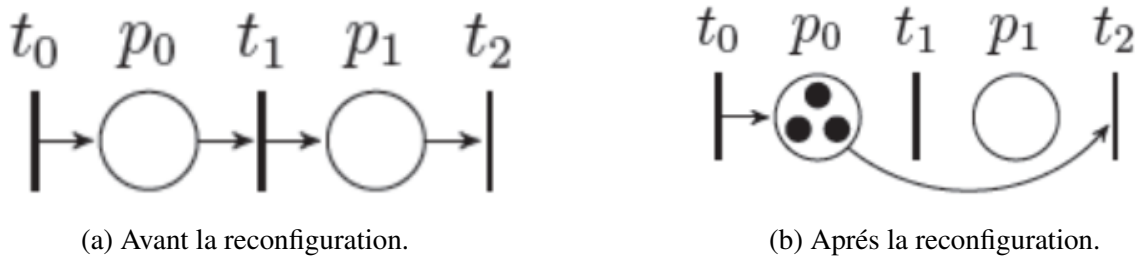


FIGURE 3.4 – Reconfiguration après application de r .

3.4 Conclusion

Dans ce chapitre, nous avons donné des détails sur les réseaux de Petri reconfigurables (Reconfigurable Petri Nets) et les systèmes de transformation graphique, en particulier l'approche à Double-PushOut (DPO), constituent des outils puissants pour la modélisation et l'analyse des systèmes dynamiques et réactifs. Dans le prochain chapitre, notre attention se porte sur la reconfiguration des réseaux de Petri colorés.

Chapitre 4

Reconfiguration dans les Réseaux de Petri Colorés

4.1 Introduction

Les Réseaux de Petri sont des outils puissants pour la modélisation et l'analyse de systèmes concurrentiels, distribués et asynchrones. Leur extension, les Réseaux de Petri Colorés (CPN - Colored Petri Nets), enrichit ces réseaux en introduisant des "couleurs" (ou types de données), permettant une modélisation plus compacte et expressive. Cependant, dans des systèmes complexes et dynamiques, les exigences peuvent évoluer avec le temps, nécessitant des modifications structurelles dans les modèles. C'est ici qu'interviennent les Réseaux de Petri Colorés Reconfigurables (RCPN - Reconfigurable Colored Petri Nets).

Un Réseau de Petri Coloré Reconfigurable combine les avantages des Réseaux de Petri Colorés et la flexibilité de la reconfiguration dynamique. Ils permettent non seulement de modéliser des systèmes complexes avec différentes entités (grâce aux couleurs) mais aussi de modifier la structure du réseau en cours d'exécution pour s'adapter aux changements dans le système.

4.2 Réseaux de Petri Colorés Reconfigurables

Un réseau de Petri coloré reconfigurable est une forme de réseau qui associe les caractéristiques des réseaux reconfigurables à l'utilisation de "couleurs" afin de gérer et d'optimiser les ressources du réseau de manière dynamique.

4.2.1 Définition des RecRdPCs

Un réseau de Petri coloré reconfigurable (RecRdPC) est un réseau de Petri coloré qui a des mécanismes qui lui permettent de modifier dynamiquement sa structure

et ses paramètres en fonction de certaines conditions ou événements. On peut faire cette reconfiguration en ajoutant, en supprimant ou en modifiant des places, des transitions et des arcs, ainsi que en modifiant les couleurs et les types de jetons.

Définition 10 Morphisme RdPC. Soit $G = (P_G, V_G, T_G, I_G, O_G, CT_G, F_G, M_G^0)$ et $H = (P_H, V_H, T_H, I_H, O_H, CT_H, F_H, M_H^0)$ deux RdPCs. Un morphisme de réseau de Petri Coloré $\phi : G \rightarrow H$ est une paire de mappages $\phi_P : P_G \rightarrow P_H$ et $\phi_T : T_G \rightarrow T_H$, où $\forall p \in P_G$ et $\forall t \in T_G$, les conditions suivantes sont remplies :

1. $V_p = V'_{\phi(p)}, \forall p \in P_G$,
2. $M(p, c) \leq M'(\phi(p), c), \forall p \in P_G$,
3. $\forall p \in I_t \Rightarrow \phi(p) \in I'_{\phi(t)}, \forall t \in T_G$,
4. $\forall p \in O_t \Rightarrow \phi(p) \in O'_{\phi(t)}, \forall t \in T_G$,
5. (a) $\forall t \in T_G, \forall c \in \text{dom}(F_t), \exists c' \in \text{dom}(F'_{\phi(t)}) : \forall p \in P :$

$$-c(\langle p, v \rangle) = c'(\langle \phi(p), v \rangle)$$

- (b) $\forall c_1 \in \text{codom}(F_t), \exists c'_1 \in \text{codom}(F'_{\phi(t)}) : \forall p \in P :$

$$-c_1(\langle p, v \rangle) = c'_1(\langle \phi(p), v \rangle)$$

- (c) $F_t(c) = c_1 \Rightarrow F_{\phi(t)}(c') = c'_1$.

Exemple. Considérons le RdPC indiqué dans Figure 4.1 .

Une occurrence de RdPC L0 est située dans H0 de telle sorte que P'0, T'1, P'1 et T'2 dans L0 sont mis en correspondance avec P0, T1, P1 et T2 dans H0.

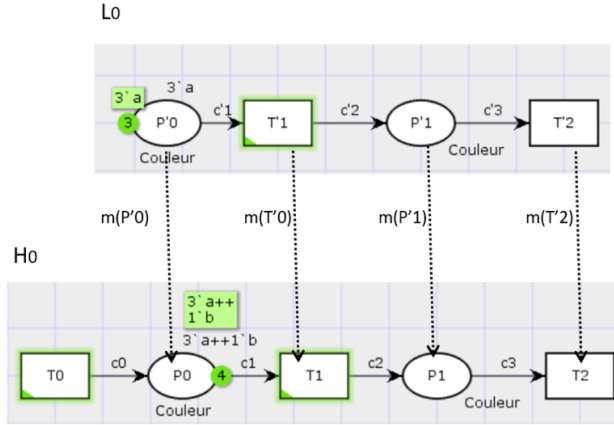


FIGURE 4.1 – Un morphisme RdPC valide.

respectivement. Considérons le RdPC indiqué dans Figure 4.2.

En revanche, le place P'0 dans L1 ne peut pas être mis en correspondance avec P0 dans H1, puisque $ML_1(P'0) \not\subseteq MH_1(P0)$.

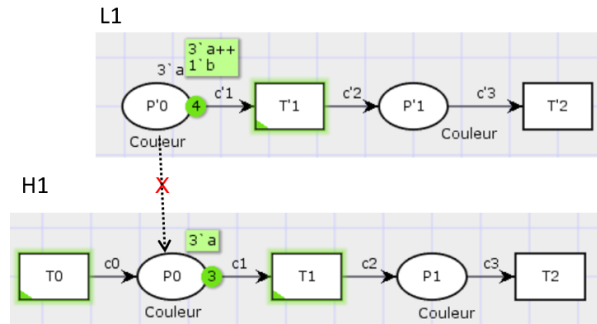


FIGURE 4.2 – Un morphisme RdPC invalide.

Définition 11 Règle de Transformation. Les transformations dans DPO sont basées sur des règles w , ce qui signifie que chaque règle de transformation graphique comprend trois RdPCs L , I et R $w = \left(L \xleftarrow{\phi_l} I \xrightarrow{\phi_r} R \right)$; et deux morphismes de RdPCS, ϕ_l et ϕ_r , tels que :

- L est désigné comme le côté gauche (une structure à supprimer),

- I comme l'interface commune,
- R comme le côté droit (une structure à insérer),
- $\phi_l : I \rightarrow L$ est un morphisme de RdPC qui fait passer I à L ,
- $\phi_r : I \rightarrow R$ est un morphisme de RdPC qui fait passer I à R .

En supposant une règle $w = \left(L \xleftarrow{\phi_l} I \xrightarrow{\phi_r} R \right)$, les composants de L qui n'appartiennent pas à l'image de ϕ_l sont dits obsolètes, tandis que les éléments de R qui n'appartiennent pas à l'image de ϕ_r sont dits frais. En appliquant la formule de manière informelle : $w = \left(L \xleftarrow{\phi_l} I \xrightarrow{\phi_r} R \right)$. L'ajout de certains éléments du côté droit R d'un RdPC G à la place de ceux du côté gauche L est connu sous le nom d'ajout de R à un RdPCG. pour commencer à appliquer w , nous localisons une correspondance de L dans G via un morphisme m . Lorsque les morphismes m , ϕ_l , et ϕ_r sont injectifs, nous éliminons les images des nœuds qui ne sont plus utilisés dans L (créant un réseau connu sous le nom de contexte) et nous ajoutons de nouveaux éléments de R . Cette transformation est représentée par $G \Rightarrow H$ par w et m , où H est le RdPC résultant. Les termes "frais" et "obsolète" désignent respectivement les parties nouvellement ajoutées et supprimées.

soit RdPC $H = (P_H, V_H, T_H, I_H, O_H, CT_H, F_H, M_H^0)$

1. $P_H = P_C \uplus (P_R \setminus P_I)$.
2. $V_H = V_C \uplus (V_R \setminus V_I)$.
3. $T_H = T_C \uplus (T_R \setminus T_I)$.
4. $I_H^t = \begin{cases} I_C^t & \text{si } t \in T_C \\ I_R^t & \text{si } t \in T_R \\ 0 & \text{autrement} \end{cases} \cdot$

$$5. O_H^t = \begin{cases} O_C^t & \text{si } t \in T_C \\ O_R^t & \text{si } t \in T_R \\ 0 & \text{autrement} \end{cases} .$$

6. CT_H est donné comme fourni dans la définition de RdPC 3.

7. F_H^t et M_H^0 sont donnés par les équations (4.1) et (4.2), respectivement.

$$F_H^t = \begin{cases} F_C^t & \text{si } t \in T_C \\ F_R^t & \text{si } t \in T_R \\ 0 & \text{autrement} \end{cases} \quad (4.1)$$

$$M_H^0(p, v) = \begin{cases} M_C^0(p, v) & \text{si } P \in P_C \setminus P_R \\ M_R^0(p, v) & \text{si } P \in P_R \setminus P_C \\ M_C^0(p, v) + M_R^0(p, v) - M_I^0(p, v) & \text{autrement} \end{cases} \quad (4.2)$$

Exemple. Dans la figure 4.3. $c, c1, c2, c3, c4, c5$ sont comme suit :

$c = \{ \{ \langle P0, \text{rouge} \rangle \langle P0, \text{rouge} \rangle \langle P0, \text{rouge} \rangle \langle P0, \text{bleu} \rangle \langle P0, \text{bleu} \rangle \} \}$,

$c1 = \{ \{ \langle P1, \text{bleu} \rangle \langle P1, \text{bleu} \rangle \langle P1, \text{bleu} \rangle \langle P1, \text{rouge} \rangle \langle P1, \text{rouge} \rangle \} \}$,

$c2 = \{ \{ \langle P2, \text{rouge} \rangle \langle P2, \text{bleu} \rangle \} \}$,

$c3 = \{ \{ \langle P2, \text{rouge} \rangle \langle P2, \text{bleu} \rangle \} \}$,

$c4 = \{ \{ \langle P0, \text{rouge} \rangle \langle P0, \text{bleu} \rangle \langle P0, \text{bleu} \rangle \langle P0, \text{orange} \rangle \} \}$,

$c5 = \{ \{ \langle P2, \text{noir} \rangle \langle P2, \text{noir} \rangle \langle P2, \text{vert} \rangle \langle P2, \text{vert} \rangle \langle P2, \text{jaune} \rangle \} \}$.

Une illustration d'une transformation RdPC basée sur la DPO peut être trouvée à la

figure 4.3. En fait, Une transformation DPO se compose de deux constructions de collage de RdPCs, souvent appelées "pushouts". Le premier pushout (illustré par l'encadré (1) de la figure 4.3) établit intuitivement l'existence d'un RdPC (inconnu) C (appelé contexte) qui peut être collé avec L sur une interface commune I , le RdPC résultant étant G , représenté par $G = C +_I L$. On peut appliquer une telle règle à G si ce RdPC existe. En faisant adhérer C et R sur une interface commune I , le deuxième pushout (représenté par la boîte (2) dans la figure 4.3) calcule le RdPC résultant H , qui est représenté par $H = C +_I R$.

Considérons la règle illustrée à la figure 4.3. Une occurrence de L est localisée dans G par RdPC, un morphisme m , où : $m(P0) = P'0$, $m(T0) = T'0$, $m(P1) = P'1$, $c_L = c_G$, $c_{L1} = c_{G1}$. Ensuite, un réseau contextuel C est défini en supprimant les images des nœuds obsolètes de L (qui sont $\{T'0, c, c1\}$) du réseau G . Ensuite, nous insérons de nouveaux nœuds de R (qui sont $\{P2, T1, c2, c3\}$) et des nœuds de C dans H . Enfin, la fonction de flux et le marquage net du nouveau réseau H sont calculés selon les équations 4.1 et 4.2, respectivement.

- (i) La connexion entre $P'0, T'2, P'3, c4$ et $c5$ dans H est égale à la connexion entre eux dans C , puisque $P'0, T'2, P'3, c4$ et $c5$ appartiennent tous deux au réseau C .
- (ii) La connexion entre $T'1, c2, P2, c3$ dans H est égale à la connexion entre eux dans R , puisque $T'1, c2, P2$ et $c3$ appartiennent tous deux au réseau R .
- (iii) $M_H^\circ(p) = M_C^\circ(p) + M_R^\circ(p) - M_I^\circ(p) = \langle 3'rouge + 2'bleu + 1'jaune \rangle$, etc.

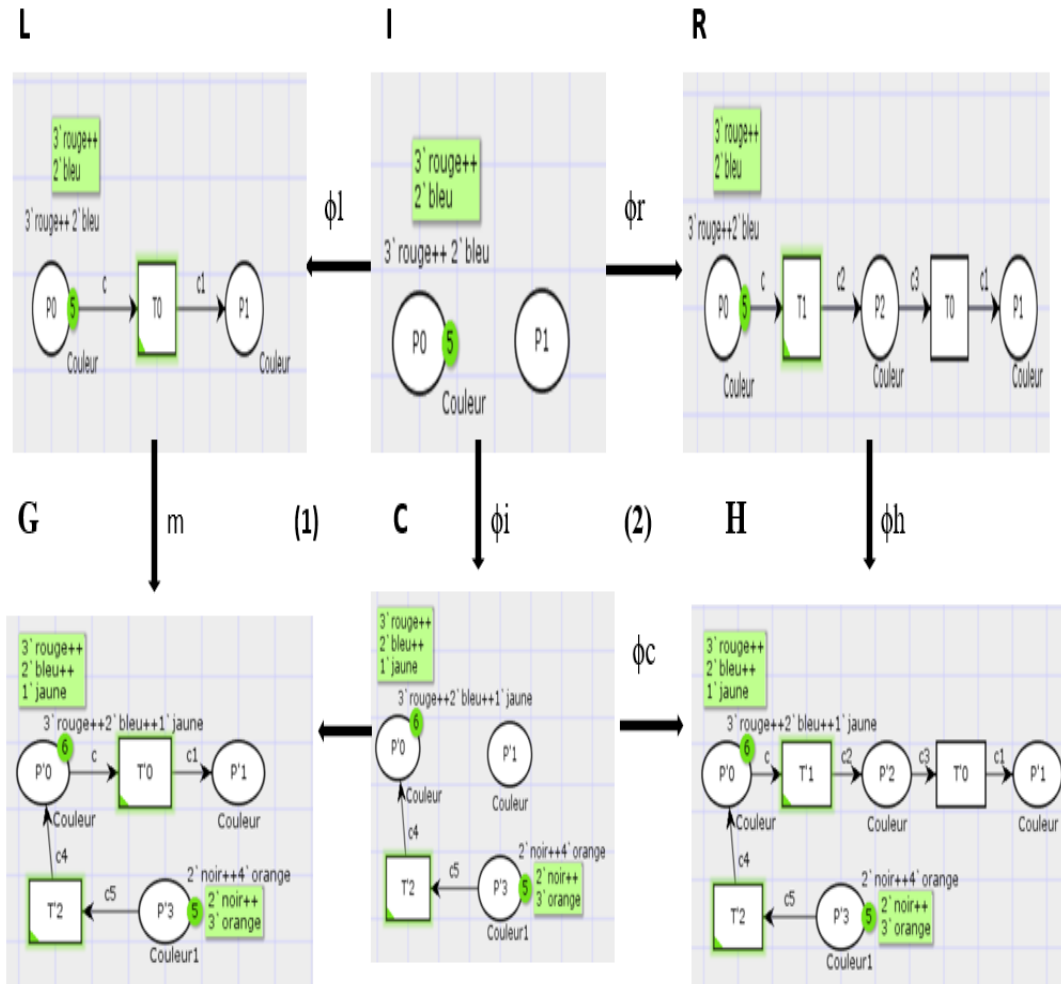


FIGURE 4.3 – Diagramme DPO.

4.3 Exemple illustratif

Dans cette section, nous démontrons l'application du formalisme proposé sur un système de production par une étude de cas système sur les systèmes reconfigurables. Premièrement, nous décrivons la structure et le comportement du système, puis nous appliquons la reconfiguration du réseau RdPC.

Considérons que le système de production qui nous intéresse produit des articles nommés H à partir des matières premières A, B et C . Il existe également un certain nombre de produits intermédiaires : D, ϵ, F, G . Il y a trois machines, $M1$ qui trans-

forme A en D , $M2$ qui transforme B en ϵ et $M3$ qui transforme C en F . Il y a un sous-ensemble qui compose D et ϵ en G et un assemblage final qui compose G et F en H . La figure 4.4 montre la nomenclature.

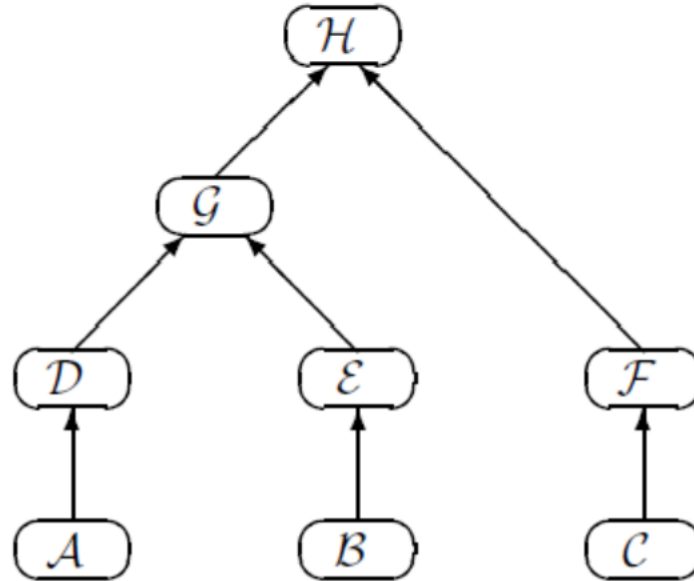


FIGURE 4.4 – La nomenclature.

Le RdPC illustré à la figure 4.5 est utilisé pour modéliser le système de production. Les places $P1, P2, \dots$ et $P11$ sont utilisés pour représenter le flux de produits. Les matières premières A, B et C entrent dans le système par les places $P1, P2$ et $P3$ respectivement. Le produit D est stocké en $P6$, ϵ en $P7$, F en $P8$, G en $P9$ et H en $P10$. Les produits finis H quittent le système par la place $P11$. La demande pour le produit H arrive par le place $DEMAND$.

Notez que nous utilisons l'état initial pour représenter le comportement de l'environnement (par exemple, l'offre et la demande). La machine $M3$ transforme les produits C en F et est modélisée par un système de file d'attente représenté par le

sous-réseau contenant les transitions T1 et T2. Initialement, il y a un jeton en place (FREE3) indiquant que la machine est prête à fonctionner.

Les machines M1 et M2 ont besoin d'un réglage à chaque fois qu'un article est traité. Ce réglage est effectué par une personne travaillant sur les deux machines. Nous pouvons considérer cette personne comme une ressource partagée. La configuration de M1 est représentée par la transition T4, la configuration de M2 est représentée par la transition T3. La personne est représentée par un jeton à la place H1 ou à la place H2. Notez que la personne alterne entre M1 et M2. Les autres parties de M1 et M2 sont modélisées de la même manière que M3. Notez que nous utilisons une commande push pour diriger les machines M1, M2 et M3. Chaque fois que la matière première est disponible et que la machine est libre, une opération est lancée. Nous utilisons un contrôle « pull » pour diriger les deux processus d'assemblage (c'est-à-dire l'assemblage à la commande). Dans cet exemple, une technique de contrôle de type KANBAN est utilisée pour réduire les stocks en cours de fabrication. Cette technique a été développée au Japon pour réaliser une production juste à temps. L'assemblage est autorisé si les composants nécessaires à l'assemblage sont disponibles et si une certaine carte, appelée Kanban, a été reçue. Un nouveau Kanban est fourni dès qu'un produit assemblé est retiré.

On obtient ainsi un processus d'assemblage axé sur la demande.

Le sous-assemblage et l'assemblage final sont représentés par T9 et T10. La livraison de l'article *H* est représentée par la transition T11. La transition T11 est déclenchée s'il existe une demande et un produit final. Si T11 est déclenchée, un nouveau Kanban est fourni au processus d'assemblage final (T10). Si T10 est déclenché, un nouveau Kanban est fourni au processus de sous-assemblage (T9). Notez que la

quantité maximale de produits stockés G et H dépend du nombre de jetons initialement disponibles dans le KANBAN 1 et le KANBAN 2.

Supposons que le système de production reçoive un flux constant de matières premières (A , B , C). Au départ, il y a un Kanban dans le KANBAN 1 et un Kanban dans le KANBAN 2. Nous nous intéressons maintenant à l'arrivée des jetons à la place $P11$.

Le système contient deux configurations de C_0 et C_1 : Dans les deux configurations C_0 et C_1 , la machine $M1$ transforme le produit A en produit D , $M2$ transforme le produit B en produit ϵ . Alors que $M3$ est en train de produire, elle peut échouer. Le système passe en configuration C_1 , de sorte que la production est arrêtée, $M3$ arrête de transformer et $M1$ et $M2$ commencent à travailler. Dans un premier temps, le système est dans la configuration C_0 qui est représentée sur la figure 4.5.

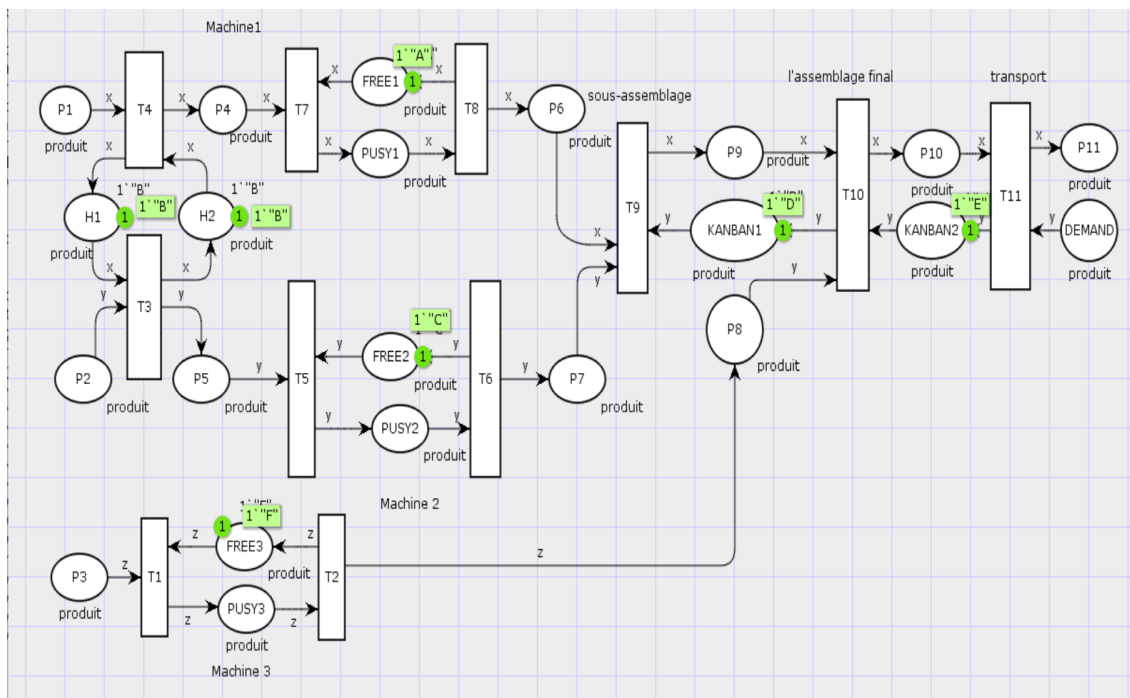


FIGURE 4.5 – Le modèle RdPC pour le système à sa première configuration.

Lorsque la machine $M3$ tombe en panne, le système passe à la configuration C_1

illustrée à la figure 4.6. C_1 est obtenue en appliquant la règle $r_0 = \langle D_0, \bullet r_0, r_0 \bullet, C_0, M_0 \rangle$ (Voir la section 9) sur C_0 où :

1. $D_0 = \{ P1, P4, H1, H2, P2, P5, P3, FREE3, PUSY3 \}$.
2. $\bullet r_0 : P1(T4)+P4(T4,T7)+H1(T4,T3)+H2(T3,T4)+P2(T3) +P5(T3,T5)+P3(T1) +FREE3(T2,T1)+PUSY3(T1,T2)$.
3. $r_0 \bullet : P1(T4)+P4(T4,T7)+H1(T4,T3)+H2(T3,T4)+P2(T3)+P5(T3,T5)+P3(\emptyset) +FREE3(\emptyset)+PUSY3(\emptyset)$.
4. $C_0 = \{P1\}$.
5. $M_0(P1)=1$ "F".

Une fois la machine M3 est réparée, la production de P1 peut être redémarrée.

C_1 est reconfigurée en C_0 en appliquant $r_1 = \langle D_1, \bullet r_1, r_1 \bullet, C_1, M_1 \rangle$ où :

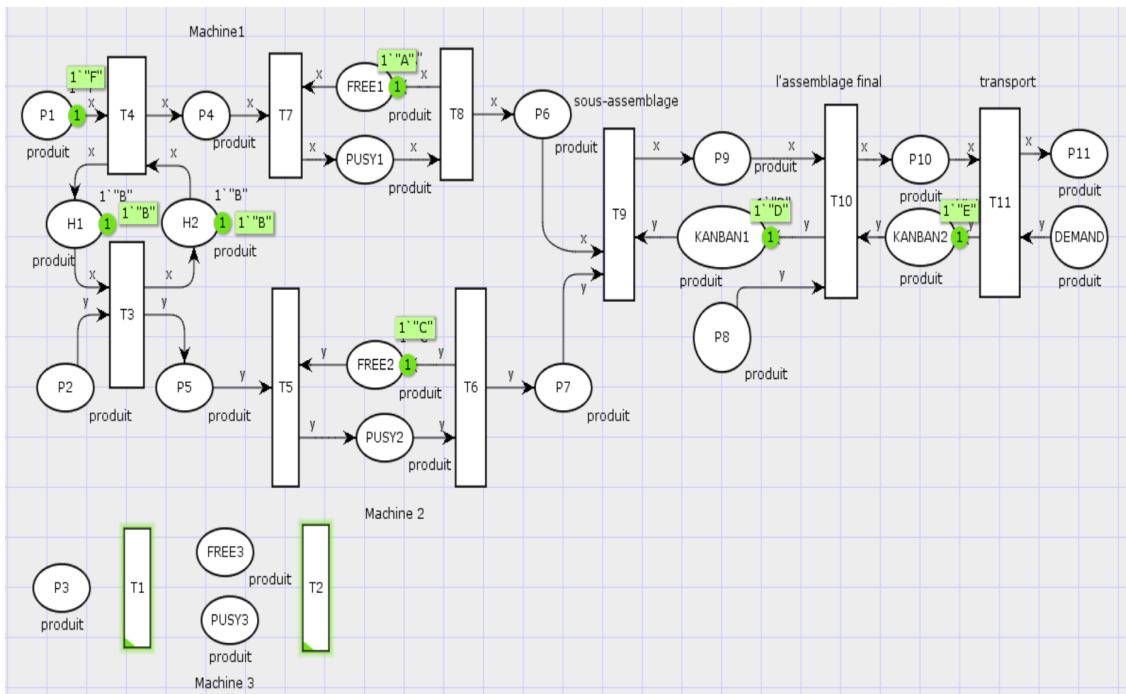


FIGURE 4.6 – Le modèle RdPC pour la configuration du système C_1 lorsque la machine M3 tombe en panne.

1. $D_1 = \{ P1, P4, H1, H2, P2, P5, P3, FREE3, PUSY3 \}$.
2. $\bullet r_1 : P1(T4)+P4(T4,T7)+H1(T4,T3)+H2(T3,T4)+P2(T3)+P5(T3,T5)+P3(\emptyset)$
 $+FREE3(\emptyset)+PUSY3(\emptyset)$.
3. $r_1 \bullet : P1(T4)+P4(T4,T7)+H1(T4,T3)+H2(T3,T4)+P2(T3) +P5(T3,T5)+P3(T1)$
 $+FREE3(T2,T1)+PUSY3(T1,T2)$.
4. $C_1 = \{FREE3\}$.
5. $M_1(FREE3)=1$ "F".

4.4 Conclusion

Dans ce chapitre, nous avons présenté Morphisme réseau de Petri coloré et règle de transformation dans (DPO) à l'aide de l'outil CPN-tools[CPN18]. Nous prévoyons d'améliorer et de compléter l'outil de développement et d'envisager l'analyse des RecRdPCs (Réseaux de Petri Colorés Reconfigurables).

Chapitre 5

Conclusion et Perspectives

Conclusion et Perspectives

Ces dernières années, l'intérêt pour les systèmes d'événements discrets et les systèmes reconfigurables a augmenté. Pour modéliser ce type de systèmes, des réseaux de Petri de faible niveau ont été développés et de nombreuses nouvelles extensions ont été développées, à partir desquelles on peut trouver des RdPCs (Haut niveau), qui permettent de concevoir ces systèmes et d'analyser leurs propriétés. En regardant ce qui précède, nous avons résolu le problème posé, à savoir la définition d'un morphisme de réseaux de Petri colorés et la règle de transformation des réseaux de Petri colorés.

Au cours de la réalisation du projet, nous avons appris à connaître :

1. Réseaux de Petri de Colorés (haut niveau) .
2. Utiliser ces connaissances pour définir de morphisme des réseaux de Petri colorés.
3. Proposer les règles de transformation des réseaux de Petri colorés.

Dans le cadre de nos travaux futurs, nous avons l'intention de nous concentrer sur d'autres questions qui restent à résoudre, en particulier :

- Développement d'un outil pour l'analyse et la modélisation des réseaux de Petri colorés reconfigurables.

Bibliographie

- [Atl12] Maen Atli. *Contributions à la synthèse de commande des systèmes à évènements discrets : nouvelle modélisation des états interdits et application à un atelier flexible*. PhD thesis, Université de Lorraine, 2012.
- [Bek09] Rebiha Berghoute Bekrar. *Identification des systèmes à événements discrets par réseaux de Petri*. PhD thesis, Université du Havre, 2009.
- [BLO03] Eric Badouel, Marisa Llorens, and Javier Oliver. Modeling concurrent systems : Reconfigurable nets. In *PDPTA*, pages 1568–1574. Citeseer, 2003.
- [CPN18] Cpn-tools. <https://cpntools.org/2018/01/16/download/>, January 2018.
- [Dar18] Amira Dardour. *Estimation et diagnostic de réseaux de Petri partiellement observables*. PhD thesis, Angers, 2018.
- [EEGH15] Hartmut Ehrig, Claudia Ermel, Ulrike Golas, and Frank Hermann. Graph and model transformation. *Monographs in Theoretical Computer Science*. Springer, page 17, 2015.
- [GUE11] Faycel GUERROUF. *Une Approche de Transformation des Diagrammes d'Activités d'UML Mobile 2.0 vers les Réseaux de Petri*. PhD thesis, Université de Batna 2, 2011.
- [HEM05] Kathrin Hoffmann, Hartmut Ehrig, and Till Mossakowski. High-level nets with nets and rules as tokens. In *Applications and Theory of Petri*

- Nets 2005 : 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005. Proceedings 26*, pages 268–288. Springer, 2005.
- [LO04a] Marisa Llorens and Javier Oliver. Introducing structural dynamic changes in petri nets : Marked-controlled reconfigurable nets. In *International Symposium on Automated Technology for Verification and Analysis*, pages 310–323. Springer, 2004.
- [LO04b] Marisa Llorens and Javier Oliver. Structural and dynamic changes in concurrent systems : reconfigurable petri nets. *IEEE Transactions on Computers*, 53(9) :1147–1158, 2004.
- [MAR] Stéphane MARIEL. Vade - mecum de l'Étudiant en rÉseaux de petri.
<http://dept-info.labri.fr/~griffaul/Enseignement/FDS/vademecum-petri.pdf>.
- [Mur89] Tadao Murata. Petri nets : Properties, analysis and applications. *Proceedings of the IEEE*, 77(4) :541–580, 1989.
- [Pet77] James L Peterson. Petri nets. *ACM Computing Surveys (CSUR)*, 9(3) :223–252, 1977.
- [PK18] Julia Padberg and Laid Kahloul. Overview of reconfigurable petri nets. *Graph Transformation, Specifications, and Nets : In Memory of Hartmut Ehrig*, pages 201–222, 2018.
- [Rez16] Sadok Rezig. *Approches canoniques pour la synthèse des contrôleurs réseaux de Petri*. PhD thesis, Université de Lorraine, 2016.
- [Roz97] Grzegorz Rozenberg. *Handbook of graph grammars and computing by graph transformation*, volume 1. World scientific, 1997.
- [SOA23] Louis Bal Dit Sollier, Alain Ourghanlian, and Saïd Amari. Description et analyse de performances des réseaux de petri colorés temporisés à

l'aide du dioïde des séries formelles colorées. In *Modélisation des Systèmes Réactifs (MSR'23)*, 2023.

[TIG20] Samir TIGANE. *Reconfiguration in Stochastic Petri Nets*. PhD thesis, Université Mohamed Khider–BISKRA, 2020.

[VdA92] Willibrordus Martinus Pancratius Van der Aalst. *Timed coloured petri nets and their application to logistics*. 1992.