

Spécification des problèmes concurrents :  
Modélisation avec les réseaux de Petri et les  
réseaux de haut-niveau

J.F. Pradat-Peyre

28 janvier 2002

## Table des matières

1.	Les réseaux de Petri . . . . .	2
2.	Les propriétés de base . . . . .	8
3.	Introduction informelle aux réseaux de haut niveau . . . . .	13
4.	Définition des réseaux colorés . . . . .	19

### 1. Les réseaux de Petri

Avec l'avènement du parallélisme surviennent de nombreux problèmes qui lui sont propres. En particulier :

- Comment exprimer ou modéliser la communication et la synchronisation à l'intérieur d'un système ou entre différents systèmes parallèles ?
- Comment valider cette modélisation ?

Un des premiers à tenter de résoudre ces questions fut C.A.Petri qui, dans les années soixante [Pet62], créa un modèle spécifique du parallélisme : les réseaux de Petri. Ce modèle connut un développement immédiat par l'intérêt qu'il suscita tant dans le domaine de la recherche théorique que dans le monde industriel.

En effet la sémantique précise associée à ce modèle en fait un excellent outil de simulation. On peut soit parcourir une ou plusieurs suites d'états du réseau soit, si le graphe d'état est fini, le décrire entièrement et ainsi vérifier directement les propriétés souhaitées par le concepteur du système.

L'originalité et l'intérêt des réseaux de Petri tiennent surtout aux nombreuses techniques de validation autres que le simple examen de ce graphe, souvent coûteux et quelquefois même, irréalisable. Parmi ces méthodes alternatives, il y a d'une part le calcul algébrique qui permet de générer des invariants (flots, semi-flots ..) et d'autre part la réduction de réseaux qui ramène l'étude des propriétés à un réseau de taille plus petite.

Rapidement, les premiers réseaux de Petri se sont révélés être néanmoins un modèle trop limité pour les concepteurs d'applications informatiques ou plus

généralement industrielles. Cette limitation est due principalement à trois facteurs : il est impossible de modéliser des comportements similaires au moyen d'une seule représentation condensée ; l'analyse du réseau n'est pas paramétrable, c'est à dire qu'il faut redessiner le réseau et recommencer son analyse lorsque l'on change le nombre d'objets ( processus, ressources) du système ; les conditions et les conséquences de l'évolution d'un réseau, de nature purement quantitative ( nombre de marques nécessaires dans les places, ajout ou retrait de marques ...), sont inadéquates pour modéliser des conditions et des évolutions qualitatives ( nomination des objets, identité des états de deux processus, numérotation circulaire, relations associatives entre objets ...)

Ainsi des abréviations de réseaux de Petri ont-elles été introduites afin de remédier à ces inconvénients. Les premières abréviations à voir le jour furent les réseaux à Prédicat/Transitions [Gen79], [Gen81], et les réseaux colorés [Jen81]. Puis devant la difficulté d'analyse de ces réseaux, due entre autres à leur absence de structuration, vinrent les réseaux réguliers[Hadd87] qui ont permis de définir de façon précise la paramétrisation d'un réseau, et d'obtenir ainsi les premiers résultats théoriques non partiels sur les réseaux colorés. La paramétrisation est indispensable si l'on veut, par exemple, valider la modélisation d'un anneau à jeton , où le nombre de sites est laissé variable et représente un paramètre du réseau.

Nous rappelons maintenant l'historique des réseaux de Petri, nous en donnons une définition, nous en précisons l'intérêt et enfin, nous décrivons les différentes extensions que sont les réseaux colorés, les réseaux réguliers ou les réseaux à prédicat-transitions.

### 1.1. *Définition des réseaux de Petri*

Les réseaux de Petri sont utilisés afin de modéliser le comportement dynamique de systèmes discrets [BRA83, Mur90]. Ils sont composés de deux types d'objets : les places et les transitions. L'ensemble des places permet de représenter l'état du système ; l'ensemble des transitions représente alors l'ensemble des événements dont l'occurrence provoque la modification de l'état du système. Plus précisément, les places jouent le rôle de variables d'état du système et sont à valeur entière. A l'occurrence d'un événement correspond le franchissement d'une transition dépendant de la satisfaction de préconditions.

Les réseaux de Petri est offrent une représentation graphique simple des systèmes modélisés. Une place est représentée par un cercle, une transition par un rectangle, et les relations de causalité au sein du système sont représentées par la présence - ou l'absence - d'arc valués reliant les places aux transitions.

Un des atouts, et non le moindre, de ce formalisme est d'offrir de nombreux

outils d'analyse structurelle, tels que la théorie des réductions ou le calcul d'invariants. Nous étudierons ces outils dans les prochains chapitres.

Cette présentation intuitive des réseaux de Petri est formalisée par les définitions suivantes.

**Définition 1.1** *Un réseau de Petri est un 4-uplet  $R = \langle P, T, W^-, W^+ \rangle$  où :*

- $P$  est un ensemble fini et non vide de places,
- $T$  est un ensemble fini de transitions,
- $W^-$  ( resp.  $W^+$  ) est la fonction d'incidence avant ( resp. arrière ) de domaine  $P \times T$  et de codomaine  $\mathbb{N}$ .

Dans cette définition,  $W^-(p, t)$  est la précondition associée à la transition  $t$  et la place  $p$ ; elle définit le nombre minimal de marques dans  $p$  nécessaire au franchissement  $t$  et retirées de  $p$  en cas de franchissement. De même,  $W^+(p, t)$  est la postcondition associée à la transition  $t$  et la place  $p$ ; elle définit le nombre de marques apportées à  $p$  par le franchissement de  $t$ .

Ces deux matrices peuvent être synthétisée en une seule : la matrice d'incidence.

**Définition 1.2** *La matrice d'incidence d'un réseau de Petri est la matrice entière  $W \in \mathcal{M}(P, T)$  définie par :  $\forall (p, t) \in P \times T, W(p, t) = W^+(p, t) - W^-(p, t)$ . Cette matrice traduit le coût global du franchissement d'une transition pour chaque place - la différence entre ce qui est produit et ce qui est consommé -.*

**Remarque** De par sa définition, cette matrice masque les "boucles" du réseau et il n'est donc pas équivalent de donner la matrice d'incidence ou de donner les deux matrices pre et post.

Nous voyons maintenant la première notion associée à la dynamique d'un réseau de Petri : le marquage.

**Définition 1.3** *Un élément de  $\mathbb{N}^P$  est appelé marquage d'un réseau de Petri  $R$  et définit l'état du réseau  $R$  à un instant donné, c'est à dire le nombre de marques  $M(p)$  présentes dans chaque place  $p \in P$  du réseau de Petri  $R$ .*

*Un réseau de Petri marqué est donc un 5-uplet  $R = \langle P, T, W^-, W^+, M_0 \rangle$  où :*

- $\langle P, T, W^-, W^+ \rangle$  est un réseau de Petri,
- $M_0$  est le marquage initial de  $R$  définissant ainsi l'état initial du réseau - c'est à dire, avant tout franchissement -.

La dynamique d'un réseau marqué est définie par la règle de franchissement des transitions.

**Définition 1.4** Une transition  $t$  est dite franchissable pour un marquage  $M \in \mathbb{N}^P$  ssi :  $\forall p \in P, M(p) \geq W^-(p)$ . Le marquage  $M'$  alors obtenu est défini par :

$$\forall p \in P, M'(p) = M(p) - W^-(p, t) + W^+(p, t) = M(p) + W(p, t)$$

On note :  $M[t > M']$ , ce qui signifie que  $t$  est franchissable pour le marquage  $M$  et atteint le marquage  $M'$ .

**Remarque** Cette règle d'évolution implique à la fois l'atomicité d'un franchissement et le fait que les notions de séquentialité, de non-déterminisme ou de concurrence sont à égalité dans ce formalisme, toutes étant vues comme des schémas particuliers de causalité. Ce point est également une grande force de ce modèle.

A partir de cette règle de franchissement, il est possible d'obtenir l'ensemble des suites finies ou infinies d'évolution du système à partir du marquage initial. Cette suite d'ensembles définit l'ensemble des marquages accessibles et peut être représentée sous forme de graphe, le graphe des marquages accessibles.

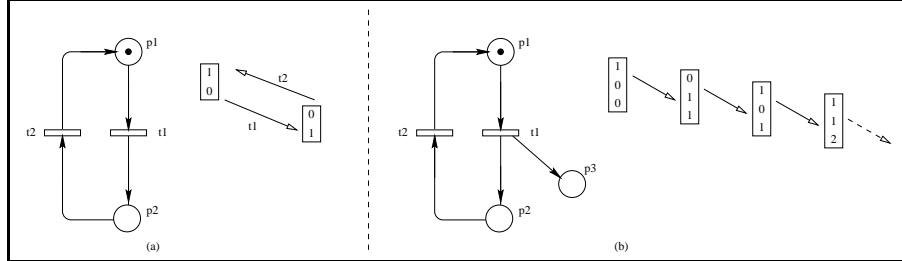
**Définition 0.1** Soit  $(R, M_0)$  un réseau de Petri marqué. L'ensemble des marquages accessibles de ce réseau, noté  $Acc(R, M_0)$  est défini par :  $Acc(R, M_0) = \{M \in \mathbb{N}^P \mid \exists s \in T^* tq M_0[s > M]\}$ . Le graphe des marquages accessibles est le graphe orienté et valué dont les noeuds sont les éléments de  $Acc(R, M_0)$  et est tel, qu'il existe un arc valué par  $t \in T$  de  $M$  vers  $M'$  ssi  $M[t > M']$ .

Ce graphe décrit complètement la sémantique du système. Cependant, il n'est pas nécessairement fini et de plus, même lorsqu'il est fini, ce graphe est souvent de taille trop importante pour que l'on puisse en tirer quelque propriété que ce soit sur le système étudié.

**Exemple :** L'exemple suivant montre deux graphes de marquages accessibles : l'un fini (a) l'autre non fini (b)

Il est cependant possible d'obtenir des informations pertinentes sur le réseau étudié - réseau borné, pas de blocage, exclusion mutuelle et même dans certains cas la vivacité - par des techniques structurels, c'est à dire indépendant du graphes des marquages accessibles.

En effet, soit  $X \in \mathbb{Z}^P$  un vecteur entier sur les places. Si  $W(t)$  désigne la  $t^{ème}$  colonne de la matrice d'incidence  $W$ , alors l'équation de changement



**Figure 1.** Un exemple d'évolution

d'état peut s'écrire :

$${}^tX.M' = {}^tX.M + {}^tX.W(t)$$

Si maintenant  ${}^tX.W(t) = 0$  alors il vient  ${}^tX.M' = {}^tX.M$ , et en particulier,  ${}^tX.M' = {}^tX.M_0$  c'est à dire que la somme des marques contenues dans l'ensemble des places défini par  $Supp(X)$  reste constante moyennant une pondération après le franchissement de la transition  $t$  :

$$\sum_{p \in P} X(p).M(p) = \sum_{p \in P} X(p).M'(p).$$

Les solutions entières  $X$  de l'équation  ${}^tX.W = 0$  seront donc des invariants du réseau. L'intérêt de tels invariants est de permettre une analyse structurale du modèle étudié c'est à dire sans développer tout ou partie du graphe des marquages accessibles dont la taille devient rapidement gigantesque dès que l'on modélise un système un peu complexe.

Dans la pratique, une famille génératrice de telles solutions dans  $\mathbb{Q}$  ou  $\mathbb{Q}^+$  est suffisante pour prouver des propriétés telles que l'exclusion mutuelle, la "bornitude" du réseau, ou encore la présence de places implicites, propriété nécessaire à l'exécution de certaines réductions.

Lorsque ces solutions se trouvent dans l'espace  $\mathbb{Q}^P$ , on parle de "flots" et lorsque celles-ci se trouvent dans l'espace  $(\mathbb{Q}^+)^P$ , on parle alors de "semi-flots" ou encore de "flots positifs". Les premiers se calculeront par l'algorithme de Gauss, tandis que les seconds se calculeront par l'algorithme de Farkas que nous allons présenter dans un chapitre suivant - cf Algo ??, page ?? -.

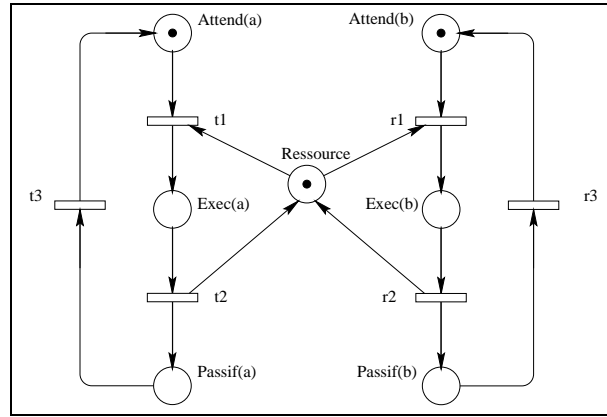
L'intérêt des semi-flots sur les flots est qu'ils permettent d'obtenir directement les propriétés recherchées tandis qu'un calcul des flots nécessitera ensuite d'effectuer de nombreuses combinaisons linéaires de flots avant de trouver l'invariant recherché.

Les semi-flots permettent de plus l'obtention de nombreux autres invariants du réseau, tels que les trappes ou les verrous.

**Définition 1.5** Soit  $W$  la matrice d'incidence d'un réseau de Petri. Alors toute solution non nulle de l'équation  ${}^tX.W = 0$  définit un invariant linéaire du réseau. Si cette solution est à composante positive, on parle de semi-flot, dans le cas contraire on parle de flot.

Une famille génératrice de semi-flots peut donc être obtenue en appliquant l'algorithme de Farkas sur la transposée de la matrice d'incidence.

**Exemple :** Supposons que nous voulions modéliser l'exclusion mutuelle de deux processus sur une ressource commune. Nous représentons le réseau de Petri correspondant .



**Figure 2.** Une solution d'exclusion mutuelle

Si l'on veut alors prouver qu'il y a bien une exclusion mutuelle entre le processus  $a$  et le processus  $b$ , la première solution est de construire le graphe des marquages accessibles. On obtient alors un graphe à 8 états- que nous ne donnons pas car déjà un peu difficile à dessiner - et l'on vérifie qu'il n'y a aucun marquage accessible définissant la présence simultanée des deux processus dans l'état exécuté.

La seconde méthode consiste à calculer une famille génératrice des semi-flots associés à ce réseau, c'est à dire à résoudre l'équation  ${}^tX.W = 0, X \geq 0$ , où  $W$  est la matrice d'incidence du réseau, par l'algorithme de Farkas.

On obtient alors les trois semi-flots :

1.  $Attend(a) + Exec(a) + Passif(a) = cst(M_0) = 1$
2.  $Attend(b) + Exec(b) + Passif(b) = cst(M_0) = 1$
3.  $Exec(a) + Exec(b) + Ressource = cst(M_0) = 1$

Les deux premiers semi-flots traduisent que les processus  $a$  et  $b$  sont bien des processus, i.e il y a séquentialité entre les transitions  $(t1, t2, t3)$  et  $(r1, r2, r3)$ .

Le dernier semi-flot nous permet de prouver l'exclusion mutuelle. En effet, celui-ci traduit le fait que quelque soit l'évolution du système, la somme des marques contenues dans les places Exec(a), Exec(b) et Ressource reste égale à la valeur définie par le marquage initial c'est à dire à 1.

Ainsi donc, il ne peut à la fois avoir une marque dans la place Exec(a) et dans la place Exec(b) ce qui correspond bien à ce que nous voulions prouver.

Le réseau présenté comporte sept places et six transitions. Si l'on voulait représenter l'exclusion mutuelle de dix processus sur une même ressource, il faudrait un réseau de 31 places et 30 transitions. De plus, s'il y a plusieurs ressources disponibles au nombre de  $n$ , un processus pourrait être dans l'état Attend et Execute pour autant de sous ensemble de l'ensemble des ressources disponibles, soit  $2^n$ . Ceci conduirait à dupliquer les places Attend et Execute  $2^n$  fois pour chaque processus. Pour seulement dix processus et cinq ressources, un réseau modélisant le système complet comprendrait 1615 places et transitions.

Il va sans dire que la taille du graphe des marquages accessibles serait très difficilement manipulable et que même un calcul des semi-flots deviendrait quelque peu problématique. En effet, en dehors de la complexité de la résolution, les résultats obtenus seraient difficilement interprétables.

Ce simple exemple met en évidence deux des principales limitations des réseaux de Petri : la taille des réseaux peut rapidement devenir très importante et, surtout, la modélisation nécessite de fixer les paramètres du modèle, ici le nombre de processus et de ressources. Si l'on change la valeur de ces paramètres, il faut refaire l'analyse.

## 2. Les propriétés de base

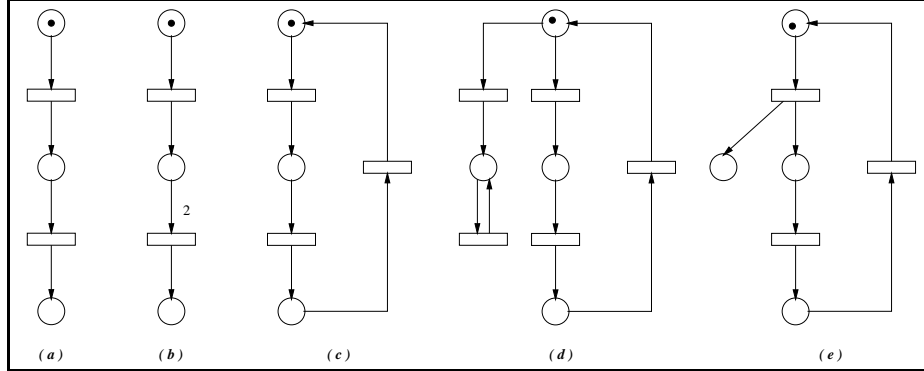
Dans le cadre de la validation il est fondamentale de définir les propriétés que nous voulons observer ou prouver. Dans la théorie des réseaux de Petri, certaines propriétés ont été plus souvent étudiées que d'autres ; en particulier la vivacité. Nous définissons maintenant ces propriétés élémentaires.

**Définition 0.2** Une réseau de Petri marqué  $(R, M_0)$  est dit

- **vivant** ssi  $\forall M \in \text{Acc}(R, M_0), \forall t \in T, \exists s \in T^*$  telle que  $M[s > M'[t >$
- **quasi-vivant** ssi  $\forall t \in T, \exists s \in T^*$  telle que  $M_0[s > M'[t >$
- **pseudo-vivant** ssi  $\forall M \in \text{Acc}(R, M_0), \exists t \in T$  telle que  $M[t >$
- **borné** ssi  $\forall p \in P, \exists k_p \in \mathbb{N}$  tel que  $\forall M \in \text{Acc}(R, M_0), M(p) \leq k_p$

Nous donnons quelques exemples de réseaux vérifiant (ou pas) ces propriétés.





**Figure 3.** Quelques exemples de réseaux de Petri

Tous les réseaux de la figure ??, excepté le réseaux (b), sont quasi-vivant. Les réseaux (c), (d) et (e) sont pseudo-vivant, les réseaux (c) et (e) sont vivant et tous, excepté le réseau (e), sont bornés.

Une autre propriété joue un rôle important dans la théorie des réseaux de Petri : l'existence d'un marquage ou état d'accueil.

**Définition 0.3** Un réseau de Petri marqué  $(R, M_0)$  admet un marquage d'accueil ssi il existe un marquage  $M_a$  vérifiant :

$$\forall M \in \text{Acc}(R, M_0), \exists s \in T^* \text{ telle que } M[s > M_a$$

Ce marquage  $M_a$  est appelé marquage d'accueil.

Un marquage d'accueil est donc un marquage que l'on peut atteindre de tout marquage accessible.

Il existe certaines relations entre ces propriétés.

**Proposition 0.4** Soit  $N = (R, M_0)$  un réseau de Petri marqué.

- Si  $N$  est vivant alors  $N$  est quasi-vivant et pseudo-vivant
- Si  $N$  est quasi-vivant et admet  $M_0$  comme marquage d'accueil alors  $N$  est vivant

De même, certaines propriétés sont monotones, c'est à dire qu'elles restent vraie lorsque l'on augmente le marquage initial.

**Définition 0.5** Une propriété  $\Pi$  est monotone ssi

$$\forall (R, M_0), \Pi((R, M_0)) \implies \forall M'_0 \geq M_0, \Pi((R, M'_0))$$

**Proposition 0.6** *Les propriétés quasi-vivant et non borné sont monotones. Les propriétés pseudo-vivant, vivant, borné et existence d'un marquage d'accueil sont non monotones.*

Cette proposition se démontre sans difficulté en utilisant le lemme de monotonie :

**Lemme 0.7 (de monotonie)** *Soit  $(R, M_0)$  un réseau de Petri marqué.  $\forall M_1 \in \text{Acc}(R, M_0)$ ,  $\forall s \in T^*$ ,  $M_1[s > M_2 \implies \forall M'_1 \geq M_1$ ,  $M'_1[s > M'_2$  et  $M'_2 \geq M_2$*

Toutes ces propriétés peuvent s'exprimer dans un langage de logique temporelle et peuvent donc être vérifiées par des techniques de construction et de parcours du graphe des marquages accessibles (lorsque celui ci est fini, c'est à dire lorsque le réseau est borné). Nous verrons dans les chapitres suivant comment des techniques structurales ou algébriques permettent de vérifier ces propriétés sans la connaissance de ce graphe.

Nous donnons ici une méthode simple, mais semi-automatique, de vérification de la vivacité d'un réseau donné qui utilise la notion de norme.

**Définition 0.8** *Soit  $(R, M_0)$  un réseau de Petri marqué. Une fonction  $\mu$  de l'ensemble des marquages de  $(R, M_0)$  vers  $\mathbb{N}$  est une **norme** pour un marquage  $M_a$  ssi :*

1.  $\mu(M) = 0 \iff M = M_a$
2.  $\forall M, \mu(M) > 0 \implies \exists s \in T^*$  telle que
  - $M[s > M'$
  - $\mu(M') < \mu(M)$

**Définition 0.9** *Soit  $(R, M_0)$  un réseau de Petri marqué. Une fonction  $\mu$  de l'ensemble des marquages de  $(R, M_0)$  vers  $\mathbb{N}$  est une **norme stricte** pour un marquage  $M_a$  ssi :*

1.  $\mu(M) = 0 \iff M = M_a$
2.  $\forall M, \mu(M) > 0 \implies \exists t \in T$  telle que
  - $M[t > M'$
  - $\mu(M') < \mu(M)$

**Proposition 0.10** *Soit  $(R, M_0)$  un réseau de Petri marqué et  $M_a$  un marquage de ce réseau. Les trois propositions suivantes sont équivalentes :*

1. il existe une norme stricte pour  $M_a$
2. il existe une norme pour  $M_a$

### 3. $M_a$ est un marquage d'accueil

Vérifier la vivacité d'un réseau peut donc se ramener à vérifier à l'aide d'une norme que le marquage initial est un marquage d'accueil et que le réseau est quasi-vivant. La difficulté consiste à trouver la norme.

#### Construction d'une norme

1. Trouver les places intervenant dans la norme : en général un sous-ensemble des places non marquées du marquage initial.
2. La norme étant souvent une combinaison linéaire des places choisie, trouver la pondération des places. On choisit généralement les pondérations égales au facteur d'éloignement du marquage d'accueil (plus le fait qu'une place soit marquée éloigne du marquage d'accueil, plus on associe à cette place une pondération forte).

#### Vérification d'une norme

1. Calculer les invariants linéaires du réseau et vérifier la première condition de la norme à l'aide de ces invariants
2. Étudier chaque cas en commençant par ceux correspondants aux plus grands facteurs d'éloignement afin de vérifier la seconde condition de la norme. La transition à trouver est celle qui réduit ce facteur. Pour établir que cette transition est franchissable on se sert à la fois de la définition du cas, des cas déjà étudiés et des invariants.

#### Exemple :

Le réseau suivant modélise une chaîne automatisée de peinture de portes d'automobile. Les portes et la peinture sont supposés être en quantité infinie et sont donc non modélisés. Pour peindre une porte il faut :

1. acquérir un robot pour déplacer la porte vers la zone de peinture (transition  $t_1$ )
2. acquérir une machine à peindre (transition  $t_2$ ) : on libère alors le robot qui n'a plus d'utilité.

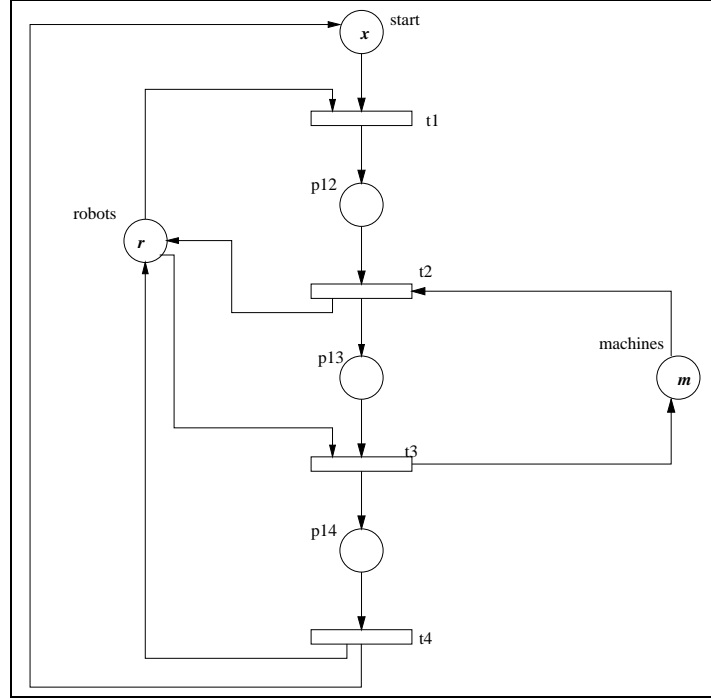
Une fois la porte peinte il faut :

1. acquérir de nouveau un robot pour déplacer la porte peinte (transition  $t_3$ ) : on libère alors la machine à peindre
2. délivrer la porte peinte (transition  $t_4$ ) : on libère alors le robot.

Le marquage initial de ce réseau est défini par :  $M_0(start) = x > 0$ ,  $M_0(robots) = r > 0$ ,  $M_0(machines) = m > 0$  et  $M_0(p) = 0$  pour les autres places  $p$ .

Les flots positifs de ce réseau sont :

1.  $f_1 \equiv start + p_{12} + p_{13} + p_{14} = x$
2.  $f_2 \equiv machines + p_{13} = m$
3.  $f_3 \equiv robots + p_{12} + p_{14} = r$



**Figure 4.** *L'exemple des robots*

On prend la fonction  $\mu(M) = 3.M(p12) + 2.M(p13) + M(p14)$  et l'on montre que cette fonction est une norme stricte pour le marquage initial.

1.  $\mu(M) = 0$  ssi  $M(p12) = M(p13) = M(p14) = 0$  et en utilisant les invariants on obtient immédiatement que  $\mu(M) = 0$  ssi  $M = M_0$ .
2. Il faut différencier les cas possibles lorsque  $\mu(M) > 0$ .

(a)  $M(p14) > 0$

La transition  $t4$  est franchissable et son franchissement fait décroître la norme de 1.

(b)  $M(p14) = 0$  et  $M(p12) > 0$

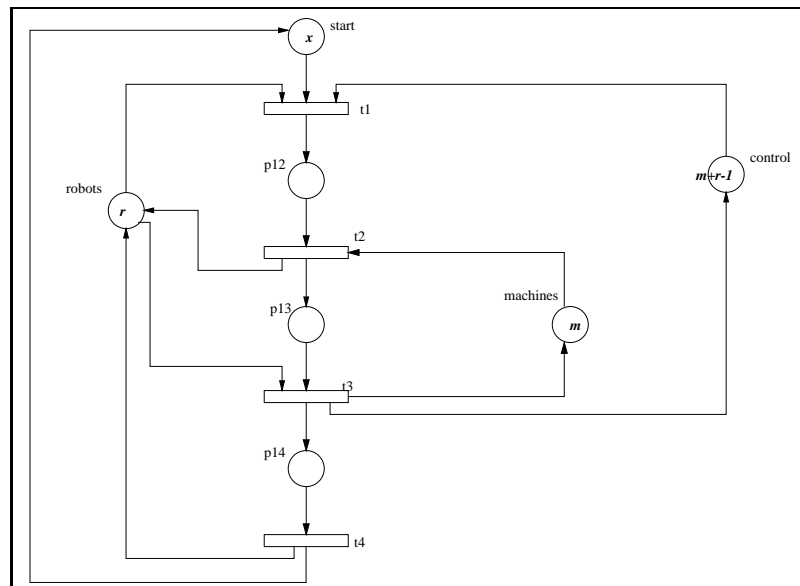
Si  $t2$  est franchissable alors la norme décroît de 1. Sinon, c'est que  $M(machines) = 0$  et  $f_2$  implique alors que  $M(p13) = m$ . Dans ce cas, si  $t3$  est franchissable alors de nouveau la norme décroît ; sinon  $f_3$  implique que  $M(p12) + M(p14) = r$ . Comme  $M(p14) = 0$ ,  $M(p12) = r$ . Pour que ce cas ne soit pas possible  $M(p14) = 0$ ,  $M(p13) = m$ ,  $M(machines) = 0$ ,  $M(robots) = 0$ ,  $M(p12) = r$   $M(start) = x - m - r$  il faut imposer que  $x - m - r < 0$ .

(c)  $M(p14) = 0$ ,  $M(p12) = 0$  et  $M(p13) > 0$

L'invariant  $f_3$  implique  $M(robots) + M(p14) = m$  et donc que soit  $t3$  soit  $t4$  est franchissable ce qui fait bien décroître la norme.

On a donc montrer que sous la condition  $x < m + r$  la fonction  $\mu$  est une norme pour le marquage initial et que donc celui ci est un marquage d'accueil. Comme il est trivial (ici) de prouver que ce réseau est quasi-vivant, on obtient que le réseau est vivant pour tout marquage satisfaisant  $x < m + r$ . On verra que des invariants (les verrous du réseau) permettent de montrer automatiquement que ce réseau est vivant si et seulement si il vérifit la condition  $x < m + r$ .

**Exemple :** Montrer que le réseau suivant est vivant en utilisant la fonction  $\mu(M) = 3.M(p12) + 2.M(p13) + M(14)$ .



**Figure 5.** *L'exemple des robots contrôlés*

### 3. Introduction informelle aux réseaux de haut niveau

Les réseaux de Petri Places-Transitions définissent un cadre théorique simple et puissant pour l'étude de la concurrence. De plus, grâce aux nombreux outils disponibles, ils supportent la spécification et l'analyse de systèmes concurrents dans de vastes domaines d'application.

Cependant, ce formalisme met en avant le contrôle au détriment de la structuration des données : il est facile de modéliser et d'analyser le contrôle d'un

système mais il est difficile d'intégrer dans cette phase de conception l'organisation des données transportées, échangées ou modifiées.

Afin de prendre en compte ce besoin sans modifier la sémantique des réseaux de Petri, différentes abréviations ont été définies. Une abréviation de réseau est obtenue en attachant de nouvelles informations aux noeuds et aux arcs et en définissant une sémantique de fonctionnement qui tient compte de ces paramètres. Le point clef est de garder la possibilité de «déplier» cette abréviation en un réseau de Petri ordinaire dont le comportement est identique. Parmi ces abréviations, on peut citer les réseaux colorés ([Jen91]), les réseaux à Prédicats/Transitions ([GL81]), les réseaux algébriques ([RV87, Rei91]), ou encore les réseaux bien formés ([Dut92]). Ces différents formalismes se caractérisent par une syntaxe plus ou moins naturelle, par les modalités de paramétrisation et par la possibilité d'analyse directe (i.e. opérant sur le formalisme de haut-niveau) et dans certains cas d'analyse paramétrée. De nombreux outils de modélisation et d'analyse basé sur les réseaux de haut-niveau sont aujourd'hui disponibles. On peut citer par exemple CPN-AMI [KPA99], Design/CPN [KCJ98], GreatSPN [CFGR95] ou encore Prod [VJKT95].

Nous étudions principalement dans ce chapitre deux de ces abréviations : les réseaux colorés et les réseaux bien formés (ou WN pour Well-formed Nets) qui ont le même pouvoir d'expression. Les réseaux colorés sont caractérisés par une sémantique fonctionnelle simple : une information de couleur (ou de type) est associée aux jetons et aux franchissements et les valuations des arcs sont des fonctions de couleur qui précisent le nombre et la couleur des jetons consommés ou produits lors du franchissement d'une transition qui se fait pour une instance particulière de couleur. Aucune contrainte syntaxique n'est imposée sur ces fonctions de couleur, ce qui simplifie la phase de modélisation mais rend difficile la définition ou l'extension de techniques d'analyse directe.

Le modèle des WN est une abréviation fonctionnelle et paramétrée des réseaux de Petri qui restreint les fonctions de couleurs à des compositions de fonctions élémentaires (*identité*, *successeur* et *diffusion*) et qui impose que les valeurs de couleurs manipulées soient des tuples de valeurs prises dans des ensembles de base appelés classes. L'intérêt de cette syntaxe plus limitée est l'obtention de techniques d'analyse plus générales que celles des RdP (telle que la construction du graphe des marquages symboliques) ainsi que l'extension paramétrée de techniques classiques (telles que les réductions structurelles ou le calcul d'invariants).

Le plan de ce chapitre est le suivant : après une introduction informelle à l'aide d'exemples aux réseaux de haut-niveau, nous définissons précisément les réseaux colorés (section 3). Nous définissons alors les réseaux bien formés (section 4) et nous terminons par une introduction à d'autres modèles de haut-niveau, les réseaux de Petri interprétés et les réseaux algébriques.

### 3.1. Un modèle client-serveur

Le réseau place/transition R1 suivant (figure 6) modélise deux clients connectés à un serveur.

- Un client dans l'état prêt, dénoté par une marque dans la place  $Cpret$ , envoie (par la transition  $cenv$ ) un message au serveur (place  $Mess$ ) et passe dans l'état attente (place  $Catt$ ). A la réception d'une réponse (place  $Rep$ ), il retourne dans l'état prêt par la transition  $crec$ .
- Le serveur dans l'état prêt ( $Spret$ ) peut recevoir ( $srec$ ) un message et passe dans l'état occupé ( $Socc$ ). Il traite ce message, puis envoie ( $senv$ ) une réponse ( $Rep$ ) et retourne dans l'état repos.

Le marquage initial  $m_0 = \langle 2.Cpret + 1.Spret \rangle$  dénote 2 clients et 1 serveur dans l'état prêt. Par souci de lisibilité, selon le cas, le marquage initial d'une place est représenté à l'intérieur ou à l'extérieur de celle-ci.

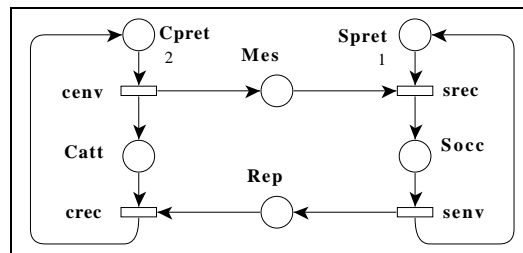


Figure 6. Le réseau R1

Le modèle est cependant trop abstrait car il ne distingue ni l'état de chacun des clients, ni à qui appartient un message ou une réponse. Un flot tel que  $[Cpret + Mess + Socc + Rep]$  exprime qu'un message est préparé, en transit, traité ou que la réponse est transmise. On en déduit que le nombre total de marques dans les places correspondantes est un invariant, égal à 2, mais on ne peut exprimer qu'il y a au plus un tel message séparément pour chaque client. On pourrait introduire un sous-réseau par client et interconnecter tous ces sous-réseaux au serveur, mais le réseau global deviendrait impraticable dès que le nombre de clients augmente. De plus, il en faudrait une nouvelle version chaque fois qu'on change le nombre de clients.

### 3.2. Distinction des clients

Les réseaux colorés ont pour but de distinguer les marques modélisant différents objets ayant les mêmes types d'états et sujets aux mêmes types d'actions.

Le réseau coloré R2 (figure 7 <sup>1</sup>) distingue les deux clients en les identifiant chacun par une couleur. La classe des clients est ici l'ensemble  $\{1..2\}$ . Les places *Cpret* et *Catt* ont cette classe pour domaine, c'est-à-dire qu'elles ne peuvent contenir que des marques de couleur  $\langle 1 \rangle$  ou  $\langle 2 \rangle$ . Le marquage initial  $\langle 1 \rangle$ .*Cpret*+  $\langle 2 \rangle$ .*Cpret* indique que tous les clients sont dans l'état prêt. Il est dénoté par  $\langle \text{client.all} \rangle$ .*Cpret*, ce qui épargne l'énumération explicite de la liste de toutes les couleurs. Un autre avantage de cette notation est qu'elle s'étend facilement à un modèle paramétré (i.e. où l'on ne préciserait pas explicitement le nombre de clients). Les places *Mess*, *Socc*, *Rep* ont le même domaine tandis que la place *Spret* reste non colorée - on dit aussi de *couleur neutre* - car le serveur est prêt à accepter un message de n'importe quel client.

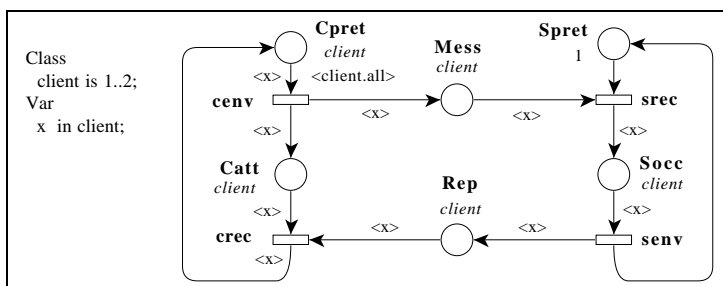


Figure 7. Généralisation à plusieurs clients (réseau R2)

Pour distinguer pour quel client est franchie une transition telle que *cenv*, on explicite le domaine des transitions en définissant un domaine de franchissement (ici *Client* pour la transition *cenv*). On étiquette alors les arcs liés à cette transition par une variable  $\langle x \rangle$  prenant ses valeurs dans la classe *Client*. Bien sûr, la classe de la variable étiquetant un arc doit être identique au domaine de la place située à l'autre extrémité de l'arc. La valeur choisie pour la variable lors d'une occurrence de franchissement précise alors quelle est la couleur de la marque à extraire des places en entrée et des marques déposées dans les places en sortie. Une transition n'est franchissable que s'il y a dans ses places en entrée assez de marques des couleurs désignées. Ainsi les franchissements relatifs à chaque client sont dénotés *cenv*( $\langle 1 \rangle$ ) ou *cenv*( $\langle 2 \rangle$ ) : ils précisent la couleur de la marque à extraire de *Cpret* qui est aussi celle des marques à produire dans *Catt* et *Mess*. Tous les arcs, sauf ceux issus de places sans domaine (ici *Spret*), sont ainsi étiquetés.

Si le client  $\langle 1 \rangle$  envoie un message qui est traité, puis le client  $\langle 2 \rangle$  envoie aussi un message, la séquence *cenv*( $\langle 1 \rangle$ ), *srec*( $\langle 1 \rangle$ ), *senv*( $\langle 1 \rangle$ ), *cenv*( $\langle 2 \rangle$ ) donne le marquage ( $\langle 1 \rangle$  +  $\langle 2 \rangle$ ).*Catt*+  $\langle 2 \rangle$ .*Mess*+  $\langle 1 \rangle$

<sup>1</sup>La représentation graphique des réseaux colorés dépend de l'outil utilisé ; dans ce chapitre la plupart des réseaux sont dessinés avec l'outil Macao de l'atelier CPN-AMI [KPA99].



$.Rep > + < 1 > .Spret$ . La transition  $crec$  n'est alors franchissable que pour le choix  $x = < 1 >$ , ce qui modélise bien un comportement effectif du système. Anticipant sur la notion de flot coloré (voir chapitre 8), nous présentons un flot noté  $[< X > .Cpret + < X > .Catt]$  qui montre que le nombre total de marques de chaque couleur reste égal à 1. Cet invariant exprime que chaque client est un processus séquentiel qui est soit dans l'état prêt soit dans l'état attente.

### 3.3. Distinction des serveurs

Pour généraliser le réseau R2 à plusieurs serveurs, il faut introduire un nouveau domaine de couleurs pour les serveurs. L'étiquetage d'une transition

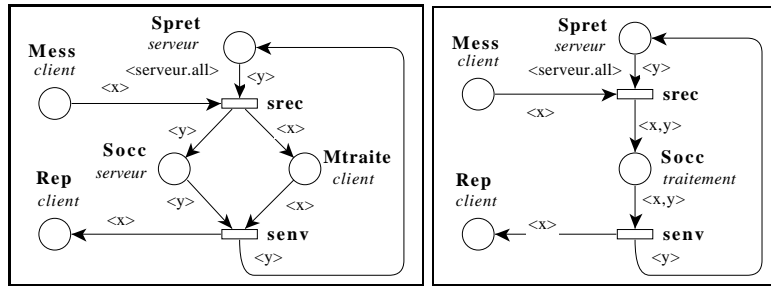


Figure 8. Généralisation à plusieurs serveurs (R3 et R4)

comme  $srec$  doit distinguer par une variable  $x$  la couleur du message traité et par une variable  $y$  la couleur du serveur traitant ce message. Il ne suffirait pas de mémoriser indépendamment dans une place  $Socc$  quel serveur traite un message et dans une place  $Mtraite$  à quel client appartient ce message. Cela engendrerait une confusion : si par exemple le serveur  $< 1 >$  traitait un message pour le client  $< 1 >$  et respectivement le serveur  $< 2 >$  pour le client  $< 2 >$ , la transition  $senv(< 1, 2 >)$  serait franchissable, ce qui ne correspondrait à aucun traitement en cours (modèle de gauche de la figure 8). Pour cette transition, le domaine de couleur est alors  $Client \times Serveur$ , ce qui signifie qu'elle instancie un couple de valeur de  $Client, Serveur$ . Il est alors nécessaire de préciser pour chaque variable apparaissant sur un arc attenant à cette transition à quelle composante du couple instancié elle est associée. Sur l'exemple donné,  $x$  est associé à la première composante et  $y$  à la seconde. Ainsi, le franchissement de  $senv$  pour le couple  $< 1, 2 >$  signifie : on associe à  $x$  la valeur 1 et à  $y$  la valeur 2 et on franchit la transition.

La généralisation de marques et de variables sous forme de n-uplets de couleurs modélise correctement de telles associations des clients et des serveurs

(modèle de droite de la figure 8). La place *Socc* a maintenant un domaine traitement correspondant à un doublet  $\langle client, serveur \rangle$ . Un franchissement tel que  $(srec(\langle 1, 1 \rangle))$  produit un doublet  $\langle 1, 1 \rangle$  dans *Socc* puis  $srec(\langle 2, 2 \rangle)$  y produit également un doublet  $\langle 2, 2 \rangle$ . Le franchissement de *senv* nécessite la présence d'un doublet  $\langle x, y \rangle$  dans *Socc*. Pour le marquage courant, cela ne peut être qu'un des doublets  $\langle 1, 1 \rangle$  ou  $\langle 2, 2 \rangle$  de sorte qu'on ne risque plus un franchissement fautif avec  $x = \langle 1 \rangle$  et  $y = \langle 2 \rangle$ . Un tel franchissement serait possible si un doublet  $\langle 1, 2 \rangle$  était produit dans *Socc* après réception d'un message du client  $\langle 1 \rangle$  par le serveur  $\langle 2 \rangle$ .

### 3.4. Le modèle coloré complet

Pour qu'un client puisse désigner le serveur qui devra traiter son message, les messages doivent comporter cette identification; ils deviennent ainsi des doublets de domaine  $\langle client, serveur \rangle$ . S'il y a des marques  $\langle x, y \rangle$  dans la place *Mess*, la transition  $srec(x, y)$  n'est franchissable que si le serveur  $y$  désigné par une de ces marques est dans l'état prêt.

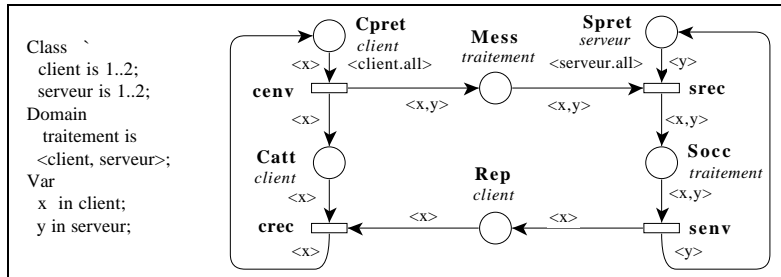


Figure 9. Le réseau R5

### 3.5. Réseau de Petri sous-jacent

Le dépliage d'un réseau coloré par rapport à un domaine de couleur produit un réseau équivalent par duplication des places, transitions et arcs concernés par ce domaine. Cette opération est applicable itérativement jusqu'à obtenir un réseau non coloré. Le réseau R6 suivant illustre le dépliage du réseau R5.

- Chaque client, 1 ou 2, engendre des places que l'on suffixe par sa couleur de marques. Ainsi on obtient *Cpret\_1* ou *Cpret\_2* et *Catt\_1* ou *Catt\_2*.

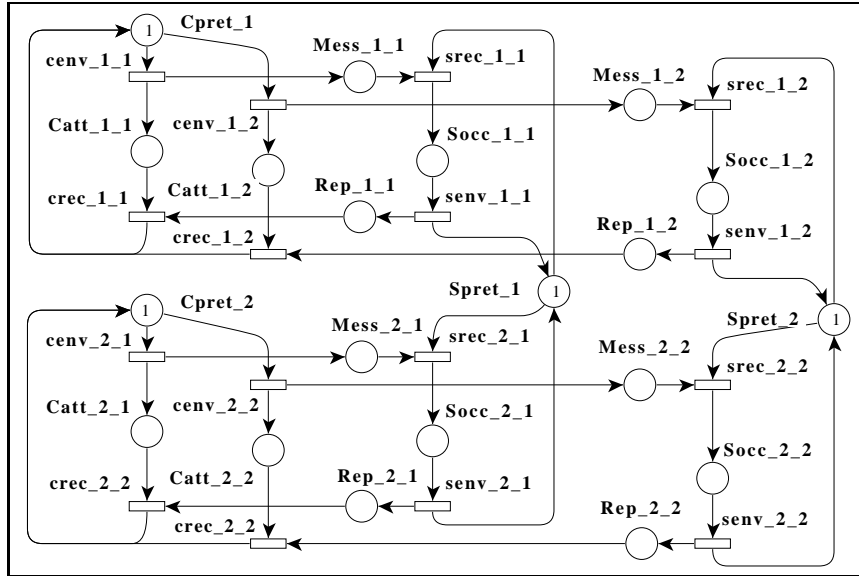


Figure 10. Le réseau  $R6$

- Chaque serveur, 1 ou 2, engendre un sous-réseau que l'on suffixe par sa couleur de marques.
- Ces suffixages se combinent dans le cas d'un domaine obtenu par produit cartésien de domaines élémentaires, de sorte que la place  $Mess$  engendre 4 places  $Mess_{1\_1}, Mess_{1\_2}$  pour un message du client 1 envoyé à chacun des serveurs et de même  $Mess_{2\_1}, Mess_{2\_2}$  pour le client 2.
- Chaque transition engendre autant de transitions suffixées qu'il y a de valeurs de n-uplets de son domaine. Ainsi la transition  $cenv$  engendre  $cenv_{1\_1}, cenv_{1\_2}, cenv_{2\_1}$  et  $cenv_{2\_2}$ .
- Le marquage initial distribue une marque, dans  $Cpret_{1}$  et  $Cpret_{2}$ , et une marque dans  $Spret_{1}$  et  $Spret_{2}$ . Ce réseau possède une base de 6 flots : 2 caractérisent les clients, 2 caractérisent les serveurs et 2 autres concernent des circuits de traitements de messages.

#### 4. Définition des réseaux colorés

Deux familles de réseaux de haut niveau, les réseaux colorés et les réseaux à prédicats/transitions ont été initialement introduits afin de prendre en compte dans un réseau de Petri des comportements similaires mais devant être distingués. Ces deux modèles étant équivalents en terme de modélisation [Jen91],

nous nous attacherons plus spécifiquement aux réseaux colorés dont la définition, plus formelle, a servi de support aux principales études théoriques.

Ces formalismes permettent de modéliser des familles de systèmes à structure identique mais de taille différente. En effet, à chaque place et à chaque transition est associé un domaine de couleur qui permet, pour le cas des places, de colorer les marques pouvant être contenues dans ces places, et pour le cas des transitions, de choisir une couleur de franchissement.

Ainsi pour distinguer plusieurs instances d'un même état, on ne duplique plus les places et les transitions comme on le fait avec les réseaux de Petri ordinaires, on les «colore» .

Les pré-conditions (resp. les post-conditions) deviennent des fonctions de couleur qui à une couleur de transition  $c_1$  et une couleur de place  $c_2$  associe le nombre de marques de couleur  $c_2$  à consommer (resp. à produire) lors du franchissement de la transition pour l'instance  $c_1$ .

#### 4.1. Notations

Avant de définir plus précisément les réseaux colorés, donnons rapidement quelques définitions de base.

**Définition 0.11** *Un multi-ensemble sur un ensemble fini et non vide  $Y$  est une application de  $Y$  vers  $\mathbb{N}$ .*

Intuitivement, un multi-ensemble est un ensemble pouvant contenir plusieurs occurrences d'un même élément. Chaque multi-ensemble  $a$  sur  $Y$  peut être représenté par la somme formelle  $a = \sum_{y \in Y} a(y).y$ , où l'entier positif ou nul  $a(y)$  désigne le nombre d'occurrences de l'élément  $y$  dans le multi-ensemble  $a$ . Par exemple, la somme  $m = 2.\alpha + 3.\beta$  désigne le multi-ensemble contenant 2 fois la valeur  $\alpha$  et 3 fois la valeur  $\beta$  sur un ensemble contenant les éléments  $\alpha$  et  $\beta$ . Nous notons  $Bag(Y)$  l'ensemble des multi-ensembles sur  $Y$ .

La somme de deux éléments de  $Bag(Y)$  et le produit d'un élément de  $Bag(Y)$  par un entier se définissent naturellement.

**Définition 0.12** *Soit  $a$  un élément de  $Bag(Y)$ .*

- *Si  $b$  est un élément de  $Bag(Y)$  alors la somme de  $a + b$  de  $a$  et  $b$  est un élément de  $Bag(Y)$  défini par :  $a + b = \sum_{y \in Y} (a(y) + b(y)).y$*
- *si  $\lambda$  est un entier positif, alors le produit de  $\lambda.a$  de  $\lambda$  par  $a$  est un élément de  $Bag(Y)$  défini par :  $\lambda.a = \sum_{y \in Y} (\lambda.a(y)).y$ .*

$Bag(Y)$  est muni d'une relation d'ordre qui n'est autre que l'extension naturelle de la relation d'ordre sur  $\mathbb{N}$ , ce que formalise la définition suivante.

**Définition 0.13** Soient  $a = \sum_{y \in Y} a(y).y$  et  $b = \sum_{y \in Y} b(y).y$  deux multi-ensembles sur  $Y$ . Alors  $a$  est supérieur ou égal à  $b$ , ce qui est noté  $a \geq b$  si et seulement si  $\forall y \in Y, a(y) \geq b(y)$ .

Les domaines de couleur d'un réseau de haut niveau sont des produits cartésiens d'ensembles finis. On manipule donc des tuples et nous précisons ici ce qu'est un tuple de valeurs et une tuple de fonctions.

**Définition 0.14 (Valeur de tuple)** Soient  $C_1, \dots, C_k$  et  $C$  des ensembles finis. On appelle un élément de  $C_1 \times \dots \times C_k$  un  $k$ -tuple (ou simplement tuple) et on le note  $\langle c_1, \dots, c_k \rangle$ .

**Définition 0.15 (Tuple de fonctions)** Soient  $f_1, \dots, f_k$  un ensemble d'applications de  $C$  dans  $Bag(C_i)$  ( $\forall i \in [1..k], f_i : C \rightarrow Bag(C_i)$ ). Le tuple de fonctions  $\langle f_1, \dots, f_k \rangle$  est une fonction  $f$  qui a pour domaine  $C$  et codomaine  $Bag(C_1) \times \dots \times Bag(C_k)$  avec  $\forall c \in C, \langle f_1, \dots, f_k \rangle(c) = \langle f_1(c), \dots, f_k(c) \rangle$ .

Une fonction d'un ensemble  $C$  vers un ensemble  $Bag(C')$  s'étend en une application linéaire de  $Bag(C)$  vers  $Bag(C')$  par les règles de calcul suivantes :

- $f(\lambda.c) = \lambda.f(c)$
- $f(c_1 + c_2) = f(c_1) + f(c_2)$

De même,  $Bag(C_1 \times \dots \times C_k)$  s'identifie à  $Bag(C_1) \times \dots \times Bag(C_k)$  par les règles de calcul :

- $\langle c_1, \dots, \lambda.c_i, \dots, c_k \rangle = \lambda.\langle c_1, \dots, c_i, \dots, c_k \rangle$
- $\langle c_1, \dots, c_i + c'_i, \dots, c_k \rangle = \langle c_1, \dots, c_i, \dots, c_k \rangle + \langle c_1, \dots, c'_i, \dots, c_k \rangle$

## 4.2. Le formalisme

Les réseaux colorés offrent une possibilité d'expression plus compacte que les réseaux ordinaires. En effet, dans un réseau coloré, une place peut contenir des jetons de différentes couleurs, et une transition peut être franchie de différentes manières suivant la couleur sélectionnée. Ceci est réalisé en attachant un domaine de couleur à chaque place et chaque transition. Ainsi, pour un nombre de places et de transition identique, le nombre de comportements pouvant être exprimés par un réseau coloré est bien plus élevé qu'avec un réseau de Petri ordinaire.

Un arc joignant une place et une transition est étiqueté par une application linéaire appelée fonction de couleur. Cette fonction détermine pour chaque

couleur de transition (ou instance de franchissement) le nombre de jetons par couleur devant être consommés ou produits dans la place lors du franchissement de la transition pour la couleur choisie. Comme pour les réseaux de Petri ordinaires, le choix de la couleur de franchissement d'une transition se fait de façon indéterministe : si une transition  $t$  est franchissable pour une couleur  $c_1$  et pour une couleur  $c_2$ , rien ne précise dans le formalisme le choix de l'instance de franchissement de  $t$ . Ces différents concepts sont formalisés par les définitions suivantes.

**Définition 0.16** Un réseau coloré est un 5-tuple  $CN = \langle P, T, \mathcal{C}, \text{Pré}, \text{Post} \rangle$  où :

- $P$  est un ensemble non vide et fini de places ;
- $T$  est un ensemble non vide et fini disjoint de  $P$  de transitions ;
- $\mathcal{C}$  est la fonction de couleur de domaine  $P \cup T$  et de codomaine  $\omega$  avec  $\omega$  un ensemble contenant les ensembles finis et non vides. Un élément de  $\mathcal{C}(s)$  est appelé couleur de  $s$  et  $\mathcal{C}(s)$  désigne le domaine de couleur de  $s$  ;
- $\text{Post}$  (resp.  $\text{Pré}$ ) est la fonction d'incidence avant (resp. arrière) qui associe à chaque place  $p \in P$  et chaque transition  $t \in T$ , une application de  $\mathcal{C}(t)$  vers  $\text{Bag}(\mathcal{C}(p))$ .

**Définition 0.17** Un marquage  $m$  d'un réseau est un vecteur indexé par  $P$  où pour chaque place  $p \in P$ ,  $m(p)$  est un élément de  $\text{Bag}(\mathcal{C}(p))$  et désigne le nombre de marques colorées dans cette place  $p$ .

Un réseau coloré marqué est donc un couple  $\langle CN, m_0 \rangle$  où  $CN$  est un réseau coloré et  $m_0$  son marquage initial, i.e. le nombre de marques colorées présentes initialement dans chaque place du réseau.

**Définition 0.18** La règle de tir (ou de franchissement) d'un réseau coloré est définie de la manière suivante :

- Une transition  $t$  est franchissable pour un marquage  $m$  et une couleur  $c_t \in \mathcal{C}(t)$  si :

$$\forall p \in P, m(p) \geq \text{Pré}(p, t)(c_t)$$

- Le franchissement d'une transition  $t$  pour un marquage  $m$  et une couleur  $c_t \in \mathcal{C}(t)$  engendre le nouveau marquage  $m'$  défini par :

$$\forall p \in P, m'(p) = m(p) + \text{Post}(p, t)(c_t) - \text{Pré}(p, t)(c_t)$$

On note alors  $m[t(c_t)]m'$ , ou encore  $m \xrightarrow{t(c_t)} m'$ , ce qui signifie que  $t$  est franchissable par rapport à la couleur  $c_t$  pour le marquage  $m$  et atteint le marquage  $m'$ . Lorsque le marquage atteint n'est pas utile, on note  $m[t(c_t)]$  le fait que  $t$  est franchissable par rapport à la couleur  $c_t$  pour le marquage  $m$ . On note également,  $\text{Acc}(CN, m_0)$  l'ensemble de tous les marquages accessibles à partir de  $m_0$ .

Un réseau coloré n'est en fait que l'abréviation d'un réseau de Petri ordinaire sous-jacent, appelé réseau déplié ou développé.

Le principe du dépliage d'un réseau coloré en un réseau de Petri se fait de la manière suivante : une place dans le réseau déplié par place et couleur de place dans le réseau coloré et la même chose pour les transitions. On déduit ensuite la valuation des arcs et le marquage des places par applications des fonctions de couleur. Comme on le voit, l'hypothèse de finitude des domaines de couleur est indispensable pour effectuer ce dépliage.

**Définition 0.19 (Réseau déplié)** Soit  $CN = \langle P, T, \mathcal{C}, \text{Pré}, \text{Post} \rangle$  un réseau coloré. Le réseau de Petri Places-Transitions développé de  $CN$  est le réseau  $CN_d = \langle P_d, T_d, \text{Pré}_d, \text{Post}_d \rangle$  défini par :

- $P_d = \bigcup_{p \in P, c_p \in \mathcal{C}(p)} (p, c_p)$  l'ensemble des places,
- $T_d = \bigcup_{t \in T, c_t \in \mathcal{C}(t)} (t, c_t)$  l'ensemble des transitions,
- $\text{Post}_d$  (resp.  $\text{Pré}_d$ ) la fonction d'incidence avant (resp. arrière) définie de  $P_d \times T_d$  vers  $\mathbb{N}$  par :
  - $\text{Post}_d(p, c_p)(t, c_t) = \text{Post}(p, t)(c_t)(c_p)$  et

$$- \text{Pré}_d(p, c_p)(t, c_t) = \text{Pré}(p, t)(c_t)(c_p)$$

Le marquage développé d'un marquage coloré se définit de façon similaire.

**Définition 0.20** Soient  $CN$  un réseau coloré et  $m$  un marquage de ce réseau. Le marquage développé de  $m$  dans le réseau développé de  $CN$  est le marquage  $m_d$  défini par :

$$\forall (p, c_p) \in P_d, m_d(p, c_p) = m(p)(c_p).$$

On obtient alors la proposition fondamentale suivante qui dit que la sémantique d'un réseau coloré et de son réseau déplié sont identiques moyennant un isomorphisme.

**Proposition 0.21**  $m[t(c_t)]m'$  dans  $CN$  si et seulement si  $m_d[t(c_t)]m'_d$  dans  $CN_d$ .

Ainsi donc, l'analyse d'un réseau coloré pourrait se ramener simplement à l'analyse de son réseau déplié : un système est modélisé par un réseau coloré, le réseau coloré est déplié, puis l'analyse se fait sur le réseau déplié en utilisant les techniques classiques d'analyse des réseaux de Petri ordinaires.

Cette méthodologie se heurte à deux problèmes importants :

1. la taille du réseau déplié est généralement très importante du fait du pouvoir de concision fourni par la formalisme coloré (ce qui peut rendre inopérant bon nombre de techniques d'analyse) ;
2. les résultats d'analyse obtenus sur le réseau déplié sont rarement, ou très difficilement, interprétables dans le réseau d'origine.

L'intérêt des réseaux colorés est donc fortement lié à l'existence d'outils de validation directe, c'est à dire opérant directement sur le modèle coloré. Malheureusement, l'extension de techniques classiques (calcul d'invariants, application de réductions, etc.) ainsi que la définition de techniques particulières (basées par exemple sur l'utilisation de symétries naturelles du modèle) sont rendues particulièrement difficiles du fait de la non structuration des fonctions et domaines de couleur. Le problème posé est alors : peut-on définir un formalisme suffisamment structuré pour permettre la définition d'algorithmes d'analyse directe tout en maintenant un pouvoir d'expression suffisamment élevé ? <sup>2</sup>

Devant la difficulté du problème posé, différentes familles de réseaux de haut niveau ont été définies : les réseaux réguliers [Had87], les réseaux ordonnés [CH88] ou encore les réseaux à prédicats/transitions unaires [GJ86]. Toutes répondent au besoin de structurer les domaines et les fonctions de couleur afin

<sup>2</sup>Une analogie peut être faite avec l'évolution des langages de programmation.



de pouvoir développer des algorithmes d'analyse opérant directement sur le modèle de haut niveau.

Ces différents formalismes ont été ensuite homogénéisés par la définition des réseaux colorés bien formés [CDFH90] (ou réseaux bien formés), que nous allons étudier dans la section suivante. Il est prouvé que ce formalisme a le même pouvoir d'expression que les réseaux colorés. En revanche, la structuration de ce modèle a permis de définir (ou d'étendre) un nombre important de techniques d'analyse opérant directement sur le réseau de haut-niveau.



## Bibliographie

- [BRA83] G.W. BRAMS. *Réseaux de Petri : Theorie et pratique*. Masson, 1983.
- [CDFH90] C. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. On well-formed colored nets and their symbolic reachability graph. In *proc. of the 11th International Conference on Application and Theory of Petri Nets*, Paris-France, June 1990.
- [CFGR95] G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaud. Greatspn 1.7 : Graphical editor and analyzer for timed and stochastic petri nets. In *Performance Evaluation, special issue on Performance Modeling Tools*, 1995.
- [CH88] J.M. Couvreur and S. Haddad. Towards a general and powerful computation of flows for parameterized coloured nets. In *9th European Workshop on Application and Theory of Petri Nets*, volume II, Venice (Italy), June 1988.
- [Dut92] C. Dutheillet. *Symétries dans les réseaux colorés ; Définition, analyse et applications à l'évaluation de performance*. PhD thesis, Université Pierre et Marie Curie, Paris, 1992.
- [GJ86] Memmi G. and Vautherin J. Analysing nets by the invariant method. In *Petri Nets : Central Model and their Properties*, Lecture Notes in Computer Science. Springer Verlag, 1986.
- [GL81] H.J. Genrich and K. Lautenbach. System modeling with high-level Petri-nets. In *Theoretical Computer Science*, number 13, pages 103–136, 1981.
- [Had87] S. Haddad. *Une catégorie régulière de réseau de Petri de haut niveau : définition, propriétés et réductions. Application à la validation de systèmes distribués*. PhD thesis, Université Pierre et Marie Curie, Paris, 1987.
- [Jen91] K. Jensen. Coloured Petri nets : A high level language for system design and analysis. In Jensen and Rozenberg, editors, *High-level Petri Nets, Theory and Application*, pages 44–119. Springer-Verlag, 1991.
- [KCJ98] L.M. Kristensen, S. Christensen, and K. Jensen. The practitioner's guide to coloured Petri nets. In *International Journal on Software Tools for Technology Transfer*. Springer-Verlag, 1998.
- [KPA99] F. Kordon and E. Paviot-Adet. Using CPN-AMI to validate a safe channel protocol. In *High-level Petri Nets, Theory and Application*, LNCS. Springer-Verlag, 1999.
- [Mur90] T. Murata. Petri nets : properties, analysis and applications. In *proceedings of the IEEE Vol 6*, number 1, pages 39–50, January 1990.

- [Rei91] Wolfgang Reisig. Petri nets and algebraic specifications. *Theoretical Computer Science*, 80 :1–34, 1991. NewsletterInfo : 38,39.
- [RV87] W. Reisig and J. Vautherin. An algebraic approach to high level petri nets. In *Proceedings of the Eighth European Workshop on Application and Theory of Petri Nets*, pages 51–72. Universidad de Zaragoza (Spain), 1987.
- [VJKT95] K. Varoaniemi, Halme J., Hiekanen K., and Pyssisalo T. prod reference manual. Technical Report 13, Helsinki Univ. of Tecnologies, Finland, 1995.