

Réseaux de Petri

Documents autorisés : cours photocopié, notes de cours, examens des 2 années antérieures

1) Problème 1 (6 points)

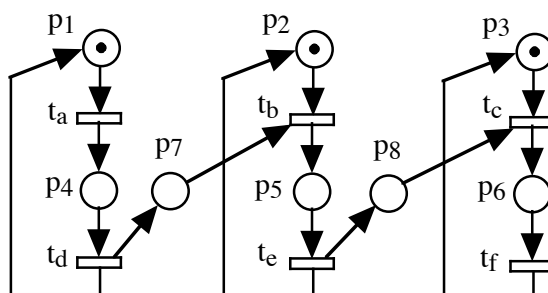


FIG. 1 – Réseau de Petri initial

On considère un système de traitement du signal à trois niveaux (principe du fonctionnement pipe-line). Un flot d'informations, décomposé en paquets, se présente à l'entrée du système. On suppose que chaque paquet provoque un franchissement de la transition t_a . Une première tâche informatique (premier niveau) traite cycliquement les paquets (un paquet par cycle). A la fin du traitement d'un paquet, elle envoie un paquet de données à une seconde tâche (second niveau). Cette seconde tâche traite également cycliquement les paquets et à la fin de chaque cycle, elle envoie un paquet de données à la troisième tâche (troisième niveau). On suppose que la sortie de la troisième tâche est la transition t_f . Chaque franchissement de t_f correspond à l'émission d'un paquet de résultats.

- Montrer que le réseau de Petri de la figure 1 peut être une spécification de ce système.
- Analyser ce réseau de Petri par réduction (bien donner les règles), quelles sont ses propriétés ? Ce système vous paraît-il correct ?
- On suppose que le fonctionnement du second niveau étant plus lent que celui du premier niveau, on décide de doubler la tâche associée à ce niveau. Comment pourrait-on prendre en compte cette modification de façon grossière sans rajouter de place, ni de transition ni d'arc au réseau de Petri de la figure 1. Les propriétés trouvées en b) sont-elles modifiées ?

2) Problème 2 (10 points)

Pour éviter tout problème de débordement de "buffer" (mémoires tampons entre deux niveaux et stockant les paquets), on se résout à spécifier le mécanisme décrit dans le problème précédent par le réseau de Petri de la figure 2.

- Analyser ce réseau de Petri par réduction (bien donner les règles), quelles sont ses propriétés ? Ce système vous paraît-il correct ?

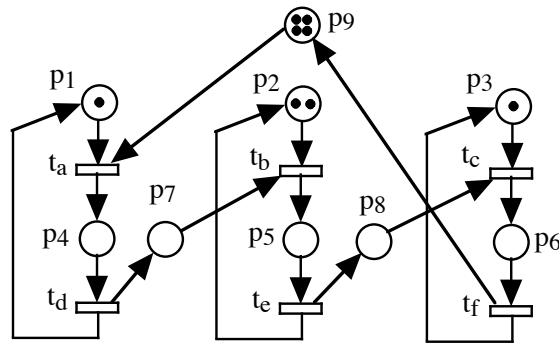


FIG. 2 – Réseau de Petri modifié

e) Calculer une base de composantes conservatives positives pour le réseau de Petri de la figure 2. Montrer que les "buffers" (celui entre le niveau 1 et le niveau 2 et celui entre le niveau 2 et le niveau 3) sont bornés et donner la valeur de la borne.

f) Expliquer le rôle de la place p_9 . Que se passerait-il si l'on ne mettait initialement qu'un seul jeton dans cette place ?

3) Problème 3 (4 points)

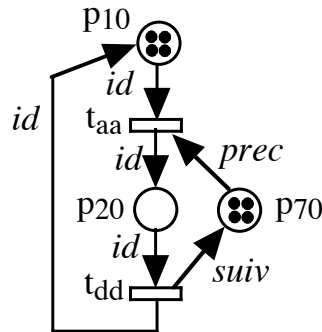


FIG. 3 – Réseau de Petri coloré

On représente le réseau de Petri de la figure 2 par le réseau de Petri coloré de la figure 3. On prend comme couleurs les noms des niveaux (**niv1** et **niv3** pour les niveaux 1 et 3, **niv21** et **niv22** pour les deux tâches de niveau 2) ainsi que les noms des "buffers" (**entrée**, **1vers2** et **2vers3**).

g) Donner les sous ensembles de couleurs des places et des transitions, les expressions des matrices id , $prec$ et $suiv$ et le marquage initial.

Correction

Correction du problème 1

a) On peut clairement identifier trois objets (ou trois entités) qui communiquent entre eux par des communications asynchrones. Ceci est identifié par la figure 4 où nous voyons les trois niveaux du système, chacun formés de deux places et de deux transitions ainsi que les places p_7 et p_8 qui correspondent à la communication entre le niveau 1 et le niveau 2 et à la communication entre le niveau 2 et le niveau 3 respectivement.

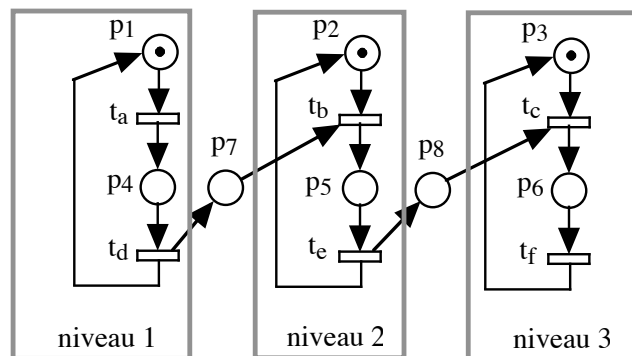


FIG. 4 – Vue de l'architecture du système

Chaque niveau est constitué d'une tâche. La place marquée initialement correspond à l'état inactif (attente de traitement) et l'autre place à l'état actif (traitement en cours). Nous avons donc l'association suivante :

- place p_1 : traitement du niveau 1 inactif (prêt),
- place p_4 : traitement du niveau 1 en cours,
- place p_2 : traitement du niveau 2 inactif (prêt),
- place p_5 : traitement du niveau 2 en cours,
- place p_3 : traitement du niveau 3 inactif (prêt),
- place p_6 : traitement du niveau 3 en cours,
- place p_7 : registre tampon (buffer) de communication entre le niveau 1 et le niveau 2,
- place p_8 : registre tampon (buffer) de communication entre le niveau 2 et le niveau 3.

Lors de l'interprétation de ce réseau de Petri, pour représenter l'interaction du système avec son environnement, la réception d'un message est associée à la transition t_a . Ce message est une demande de traitement associée à un bloc de données. A la transition t_f est associée l'émission d'un message résultat avec un bloc de données décrivant le résultat obtenu. Pour que la transition t_a soit franchie, il faut donc qu'il y ait un jeton dans la place p_1 (tâche prête) et qu'un message soit présent à l'entrée du système. Le bloc de données associé au message est vu comme un tout autonome et non partitionnable. Son traitement sera associé à un jeton dans la place p_4 pour le premier niveau. Dans le modèle représenté, rien n'est dit sur la façon de stocker les messages en attente. On sait seulement qu'il seront traités un par un par le niveau 1.

Remarque : Il n'est pas cohérent d'associer les messages en attente à la place p_1 . D'une part cela voudrait dire qu'il y a toujours un message en attente, et un seul message en attente, dès que l'on franchit la transition t_d . D'autre part, cela introduirait une dissymétrie entre les niveaux. En effet pour le niveau 2 on distingue bien la place p_2 associée au fait que le niveau est disponible et la place p_7 qui contient les blocs de données à traiter. Si l'on veut raisonner sous la forme d'un réseau de Petri "autonome", il faut imaginer une place p_e en entrée de la transition t_a qui contient les blocs de données en entrée et une place p_s en sortie de t_f qui contient les blocs de résultats en sortie. Mais alors le problème de savoir comment les jetons sont introduits dans p_e et enlevés de p_s se posera. Il faut toujours établir une délimitation entre la partie du système pour laquelle on fait un modèle formel et son environnement.

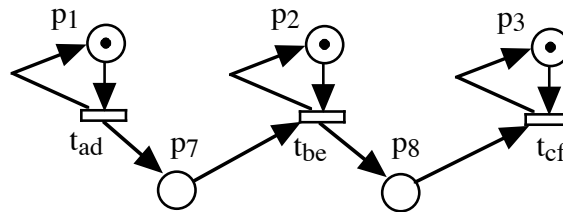


FIG. 5 – Réseau de la figure 1 après la première étape de réduction

b) La place p_4 est substituable. En effet, elle est la seule place en entrée de la transition t_d . Le franchissement de t_a est une condition suffisante pour le franchissement de t_d qui aura nécessairement lieu ensuite. De plus le marquage initial de la place p_4 est nul ($M_0(p_4) = 0$) et la transition de sortie de p_4 (t_d) possède au moins une place en sortie (la place p_1 par exemple). Donc nous aurons équivalence vis-à-vis des trois bonnes propriétés.

Il en est de même pour les places p_5 et p_6 . Attention il n'en est pas de même pour les places p_7 et p_8 car les transitions t_b et t_c ont plus d'une place en entrée. Nous obtenons le réseau de Petri de la figure 5.

Nous observons que, dans le réseau de Petri de la figure 5, les places p_1 , p_2 et p_3 ne sont reliées au reste du réseau de Petri que par des boucles élémentaires. Il s'agit de boucles élémentaires de poids 1 et donc les places seront implicites dégénérées si leur marquage initial vaut au moins 1. C'est-à-dire si :

$$M_0(p_1) \geq 1 \quad M_0(p_2) \geq 1 \quad M_0(p_3) \geq 1 \tag{1}$$

Ces conditions sont remplies. Nous obtenons donc le réseau de la figure 6 qui est équivalent au réseau de Petri de la figure 1 vis-à-vis des trois bonnes propriétés.

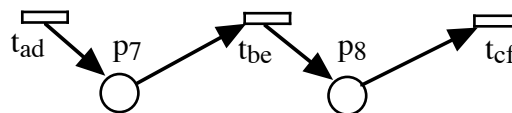


FIG. 6 – Réseau de la figure 1 après la deuxième étape de réduction

Nous observons que, dans le réseau de Petri de la figure 6, les places p_7 et p_8 sont substituables car leur transitions de sortie n'ont qu'une place en entrée. Elles ne sont pas initialement marquées

($M_0(p_7) = M_0(p_8) = 0$). Toutefois, si la transition (t_{bc}) de sortie de la place p_7 possède bien une place en sortie (la place p_8), il n'en est pas de même de celle de p_8 . En effet la transition t_{cf} est une transition *puits* (elle ne possède aucune place en sortie). Nous pouvons donc par réduction transformer un réseau de Petri non borné en un réseau de Petri borné. Remarquons que la transition t_{ad} , en entrée de la place p_7 , est une transition *source* (elle ne possède aucune place en entrée). Elle peut donc être franchie sans condition et la place p_7 ne peut pas être bornée. Le réseau de la figure 6 est donc *non borné*. Nous pouvons également remarquer que l'existence de la transition t_{cf} fait que le réseau de la figure 6 s'il est vivant, ne peut pas être borné. La réduction donne alors le réseau de la figure 7.



FIG. 7 – Réseau de la figure 1 après la troisième étape de réduction

Ce réseau est le réseau complètement réduit. Il est k -borné, vivant et réinitialisable pour son marquage initial. Nous déduisons donc que le réseau de Petri de la figure 1 est *non borné, vivant et réinitialisable* puisque lors de l'étape précédente nous avons montré que le réseau de la figure 6 était non borné.

Si le réseau de Petri est non k -borné pour son marquage initial, il n'est pas correct. Cela veut dire qu'il peut y avoir des places où un nombre infini de jetons s'accumulent dans certaines circonstances. On pressent que cela peut être le cas des registres tampons (buffers) p_7 et p_8 c'est-à-dire que la communication entre les niveaux 1, 2 et 3 est probablement incorrecte.

En effet, si nous exécutons la séquence $t_a; t_d$ de façon répétitive sans franchir t_b , le contenu en jeton de la place p_7 va augmenter indéfiniment. Si le rythme d'arrivée des données (franchissement de t_a) est grand et si la durée de l'exécution du niveau 1 est plus petite que celle du niveau 2, alors le registre tampon (buffer) entre les niveaux 1 et 2 débordera inéluctablement.

Remarque Dans la figure 6, nous aurions pu d'abord traiter la place p_7 comme une place substituable avec conservation des trois bonnes propriétés puis remarquer que le réseau obtenu est le réseau de Petri *parenthèse* qui est non borné. Puis, nous aurions pu éliminer la place p_8 en sachant que nous n'avions pas équivalence pour la propriété de k -borné. Nous aurions alors trouvé le réseau de Petri complètement réduit et le même résultat que ci-dessus.

c) Si le fonctionnement du niveau 2 est plus lent que celui des autres, pour assurer un débit correct il faut qu'il y ait deux tâches qui travaillent en parallèle (chacune prenant en moyenne un paquet de données sur deux) pour doubler le débit de ce niveau. Nous obtenons alors le réseau de Petri de la figure 8 qui est une représentation grossière (abstraite) dans le sens où l'on ne spécifie pas comment les messages arrivant dans le registre tampon (buffer) p_7 sont affectés aux tâches du niveau 2. Les deux tâches du niveau 2 sont en effet non différenciées dans ce modèle du système.

Remarque On pourrait envisager une coloration du réseau de Petri pour distinguer les deux tâches. Mais si l'on ne spécifie pas la politique d'affectation des tâches aux blocs de données en attente, cela

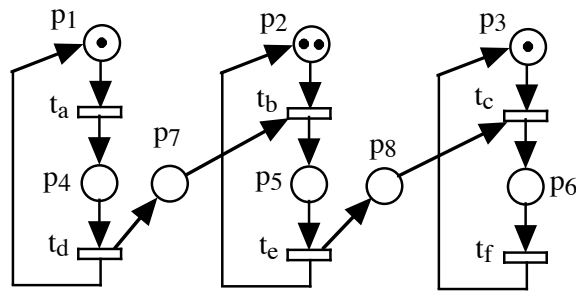


FIG. 8 – Réseau de Petri, avec doublement des tâches du niveau 2

n’apporte pas grand chose (voir le problème 3). Spécifier une politique d’affectation nécessiterait en particulier de conserver la date d’arrivée des blocs de données (c’est-à-dire des jetons) dans la place p_7 et cela devient vite complexe.

L’analyse par réduction du réseau de Petri de la figure 8 peut se faire en appliquant exactement les mêmes règles que celles qui ont été utilisées pour la réduction du réseau de Petri de la figure 1. Un point important est que la condition pour que la place p_2 soit implicite dégénérée est que son marquage initial soit supérieur ou égal à 1 ($M_0(p_2) \geq 1$). Comme nous avons $M_0(p_2) = 2$ elle est remplie.

Le réseau de Petri de la figure 8 est donc *non borné, vivant et réinitialisable*. Si, pour un débit d’arrivée des données donné et des durées de traitement des niveaux 1 et 2 donnés, le problème du débordement du tampon (buffer) p_7 peut sembler résolu, il peut réapparaître si l’environnement change (augmentation du débit d’arrivée des données et données plus simples entraînant un traitement du niveau 1 plus rapide). Le système spécifié continue donc d’être incorrect.

Correction du problème 2

d) Considérons maintenant le réseau de Petri de la figure 2. La place p_4 est substituable. En effet, elle est la seule place en entrée de la transition t_d . Le franchissement de t_a est une condition suffisante pour le franchissement de t_d qui aura nécessairement lieu ensuite. De plus le marquage initial de la place p_4 est nul ($M_0(p_4) = 0$) et la transition de sortie de p_4 (t_d) possède au moins une place en sortie (la place p_1 par exemple). Donc nous aurons équivalence vis-à-vis des trois bonnes propriétés. Il en est de même pour les places p_5 et p_6 . Nous obtenons le réseau de Petri de la figure 9.

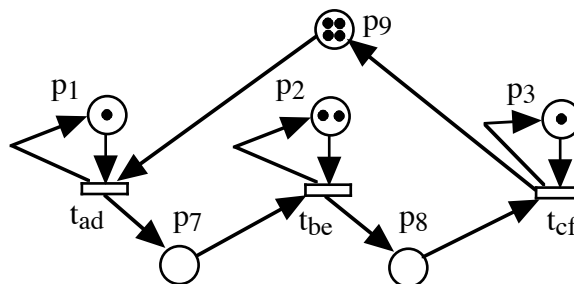


FIG. 9 – Réseau de la figure 2 après la première étape de réduction

Nous observons que, dans le réseau de Petri de la figure 9, les places p_1 , p_2 et p_3 ne sont reliées au reste du réseau de Petri que par des boucles élémentaires. Il s’agit de boucles élémentaires de poids

1 et donc les places seront implicites dégénérées si leur marquage initial vaut au moins 1. C'est le cas. Nous obtenons donc le réseau de la figure 10 qui est équivalent au réseau de Petri de la figure 2 vis-à-vis des trois bonnes propriétés.

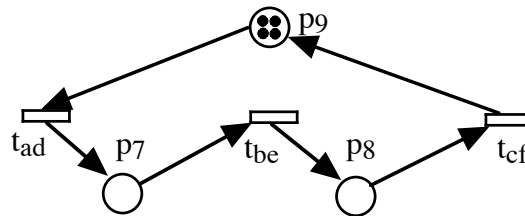


FIG. 10 – Réseau de la figure 2 après la deuxième étape de réduction

Nous observons que dans le réseau de Petri de la figure 10, maintenant, les places p_7 et p_8 sont substituables car leurs transitions de sortie n'ont qu'une place en entrée. Elles ne sont pas initialement marquées ($M_0(p_7) = M_0(p_8) = 0$). De plus, ce qui n'était pas le cas pour le réseau de Petri de la figure 6, leurs transitions de sorties (t_{be} et t_{cf} respectivement) possèdent au moins une place en sortie (p_8 et p_9 respectivement). Donc nous pouvons éliminer ces deux places et nous aurons un réseau de Petri réduit équivalent au réseau de Petri de la figure 2 vis-à-vis des *trois* bonnes propriétés. Nous obtenons le réseau de Petri de la figure 11.

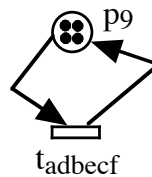


FIG. 11 – Réseau de la figure 2 après la troisième étape de réduction



FIG. 12 – Réseau de la figure 2 après la quatrième étape de réduction

Finalement, dans le réseau de Petri de la figure 11, nous pouvons remarquer que la place p_9 n'est reliée au reste du réseau de Petri que par une boucle élémentaire de poids 1. Comme son marquage initial est supérieur à ce poids, elle est implicite dégénérée. Son élimination conduit au réseau de Petri de la figure 12 qui est un réseau de Petri complètement réduit et qui est donc *k-borné*, *vivant et réinitialisable*. Cette spécification est donc maintenant *cohérente* et le système semble, a priori, *correct*.

e) Construisons d'abord la matrice d'incidence C du réseau de Petri de la figure 2 et cherchons la base des solutions de l'équation (inconnue f , T est le symbole de transposition des matrices) :

$$C^T \cdot f = 0 \tag{2}$$

$$C = \begin{matrix} & t_a & t_b & t_c & t_d & t_e & t_f \\ \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \end{matrix} & \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} \quad (3)$$

Elle n'a aucune colonne ne comprenant qu'une composante différente de zéro. Par contre toutes les colonnes ne comprennent qu'une seule composante non nulle d'un certain signe avec toutes les autres composantes non nulles du signe opposé. Nous pouvons par exemple éliminer la colonne t_a et la ligne p_4 après avoir remplacé la ligne p_1 par $p_1 + p_4$ et la ligne p_9 par $p_4 + p_9$ (p_4 est l'unique élément non nul positif et les composantes négatives sont p_1 et p_9).

Comme cela concerne des lignes différentes, nous pouvons sans re-écrire la matrice faire de même pour la colonne t_e et la ligne p_5 (seule composante non nulle négative) après le remplacement de la ligne p_2 par $p_2 + p_5$ et celui de la ligne p_8 par $p_5 + p_8$.

Nous obtenons alors la matrice suivante :

$$C_1 = \begin{matrix} & t_b & t_c & t_d & t_f \\ \begin{matrix} p_1 + p_4 \\ p_2 + p_5 \\ p_3 \\ p_6 \\ p_7 \\ p_5 + p_8 \\ p_4 + p_9 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \end{matrix} \quad (4)$$

Nous pouvons maintenant éliminer la colonne t_b et la ligne p_7 après avoir remplacé la ligne $p_5 + p_8$ par $p_5 + p_7 + p_8$. Comme cela porte sur des lignes différentes nous pouvons avant de re-écrire la matrice éliminer la colonne t_f et la ligne p_6 après avoir remplacé la ligne p_3 par $p_3 + p_6$ et la ligne $p_4 + p_9$ par $p_4 + p_6 + p_9$.

Nous obtenons alors la matrice suivante :

$$C_2 = \begin{matrix} & t_c & t_d \\ \begin{matrix} p_1 + p_4 \\ p_2 + p_5 \\ p_3 + p_6 \\ p_5 + p_7 + p_8 \\ p_4 + p_6 + p_9 \end{matrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \end{matrix} \quad (5)$$

Nous pouvons maintenant éliminer (par exemple) la colonne t_c et la ligne $p_5 + p_7 + p_8$ après

avoir remplacé la ligne $p_4 + p_6 + p_9$ par $p_4 + p_5 + p_6 + p_7 + p_8 + p_9$. Nous obtenons la matrice :

$$C_3 = \begin{matrix} & & & & t_d \\ & & & & \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \\ & p_1 + p_4 & & & \\ & p_2 + p_5 & & & \\ & p_3 + p_6 & & & \\ p_4 + p_5 + p_6 + p_7 + p_8 + p_9 & & & & \end{matrix} \quad (6)$$

La matrice C_3 ayant toutes ses composantes nulles, nous avons obtenu une base des solutions de l'équation 2, c'est-à-dire une base de composantes conservatives positives. Comme cette base a été obtenue en ne faisant que des combinaisons linéaires positives de lignes, il s'agit d'une base de composantes conservatives positives. Elle est formée de quatre vecteurs et est donnée ci-dessous.

$$B = \begin{matrix} & f^1 & f^2 & f^3 & f^4 \\ p_1 & \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{array} \right] \\ p_2 & \\ p_3 & \\ p_4 & \\ p_5 & \\ p_6 & \\ p_7 & \\ p_8 & \\ p_9 & \end{matrix} \quad (7)$$

Cette base réalise une couverture des places (toute place appartient à au moins une composante conservative positive) et donc le réseau de Petri de la figure 2 est k-borné pour tout marquage initial. Tout risque de débordement est donc *structurellement* écarté. De cette base on peut déduire les 4 p-invariants suivants :

$$M(p_1) + M(p_4) = 1 \quad (8)$$

$$M(p_2) + M(p_5) = 2 \quad (9)$$

$$M(p_3) + M(p_6) = 1 \quad (10)$$

$$M(p_4) + M(p_5) + M(p_6) + M(p_7) + M(p_8) + M(p_9) = 4 \quad (11)$$

De l'équation 11 nous pouvons déduire (les marquages sont des nombres positifs ou nuls) :

$$M(p_7) \leq 4 \quad (12)$$

$$M(p_8) \leq 4 \quad (13)$$

Cela veut dire qu'au plus 4 données peuvent être à un instant donné dans chacun des tampons. On peut donc les dimensionner de façon certaine et simple.

Si l'on dispose d'un mécanisme d'allocation de mémoire dynamique et que chaque paquet de données demande une place importante en mémoire, on peut remarquer que :

$$M(p_7) + M(p_8) \leq 4 \quad (14)$$

et donc qu'il suffit de disposer *globalement* de 4 blocs de mémoire pour les deux tampons (buffers).

f) Contrairement aux places p_7 et p_8 , la place p_9 ne correspond pas à un tampon qui va stocker des données entre les niveaux 3 et 1. Il s'agit d'un sémaphore qui va bloquer le fonctionnement du niveau 1 lorsqu'il y a *globalement* 4 paquets de données dans le système. Ces paquets sont soit en attente dans les tampons p_7 et p_8 , soit en cours de traitement dans les niveaux 1, 2 ou 3. Ce fonctionnement est formellement défini par l'invariant de place 11. En effet nous aurons $M(p_9) = 0$ et blocage de l'entrée du niveau 1 (franchissement de t_a) exactement lorsque

$$M(p_4) + M(p_5) + M(p_6) + M(p_7) + M(p_8) = 4 \quad (15)$$

Les bornes données par les invariants de places 8, 9 et 10 pour les places p_4 , p_5 et p_6 (traitements dans les niveaux 1, 2 et 3) sont :

$$M(p_4) \leq 1 \quad M(p_5) \leq 2 \quad M(p_6) \leq 1 \quad \text{ce qui donne} \quad M(p_4) + M(p_5) + M(p_6) \leq 4 \quad (16)$$

Elle sont donc plus restrictives que celles données pour ces mêmes places par l'invariant de places 11 :

$$M(p_4) \leq 4 \quad M(p_5) \leq 4 \quad M(p_6) \leq 4 \quad \text{avec} \quad M(p_4) + M(p_5) + M(p_6) \leq 4 \quad (17)$$

Cela veut dire que la contrainte d'utilisation maximale des ressources est imposée par le nombre de ressources disponibles et non par le mécanisme de protection contre les débordements des registres tampons.

Si l'on ne mettait initialement qu'un seul jeton dans la place p_9 , l'invariant de places 11 deviendrait :

$$M(p_4) + M(p_5) + M(p_6) + M(p_7) + M(p_8) + M(p_9) = 1 \quad (18)$$

Les bornes des places concernant les traitements en cours données par cet invariant deviendraient :

$$M(p_4) \leq 1 \quad M(p_5) \leq 1 \quad M(p_6) \leq 1 \quad \text{avec} \quad M(p_4) + M(p_5) + M(p_6) \leq 1 \quad (19)$$

On aboutirait à la situation absurde où le mécanisme de protection contre les débordements empêcherait les deux tâches du niveau 2 d'être actives simultanément, et même empêcherait tout traitement en parallèle entre les trois niveaux. C'est tout l'intérêt de la structure pipe-line qui serait perdu.

Les bornes données par les équations 16 et 17 montrent que le marquage initial de la place p_9 ne doit pas être inférieur à 4. Il semblerait même nettement souhaitable de le prendre égal à 6 pour pouvoir avoir un paquet de données en attente dans chaque tampon et un paquet de données en traitement dans chacune des tâches (deux tâches pour le niveau 2) du système. En effet, cette situation correspond à :

$$M(p_4) + M(p_5) + M(p_6) + M(p_7) + M(p_8) = 6 \quad (20)$$

et si l'on met initialement 6 jetons dans la place p_9 , c'est cette condition qui impliquera $M(p_9) = 0$ et le blocage de l'entrée du système (transition t_a).

Correction du problème 3

g) L'ensemble des couleurs est donc :

$$\mathcal{Coul} = \{niv1, niv21, niv22, niv3, entree, 1vers2, 2vers3\} \quad (21)$$

La place p_{10} du réseau de Petri de la figure 3 correspond au repliage des places p_1 , p_2 et p_3 . Mais la place p_2 contient initialement deux jetons qui représentent les deux tâches de ce niveau. Au vu de l'ensemble des couleurs \mathcal{Coul} proposé, il est clair que l'énoncé suggère de différencier ces deux tâches. Nous aurons donc 4 couleurs pour la place p_{10} (une pour p_1 , deux pour p_2 et une pour p_3).

De par les fonctions associées aux arcs du réseau coloré de la figure 3, nous déduisons que les sous-ensembles de couleurs des places p_{10} et p_{20} ainsi que des transitions t_{aa} et t_{dd} sont identiques. Cela semble logique puisque p_{20} correspond au repliage des places p_4 , p_5 et p_6 et que la place p_5 peut contenir deux jetons. La transition t_{aa} correspond au repliage des transitions t_a , t_b et t_c et la transition t_{dd} a celui des transitions t_d , t_e et t_f . Nous avons donc :

$$\mathcal{Csec}(p_{10}) = \mathcal{Csec}(p_{20}) = \mathcal{Csec}(t_{aa}) = \mathcal{Csec}(t_{dd}) = \{niv1, niv21, niv22, niv3\} \quad (22)$$

La place p_{70} correspond au repliage des places p_7 , p_8 et p_9 . L'énoncé suggère clairement de coder directement ces places (p_7 par $1vers2$, p_8 par $2vers3$ et p_9 par $entree$) et de ne différencier ni les 4 jetons dans la place p_9 ni les paquets de données dans les tampons p_7 et p_8 qui seront tous traités de la même façon dans cette représentation abstraite. Nous avons donc :

$$\mathcal{Csec}(p_{70}) = \{entree, 1vers2, 2vers3\} \quad (23)$$

Pour exprimer la fonction (décrite par une matrice) id , prenons par exemple le cas de l'arc (p_{10}, t_{aa}) . Les lignes de la matrices sont les éléments de $\mathcal{Csec}(t_{aa})$ définissant les diverses manières de franchir cette transition. Les colonnes sont les éléments de $\mathcal{Csec}(p_{10})$ décrivant les couleurs des jetons pouvant être dans cette place. La matrice id associe à chaque façon de franchir la transition t_{aa} les jetons consommés dans la place p_{10} . Nous avons :

$$id = \begin{matrix} & & niv1 & niv21 & niv22 & niv3 \\ \begin{matrix} niv1 \\ niv21 \\ niv22 \\ niv3 \end{matrix} & \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] & & & & \end{matrix} \quad (24)$$

Les autres fonctions id sont toutes identiques puisqu'elles sont associées à des arcs (indépendamment de leur orientation) reliant des places et des transitions qui ont toutes le même sous ensemble de couleurs (celui donné par l'équation 22).

Il n'en est pas de même pour les arcs entrant et sortant de la place p_{70} . La fonction $prec$ décrit les jetons consommés par les transitions t_a , t_b et t_c dans les places p_9 , p_7 et p_8 respectivement. La fonction $souv$ décrit les jetons produits par les transitions t_d , t_e et t_f dans ces mêmes places. Les lignes des matrices sont les mêmes que pour la matrice id puisque ce sont les éléments de $\mathcal{Csec}(t_{aa})$ et de $\mathcal{Csec}(t_{dd})$. Par contre les colonnes sont les éléments de $\mathcal{Csec}(p_{70})$. Nous avons :

