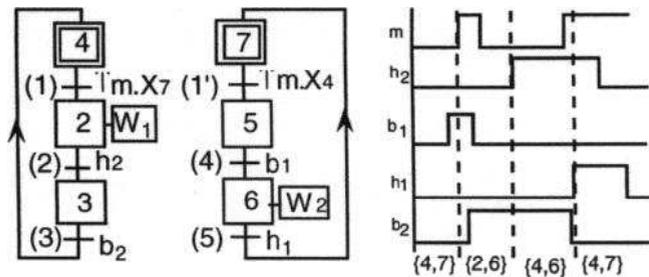


**Exercice corrigé d'Informatique du temps réel**  
**Date : 28 février 2000**  
**corrigé indicatif**

1) Donner les situations successives de l'exemple ci-dessous



R1) les situations sont indiquées sur le diagramme ci-dessus à droite.

2) Donner la traduction en pseudo-code du GRAFCET de la question 1 (exécutif par état).

R2) Le graphe peut être représenté par deux branches donc deux variables d'état. Pour le reste la traduction suit le schéma présenté à la figure 9.19 du cours.

3) Dans le Grafcet de la question 1, le comportement sera-t-il différent si l'on utilise l'algorithme d'interprétation avec recherche de stabilité ou celui sans recherche de stabilité ? (justifier la réponse)

R3) L'algorithme d'interprétation sans recherche de stabilité va ajouter deux situations stables {2,5} et {3,6} alors que l'autre algorithme ne va pas les donner. Le seul effet visible sera un léger retard dans l'action W2.

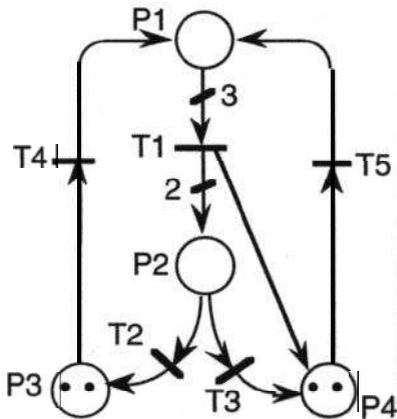
4) Le jeu de tâches périodiques du tableau ci-dessous est-il ordonnançable ? (justifier la réponse)

R4) Il est assez facile de créer un ordonnancement satisfaisant en utilisant l'algorithme EDF. La durée du cycle est de 10 unités. Les tâches sont exécutées selon l'ordre: A, B (1 unité), A, B (fin), C, A, A, C, A.

tâche	T	D	C
A	2	2	1
B	10	3	2
C	5	5	1

5) Le réseau de Petri ci-dessous est-il borné ? (justifier la réponse)

R5) On peut vérifier que l'invariant linéaire de ce réseau vaut  $m(P1) + m(P2) + m(P3) + m(P4) = 4$  et donc que le réseau est borné.



6) Quels registres doivent être sauves lors d'une interruption ? (justifier votre réponse)

R6) Il faut préserver tous les registres utilisés dans la routine d'interruption et dans les procédures qu'elle appelle. Le travail peut être facilité par le fait que les compilateurs effectuent certaines sauvegardes.

7) Le jeu de tâches périodiques ci-dessous peut-il être ordonnancé de manière non préemptive ? (justifier votre réponse)

R7) Il peut l'être à l'aide de l'algorithme EDF non préemptif car la fonction de garantie est vérifiée.

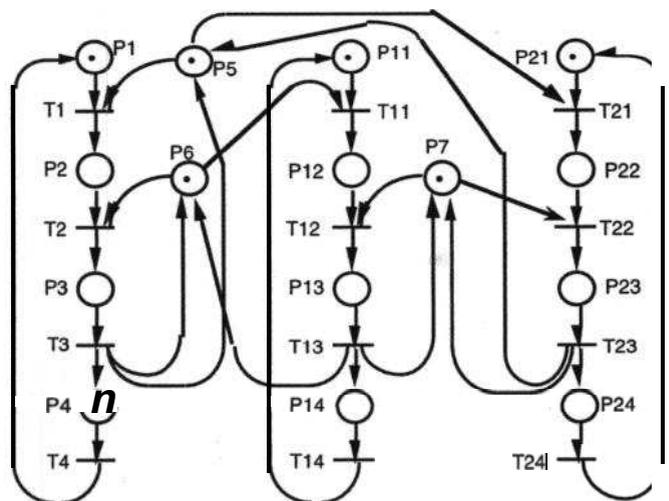
tâche	T	D	C
A	14	14	6
B	20	20	6
C	30	30	6

8) Quel problème résout l'héritage de priorité ?

R8) Le problème de l'inversion de priorité. L'inversion de priorité se produit quand plusieurs tâches utilisent un même verrou ou un même sémaphore. Une tâche en attente du verrou peut être retardée par une tâche de plus basse priorité qui ne possède pas le verrou. Ce phénomène se produit si la tâche qui possède le verrou est de plus basse priorité que la tâche bloquée et qu'une tâche de plus haute priorité que cette dernière est prête. Si le système base ses décisions d'ordonnancement sur les priorités relatives des tâches, il choisira la tâche de plus haute priorité. Si cette dernière, qui n'est pas impliquée dans la prise du verrou, est de plus basse priorité que la tâche en attente du verrou, il y a inversion de priorité. Le phénomène est donc dû à une limitation du mécanisme des priorités.

9) 3 tâches partagent des ressources protégées par des verrous S1, S2 et S3. La tâche A prend les verrous S1 puis S2 pour les relâcher ensuite simultanément. La tâche B prend les verrous S2 puis S3 pour les relâcher ensuite simultanément. La tâche C prend les verrous S1 puis S3 pour les relâcher ensuite simultanément. Utiliser les réseaux de Petri pour modéliser ce comportement.

R9) Le RdP ci-dessous donne une représentation possible.

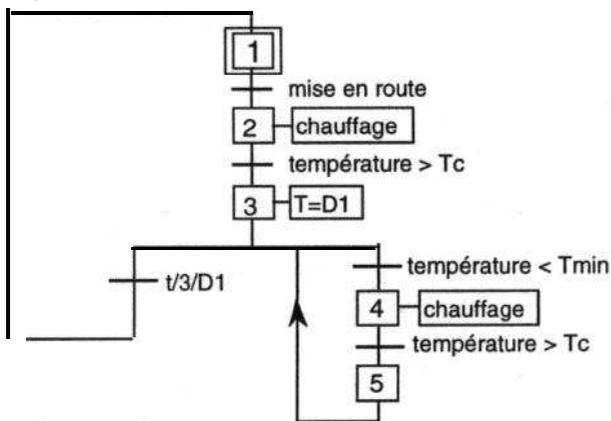


- 10) Montrer qu'il ne peut y avoir un blocage fatal dans le système décrit à la question 9.
- R10) Une analyse par graphe d'accessibilité montre que, depuis n'importe quelle situation accessible, toutes les transitions peuvent être franchies. Il n'y a donc aucun blocage possible.
- 11) Dans l'ensemble des tâches ci-dessous, la tâche B est destinée à traiter un événement selon la technique de la scrutation, Quel sera le temps minimum et le temps maximum écoulé entre l'apparition de l'événement et la fin du traitement par la tâche B ? (justifier la réponse)
- R11) Le calcul des temps de réponse avec l'algorithme DM montre que l'ensemble est ordonnançable selon DM. La réalisation de l'ordonnancement montre que la tâche B s'exécute entre  $t=7$  et  $t=10$  puis entre  $t=17$  et  $t=20$ . Si les temps d'exécution des tâches sont constants, le temps de réponse aux événements sera compris entre 3 et 23 unités de temps. Dans le cas où les temps d'exécution des tâches peuvent varier, le temps de réaction sera compris entre 0 et 25 unités de temps.

tâche	T	D	C	P(DM)	R(DM)
A	10	8	5	2	7
B	15	10	3	1	10
C	5	4	1	3	1

- 12) Est-il possible de respecter la politique d'ordonnancement EDF avec un système d'exploitation où les priorités affectées à une tâche sont fixes ? (justifier la réponse)
- R12) Oui, cela est possible en calculant les priorités comme sont déterminés niveaux de préemption dans l'algorithme SRP.
- 13) Vous disposez du code source mais pas de la cible. Quelles possibilités s'offrent à vous pour déterminer le temps d'exécution de votre programme ? (justifier la réponse)
- R13) La simulation et le calcul.
- 14) Un appareil permet de cuire des oeufs. L'utilisateur met en place l'oeuf, ferme le couvercle, choisit la durée D1 et fait démarrer la cuisson. Le système de contrôle doit alors enclencher le chauffage de l'eau. Il doit arrêter le système D1 unités de temps après que la température de cuisson Tc (constante) a été atteinte. Pour éviter les accidents, le chauffage doit être arrêté dès que la température de cuisson a été atteinte. Il doit être remis en route aussitôt que la température descend en dessous d'un seuil Tmin. Modéliser le fonctionnement du contrôleur avec un GRAFCET fonctionnel.

R14)



- 15) Dans un ordonnancement à priorités fixes, on dit qu'à tout instant doit s'exécuter la tâche qui possède la priorité la plus élevée. Si un noyau temps réel est utilisé, il est dit dans le cours qu'un des instants auxquels les décisions d'ordonnancement sont prises est celui du relâchement d'un verrou. Donner deux raisons qui justifient cette affirmation ?
- R15) Lors du relâchement d'un verrou et si l'héritage de priorité est implanté, la tâche va voir sa priorité réduite. Il peut alors y avoir une tâche de plus haute priorité qui est prête. Même si l'héritage de priorité n'est pas actif, une tâche de plus haute priorité peut être en attente du verrou.
- 16) Pour le programme ci-dessous, indiquer sur un diagramme de temps quelles sont les coroutines actives à chaque instant.

MODULE Coroutines;

```

FROM SYSTEM IMPORTI
  PRGCESS, TRANSFER, NEWPRQCESS;

```

```

VAR pp,p1, p2,p3: PRQCESS;

```

```

PROCEDURE X;
BEGIN
  LOOP
    ....
    TRANSFER(p1,p2);
    ....
  END X;

```

```

PROCEDURE y;
BEGIN
  LOOP
    ....
    TRANSFER(p2,p3);
    ....
  END Y;

```

```

PRGCEDEURE Z;
BEGIN
  LOOP
    ....
    TRANSFER(p3,p1);
    ....
  END Z;

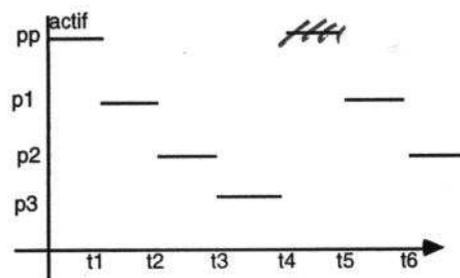
```

```

BEGIN
  ....
  NEWPROCESS(X,.....,p1);
  NEWPROCESS(Y,.....,p2);
  NEWPROCESS(Z,.....,p3);
  TRANSFER(pp,p1);
END Coroutines.

```

R16:

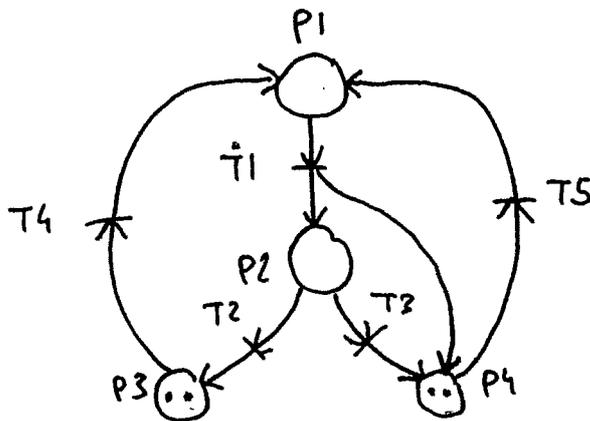


Exercice 5 2000

↖ G et Rdp p. 101-102

Le réseau de Pétri ci-dessous est-il borné ?

R-borné: le nb de jetons quelle que soit la place ne dépasse pas le pour tous les marquages de  $R(M_0)$



$$R(M_0) = \left\{ \begin{matrix} P1 & P2 & P3 & P4 \\ (0, 0, 2, 2) \\ (2, 0, 0, 0) \end{matrix} \right\}$$

Autre méthode: Invariant de place → G et Rdp p. 137-138

$$I = \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{matrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad \text{et} \quad C = \begin{matrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{matrix} \begin{vmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}$$

in les transitions

$$= (I - C) \cdot X^T = \begin{bmatrix} -1 & 1 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = 0 \Rightarrow \begin{cases} -P_0 + P_1 + P_3 = 0 \\ -P_1 + P_2 = 0 \\ P_0 - P_2 = 0 \\ P_0 - P_3 = 0 \end{cases}$$

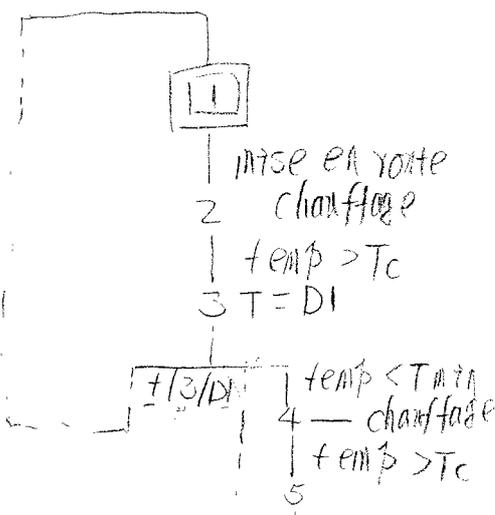
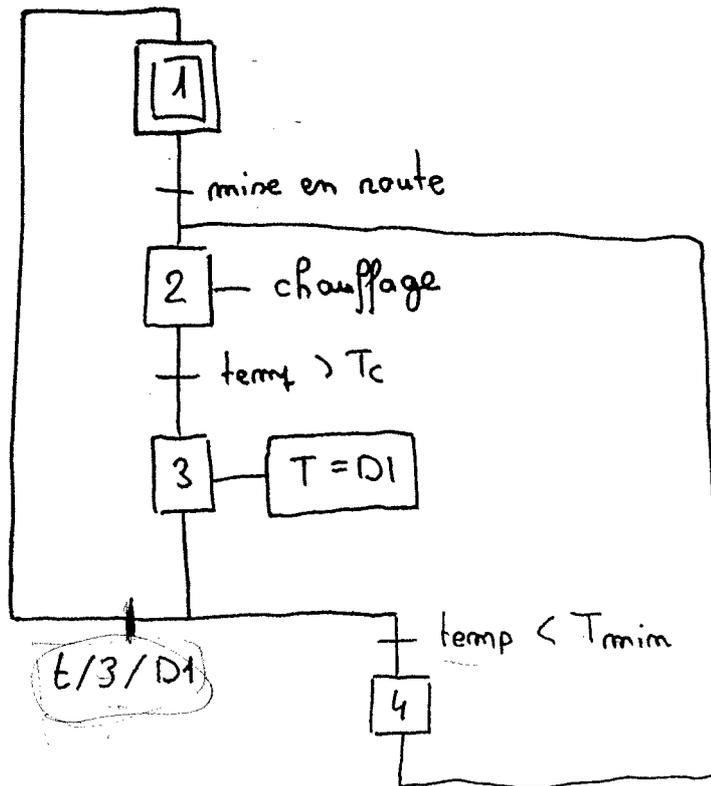
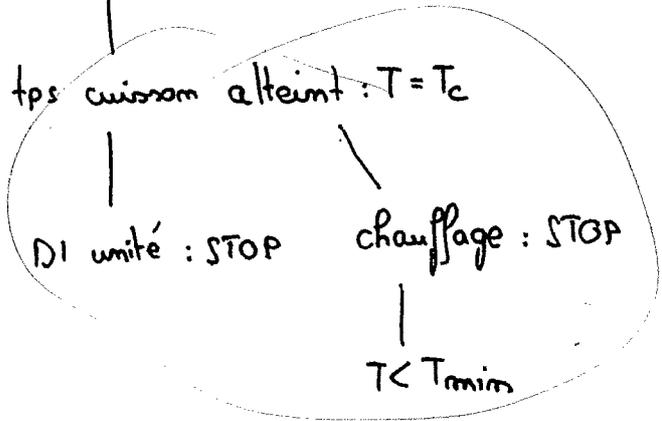
$-P_0 + P_1 + P_3 = 0$

Exercice 14 2000

Cuire des oeufs :

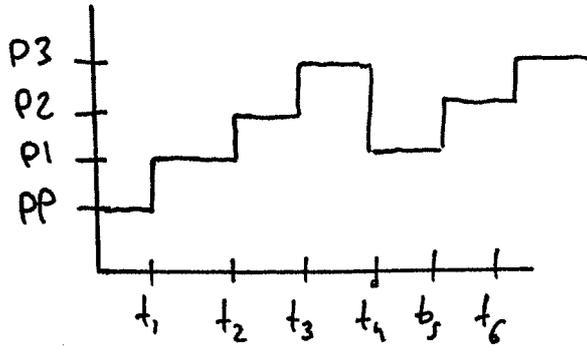
Place l'oeuf  
 ↓  
 ferme le couvercle  
 ↓  
 choix durée D1  
 ↓  
 démarrer cuisson

Chauffage de l'eau



# Exercice 16 2000

Indiquer sur un diagramme de temps quelles sont les conoutines actives à chaque instant



Avec les transferts:  $PP \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P1 \rightarrow P2 \rightarrow P3 \dots$