



République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel
Spécialité : GI

Thème

**Modélisation et simulation des chaînes de production
dans les Systèmes à Évènements Discrets :
Application à la chaîne de production de beure**

Présenté et soutenu publiquement par

MANSOURI Mohammed El Amine & MADANI Fateh

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
Mr. ADJLOUA Abdelaziz	MCB	Univ. Oran-2, IMSI	Président
Mr. Abderrahim BENFEKIR	MCB	Univ. Oran-2, IMSI	Encadreur
Mr. BENARBIA Taha	MCB	Univ. Oran-2, IMSI	Examineur

Juin 2022

Dédicaces

Tout d'abord, merci à Dieu Tout-Puissant d'avoir fait de ce travail une réalité.

Je tiens à exprimer mes sincères remerciements :

À mon père et ma mère pour leur soutien tout au long de ma vie.

À mes grands-parents pour leur soutien tout au long de mon parcours universitaire.

Je remercie tous mes frères, mes amis et mes collègues.

MADANI Fateh

Dédicaces

J'exprime ma reconnaissance et je dédie ce modeste travail à tous ceux que je n'arriverais jamais à leur exprimer mon amour sincère.

*Mon père "Abdellah", le professeur de ma vie qui m'a arrosé de tendresse et d'espoirs
Que Dieu ait pitié de lui.*

*A mon soutien dans ma vie, qui m'a appris, m'a supporté et ma dirigé vers la gloire ...ma
mère.*

A mon frère "Youcef", ma sœur "Assia", Petit "Abdellah", Que Dieu les protège.

*A ma famille **MANSOURI**, qui ma doté d'une éducation digne, son amour a fait de moi
ce que je suis aujourd'hui.*

MANSOURI Mohammed El Amine

Remerciements

Ce n'est pas la rédaction d'un tel rapport qui exige un remerciement, mais si on remercie des gens c'est par ce qu'ils méritent.

*Au terme de ce travail, nous tenons à remercier **M. A BENFKIR** pour le temps qu'il a consacré à nous apporter les outils méthodologiques indispensables à la conduite de cette recherche. Son exigence nous a grandement stimulés.*

*Nous adressons encore nos remerciements à ; **M. T. BENARBIA** et **M. A ADJLOUA** qui ont accepté de faire partie du jury.*

Nos remerciements vont également à tous les profs de l'Institut de Maintenance et de Sécurité Industrielle –Oran, qui nous ont enseignés durant notre formation universitaire.

Nous remercions également nos parents pour leur soutien moral et financier durant nos études.

Enfin, Merci à tous nos collègues, amis, et à tous ceux qui nous ont aidé et soutenu de près ou de loin.

Résumé

Nombreux procédés industriels sont des systèmes événement discret (SED) par nature, ce qui signifie que leur comportement résulte de l'évolution et de l'interaction de variable continue et de variables discrètes. Ces systèmes sont classiquement constitués de processus continus interagissant avec (ou régis par) des processus discrets.

Les objectifs que l'on peut assigner à l'étude des système événement discret à apporter une solution en termes de modèle, de méthode de performance et en particulier les fonctions de commande et de surveillance. Nous nous intéressons dans ce travail à la modélisation et à l'analyse des systèmes discret à dominante événementielle. L'approche de modélisation est basée sur l'utilisation des réseaux de Pétri temporisé (RdP). La méthode est illustrée sur un exemple industriel : unité de production du beurre.

Mots clé : Systèmes à événement discrets, Réseau de Pétri, Modélisation, Simulation, Chaîne de production du beurre.

Abstract

Many industrial processes are discrete event systems (DES) by nature, which means that their behavior results from the evolution and the interaction of continuous variable and discrete variables. These systems are classically made up of continuous processes interacting with (or governed by) discrete processes.

The objectives that can be assigned to the study of discrete event systems to provide a solution in terms of model, performance method and in particular the control and monitoring functions. We are interested in this work in the modeling and analysis of discrete systems with a dominant event. The modeling approach is based on the use of timed Petri nets (RdP). The method is illustrated on an industrial example: butter production unit.

Keywords: Discrete event systems, Petri net, Modeling, Simulation, Butter production chain.

Liste des abréviations

SED : Système à Évènements Discret.

RdP : Réseau de Petri.

\mathcal{P} : ensemble non vide de places d'un réseau de Petri.

\mathcal{T} : ensemble non vide de transitions d'un réseau de Petri.

m_p : nombre de jetons initialement contenus dans la place p .

τ_p : durée minimale de séjour d'un jeton dans la place p .

GET : Graphe d'Évènements Temporisé.

GETG : Graphe d'Évènements Temporisé Généralisé.

θ : vecteur T-invariant associé au graphe d'évènements temporisé généralisé.

ppcm : plus petit commun multiple.

M_{qp} : poids de l'arc allant de la place p vers la transition x_q .

$M_{pq'}$: poids de l'arc allant de la transition $x_{q'}$ vers la place p .

\mathcal{N}_p : marque valuée associée à la place p .

\mathcal{N}_q : marque valuée associée au circuit q .

\oplus : addition dans un dioïde.

\otimes : multiplication dans un dioïde.

\mathbb{Z}_{\max} : algèbre $(\mathbb{Z} \cup \{-\infty\}, \max, +)$.

$\bar{\mathbb{Z}}_{\max}$: algèbre $(\mathbb{Z} \cup \{\pm\infty\}, \max, +)$.

\mathbb{Z}_{\min} : algèbre $(\mathbb{Z} \cup \{+\infty\}, \min, +)$.

$\bar{\mathbb{Z}}_{\min}$: algèbre $(\mathbb{Z} \cup \{\pm\infty\}, \min, +)$.

\preceq : relation d'ordre liée à un dioïde.

a^* : étoile de Kleene.

$c_q(t)$: variable compteur associée à la transition c_q d'un graphe d'évènements temporisé.

$\mathcal{D}_{\min}[[\delta]]$: dioïde d'opérateurs permettant de modéliser les graphes d'évènements temporisés généralisés.

$\gamma^v, v \in \mathbb{Z} \cup \{\pm\infty\}$: opérateur de stock sur les compteurs.

$\delta^\tau, \tau \in \mathbb{Z} \cup \{\pm\infty\}$: opérateur de retard sur les compteurs.

$\mu_m, m \in \mathbb{Q}$: opérateur multiplicateur sur les compteurs.

$\bar{\mathbb{Z}}_{min}^z$: ensemble de signaux de \mathbb{Z} dans \mathbb{Z}_{min} .

$[a]$: partie entière supérieure de a .

$\lfloor a \rfloor$: partie entière inférieure de a .

C : matrice d'incidence.

K : P-invariant.

Θ : T-invariant.

M_0 : vecteur marquage initial.

$u.t$: unité de temps.

Sommaire

Dédicaces

Remerciements

Résumé

Sommaire

Liste des figures

Liste des Notations

Introduction générale.....	01
Chapitre I :Outils algébriques.....	02
1.1 Algèbre des dioïdes.....	02
1.2 Éléments de la théorie des systèmes linéaires.....	05
1.2.2 Systèmes élémentaires.....	07
1.2.3 Opérations sur les systèmes.....	08
1.3 Conclusion.....	09
Chapitre II: Introduction aux GETG.....	10
2.1 Les réseaux de Pétri.....	10
2.1.1 Définitions et notations.....	11
2.1.2 Equation fondamentale.....	12
2.1.3 Marquages accessibles.....	13
2.1.4 Concurrence et synchronisation dans les RdP.....	14
2.1.5 Quelques propriétés des RdP.....	15
2.1.6 Invariants d'un réseau de Pétri.....	16
2.2 Graphes d'Événements Temporisés.....	19
2.2.1 Conditions initiales.....	20
2.3 Graphes d'Événements Temporisés Généralisés.....	22
2.3.1 Gain d'un circuit	23

2.4 Conclusion	25
Chapitre III : Modélisation de la chaîne de production de beure dans des GETG	26
3.1 Modélisation des GET dans l'algèbre (min, +).....	26
3.1.1 Forme récurrente explicite.....	30
3.2 Modélisation des GETG.....	32
3.2.1 Equations récurrentes des GETG.....	32
3.2.2 Modélisation à base d'opérateurs des GETG.....	34
3.3 Modélisation de la chaîne de beure	39
3.3.1 Description du processus de fabrication.....	39
3.3.2 Modélisation graphique.....	40
3.3.3 Modélisation mathématique.....	41
3.4 Conclusion.....	42
4 Chapitre IV Simulation de la chaîne de production de beure dans PIPE2.....	44
4.1 Introduction	44
4.1.1 Définition du logiciel de simulation(PIPE 2)	44
4.2 Simulation	44
4.2.1 Classification	44
4.2.2 Les resultats d'analyse des invariants dans les Réseaux de pétri	45
4.2.3 Incidence et marquage du Réseaux de Pétri	46
4.2.4 Siphons minimaux et pièges minimaux.....	48
4.2.5 Graphe d'accessibilité/couverture	50
4.2.6 Simulation	51
4.2.7 Analyse de l'espace d'état	52
4.3 Conclusion	52
Conclusion générale.....	53
Références Bibliographies.....	54

Introduction Générale

On présentera ici quelques techniques de modélisation et de simulation des systèmes à événements discrets (SED). Les SED sont utilisés pour la modélisation de systèmes pour lesquels les techniques continues ne sont pas adaptées.

En effet la théorie classique des systèmes continus s'attache à la description des équations différentielles ou aux dérivées partielles, comme $x' = f(x,u)$, où x représente l'état, U une commande et où f est un opérateur causal suffisamment régulier pour que X soit définie presque partout. Ce type de modélisation n'est pas facilement utilisable pour prendre en compte les phénomènes de synchronisations (par exemple synchronisation d'arrivées de pièces pour un assemblage ou synchronisation de feux dans un réseau de transport), de transitions d'états (système informatique), de saturations (utilisation de stocks de dimension finie), gestion des conflits (utilisation d'une ressource par plusieurs utilisateurs) et de façon plus générale tous les phénomènes de nature discrète.

Les systèmes à événements discrets sont précisément des systèmes où se produisent des changements brusques d'état. Ces changements d'état interviennent à l'occurrence ou constituent, des événements discrets. Contrairement au cas des systèmes linéaires, il n'existe pas pour les SED de modèle naturel. De nombreuses théories complémentaires se sont développées et permettent chacune de traiter des phénomènes spécifiques : équations logiques, chaînes de Markov, réseaux de files d'attente, graphes d'événements, systèmes min-plus (ou max-plus) linéaires, systèmes de transitions, automates et langages...

L'objectif est de commencer à étudier une classe assez générale de SED, pour laquelle la théorie n'est pas très développée. Cette classe correspond aux systèmes modélisables par des réseaux de Pétri temporisés. Ce rapport, de nature essentiellement bibliographique, a pour objet de présenter quelques directions et cas particuliers pour lesquels des théories sont connues. Ce travail a pour prétexte une étude de cas : un réseau ferroviaire. En utilisant les différentes approches des SED on tentera dans un premier temps de modéliser le réseau, puis dans un second de le simuler afin d'en sécuriser le fonctionnement.

Les modèles de base utilisés pour l'étude des SED sont essentiellement graphiques : Grafcet, automates, réseaux de Pétri. Des modèles algébriques ont été développés, et utilisés dans une optique de commande, en particulier dans deux cas : les automates et les graphes d'événements. Il s'agit respectivement de la théorie des langages et de la supervision de Ramadge et Wonham, et de la théorie des systèmes max-plus linéaires. Ces théories utilisent les mêmes outils mathématiques : monoïdes, dioïdes, treillis, structures ordonnées.

Dans ce travail nous nous sommes intéressés à l'étude des systèmes linéaires constituent une classe de systèmes à événements discrets (SED) [2] pour lesquels il existe des

modèles linéaires (représentation d'état ou de type entrée-sortie) analogues dans la forme à ceux utilisés dans la théorie des systèmes linéaires conventionnelle.

Cette approche permet de considérer des problèmes de modélisation et de contrôle, adaptés à certaines classes de systèmes (systèmes manufacturiers avec des synchronisations, systèmes de transport avec correspondance, modèles d'échanges de données dans des réseaux informatiques, les chaînes de production industrielles [3], [4]), ceci dans l'esprit de l'automatique linéaire. Les travaux que nous avons traités dans ce thème ont principalement concerné la modélisation, la simulation, l'élaboration d'outils de calcul pour des systèmes décrits par des Graphes d'Événements Temporisés (GET), ou par des classes de réseaux de Petri englobant les GET. L'accent sera mis notamment sur la possibilité de modéliser une chaîne de production de beurre dans une unité industrielle et faire la simulation dans un environnement *PIPE2* logiciel de simulation des réseaux de Pétri.

Pour ce là, plus que l'introduction générale le plan de mémoire est organisé en quatre chapitres comme suit :

Le premier chapitre, la présentation des outils algébriques nécessaires à la modélisation et à la commande des graphes d'événements temporisés généralisé.

Le deuxième chapitre, présentation des réseaux de Petri, qui sont un langage graphique permettant de décrire les Systèmes à Évènements Discrets, Ensuite on a présenté une structure particulière des réseaux de Petri, appelée les Graphes d'Évènement Temporisés(GET) qui permet de modéliser les systèmes où interviennent des phénomènes de synchronisation et de temporisation, et finalement on introduit les GET généralisés.

Le troisième chapitre propose la modélisation des graphes généralisés dans des GETG, la modélisation Ensuite, on a présenté le modèle proposé de la chaîne de production de beurre qui modélisant le comportement dynamique de notre système de production dans l'algèbre de dioïde.

Le quatrième chapitre est consacré à la simulation du chaîne de production de beurre dans PIPE2 est l'analyse de performances des résultats obtenus.

Et finalement, nous terminons notre travail par une conclusion générale et les perspectives pour les travaux futurs.

Chapitre I

Outils algébriques

Dans cette première partie, on rappelle certaines propriétés propres à la structure algébrique d'un dioïde. Sans être assommant, nous présentons un ensemble de définitions, de notations et résultats relatifs à l'algèbre de dioïde.

1.1 Algèbre des dioïdes

Définition 1.1 On appelle un dioïde (ou semi-anneau idempotent) un ensemble \mathcal{D} muni de deux lois de compositions internes notées \oplus et \otimes telles que les axiomes suivants sont respectés :

- L'addition est associative : $(a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus b \oplus c$,
- L'addition est Commutative : $a \oplus b = b \oplus a$,
- L'addition est idempotente : $a \oplus a = a$,
- L'addition admet un élément neutre noté ε appelé zéro : $a \oplus \varepsilon = \varepsilon \oplus a = a$.
- La multiplication est associative : $(a \otimes b) \otimes c = a \otimes (b \otimes c) = a \otimes b \otimes c$,
- Distributive à gauche et à droite par rapport à l'addition :
 $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ et $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$,
- Le zéro est absorbant pour la multiplication : $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$,
- La multiplication admet un élément neutre noté e : $a \otimes e = e \otimes a = a$.

De plus, un semi-anneau idempotent, couramment noté $(\mathcal{D}; \oplus; \otimes)$, est commutatif lorsque la loi multiplicative \otimes est commutatif, soit lorsque $a \otimes b = b \otimes a$.

Exemple 1.1 (Dioïdes \mathbb{Z}_{\min}). $\mathbb{Z}_{\min} = (\mathbb{Z} \cup \{+\infty\}, \min, +)$ est un dioïde commutatif, où la loi additive \oplus correspond à l'opération *min* usuelle ; la loi multiplicative \otimes est équivalente à l'*addition* usuelle. L'élément zéro de \mathbb{Z}_{\min} est $\varepsilon = +\infty$ et l'élément identité est $e = 0$. Ce dioïde est appelé l'algèbre $(\min, +)$.

Application : On s'intéresse à la production de bicyclettes, on peut dire qu'une bicyclette est la "somme" d'une paire de roues et d'un cadre. Cette nouvelle "somme", notée \oplus pour la distinguer du + habituel, correspond à l'opération d'assemblage. On notera que l'on se permet alors d'"ajouter" des grandeurs ne s'exprimant pas dans les mêmes unités. Si on a 2 roues et 1 cadre, on ne peut fabriquer qu'une seule bicyclette, donc $1 \oplus 2 = 1$, ce qui est équivalent dans l'algèbre usuelle à la min $(2, 1) = 1$.

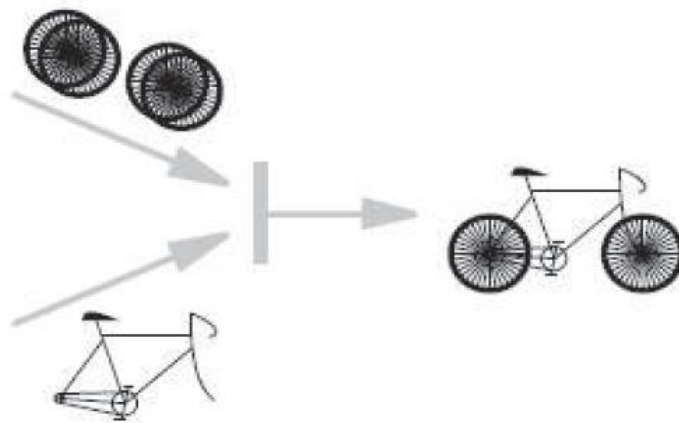


Figure 1.1 assemblage de bicyclette

Exemple 1.2 (Dioïdes \mathbb{Z}_{\max}). $\mathbb{Z}_{\max} = (\mathbb{Z} \cup \{+\infty\}, \max, +)$ est un dioïde commutatif, où la loi additive \oplus correspond à l'opération \max usuelle ; la loi multiplicative \otimes est équivalente à l'addition usuelle. L'élément zéro de \mathbb{Z}_{\min} est $\varepsilon = -\infty$ et l'élément identité est $e = 0$. Ce dioïde est appelé *l'algèbre (max, +)*.

Application : $1 \oplus 3 = 3 \Leftrightarrow \max(1, 3) = 3$

Définition 1.2 (Dioïde complet). Un dioïde $(\mathcal{D}, \oplus, \otimes)$ est complet si :

- Il est fermé pour les sommes infinies ;
- La loi \otimes distribue sur les sommes infinies, c'est-à-dire si pour tout $a \in \mathcal{D}$ et tout sous ensemble $\beta \subseteq \mathcal{D}$,

$$a \otimes \left(\bigoplus_{b \in \beta} b \right) = \bigoplus_{b \in \beta} (a \otimes b) \quad \text{Et} \quad \left(\bigoplus_{b \in \beta} b \right) \otimes a = \bigoplus_{b \in \beta} (b \otimes a)$$

Un dioïde complet admet un élément maximum $\bigoplus_{a \in \mathcal{D}} a$, que l'on notera T pour Top . Il est absorbant pour l'addition, autrement dit $\forall a \in \mathcal{D}, T \oplus a = T$.

Notons que, d'après la définition d'un dioïde, l'élément zéro ε est absorbant pour la multiplication pour tout élément de \mathcal{D} , aussi, on a : $T \otimes \varepsilon = \varepsilon \otimes T = \varepsilon$.

Exemple 1.3 On vérifie facilement que les dioïdes suivants sont complets :

$(\mathbb{Z} \cup \{-\infty, +\infty\}, \min, +)$ avec $\varepsilon = +\infty$, $e = 0$ et $T = -\infty$ est un dioïde noté par $\overline{\mathbb{Z}}_{\min}$

On a dans $\overline{\mathbb{Z}}_{\min}$ la règle suivante :

$$(-\infty) + (+\infty) = (+\infty).$$

$(\mathbb{Z} \cup \{-\infty, +\infty\}, \max, +)$ avec $\varepsilon = -\infty$, $e = 0$ et $T = +\infty$ est un dioïde noté par $\overline{\mathbb{Z}}_{\max}$

On a dans $\overline{\mathbb{Z}}_{\max}$ la règle suivante :

$$(-\infty) + (+\infty) = (-\infty).$$

Définition 1.3 (Dioïde matriciel). L'ensemble des matrices carrées de dimension n , à coefficients dans un dioïde $(\mathcal{D}, \oplus, \otimes)$, est un dioïde matriciel, noté $(\mathcal{D}^{n \times n}, \oplus, \otimes)$, où les opérations sont définies, à partir des opérations du dioïde $(\mathcal{D}, \oplus, \otimes)$ de manière analogue à l'algèbre classique, de la façon suivante :

$$\forall A, B \in \mathcal{D}^{n \times n};$$

$$A \oplus B : (A \oplus B)_{ij} = A_{ij} \oplus B_{ij}, \forall i, j = 1, \dots, n,$$

$$A \otimes B : (A \otimes B)_{ij} = \bigoplus_{k=1}^n A_{ik} \otimes B_{kj}, \forall i, j = 1, \dots, n,$$

L'élément identité de $\mathcal{D}^{n \times n}$ est la matrice, notée Id_n , composée de e sur la diagonale et de ε partout ailleurs. L'élément zéro est la matrice composée exclusivement de ε .

La somme et le produit de deux matrices de dimensions compatibles, pas nécessairement carrées, peuvent être définis de la façon suivante :

- $A \in \mathcal{D}^{n \times p}, B \in \mathcal{D}^{n \times p};$

$$A \oplus B : (A \oplus B)_{ij} = A_{ij} \oplus B_{ij} \quad \forall i = 1, \dots, n, \forall j = 1, \dots, p,$$

- $C \in \mathcal{D}^{n \times p}$, $B \in \mathcal{D}^{p \times q}$;

$$C \otimes B : (C \otimes B)_{ij} = \bigoplus_{k=1}^p A_{ik} \otimes B_{kj}, \quad \forall i = 1, \dots, n, \forall j = 1, \dots, q,$$

Exemple 1.4 Calcul Matriciel dans l'algèbre $(\max, +)$.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

- $a = (1 \otimes 5) \oplus (2 \otimes 7) = \max(1 + 5, 2 + 7) = 9$
- $b = (1 \otimes 6) \oplus (2 \otimes 8) = \max(1 + 6, 2 + 8) = 10$
- $c = (3 \otimes 5) \oplus (4 \otimes 7) = \max(3 + 5, 4 + 7) = 11$
- $d = (3 \otimes 6) \oplus (4 \otimes 8) = \max(3 + 6, 4 + 8) = 12$

1.2 Éléments de la théorie des systèmes linéaires

1.2.1 Notions sur les systèmes

Cette partie est consacrée à la présentation des éléments de la théorie des systèmes linéaires dans les dioïdes.

Définition : 1.4 (Signal). On appelle *signal* une application u de $\mathbb{Z} \rightarrow (\mathcal{D}, \oplus, \otimes)$.

$t \rightarrow u(t)$. On note $\mathcal{D}^{\mathbb{Z}}$ l'ensemble des signaux muni de deux opérations :

- Une loi additive interne, notée \oplus , définie par :

$$\forall u_1, u_2 \in \mathcal{D}^{\mathbb{Z}}, (u_1 \oplus u_2)(t) = u_1(t) \oplus u_2(t).$$

- Une loi externe, qui joue le rôle du produit d'un signal par un scalaire, notée « \bullet », définie par :

$$\forall a \in \mathcal{D}, \forall u \in \mathcal{D}^{\mathbb{Z}}, (a \bullet u)(t) = a \otimes u(t).$$

L'ensemble $\mathcal{D}^{\mathbb{Z}}$ est composé de signaux. En effet :

- Un signal est croissant s'il satisfait

$$\forall t \in \mathcal{D}, \forall \tau \in \mathbb{N}, u(t + \tau) \succcurlyeq u(t).$$

- Un signal est décroissant s'il satisfait

$$\forall t \in \mathcal{D}, \forall \tau \in \mathbb{N}, u(t + \tau) \preccurlyeq u(t).$$

Définition 1.5 (Système linéaire). Un système, noté \mathbb{S} , est un opérateur définie de $\overline{\mathbb{Z}}_{\min}^{\mathbb{Z}}$ vers $\overline{\mathbb{Z}}_{\min}^{\mathbb{Z}}$

$$\forall u_1, u_2 \in \mathcal{D}^{\mathbb{Z}}, \mathbb{S}(u_1 \oplus u_2) = \mathbb{S}(u_1) \oplus \mathbb{S}(u_2) \text{ (Propriété d'additivité)}$$

$$\forall u_1 \in \mathcal{D}^{\mathbb{Z}}, \forall a \in \mathcal{D}, \mathbb{S}(a \otimes u_1) = a \otimes \mathbb{S}(u_1) \text{ (Propriété d'homogénéité)}$$

Définition 1.6 (Produit de convolution). Le produit de convolution de deux signaux est noté \otimes et est défini par :

$$\forall u_1, u_2 \in \mathcal{D}^{\mathbb{Z}}, (u_1 \otimes u_2)(t) = \bigoplus_{\tau \in \mathbb{Z}} u_1(\tau) u_2(t - \tau) = \bigoplus_{\tau \in \mathbb{Z}} u_1(t - \tau) u_2(\tau)$$

Exemple 1.5 Le produit de convolution de deux signaux de $\overline{\mathbb{Z}}_{\min}^{\mathbb{Z}}$ satisfait la relation suivante :

$$\forall u_1, u_2 \in \overline{\mathbb{Z}}_{\min}^{\mathbb{Z}}, \forall t \in \mathbb{Z}, (u_1 \otimes u_2)(t) = \min_{\tau \in \mathbb{Z}} \{u_1(\tau) + u_2(t - \tau)\}$$

On parle de produit d'*inf*-convolution.

De manière duale, le produit de convolution de deux signaux de $\overline{\mathbb{Z}}_{\max}^{\mathbb{Z}}$ donne :

$$\forall u_1, u_2 \in \overline{\mathbb{Z}}_{\max}^{\mathbb{Z}}, \forall t \in \mathbb{Z}, (u_1 \otimes u_2)(t) = \max_{\tau \in \mathbb{Z}} \{u_1(\tau) + u_2(t - \tau)\}$$

On parle de produit de *sup*-convolution.

Définition 1.7 (Système continu). Un système \mathbb{S} est continu si pour tout sous-ensemble fini ou infini $(u_i)_{i \in I}$ d'éléments de $\mathcal{D}^{\mathbb{Z}}$, on a :

$$\mathbb{S}\left(\bigoplus_{i \in I} u_i\right) = \bigoplus_{i \in I} \mathbb{S}(u_i)$$

Sachant que $(\mathcal{D}, \oplus, \otimes)$ est dioïde complet.

Définition 1.8 (Système causal). Un système \mathbb{S} est causal si pour toutes entrées u_1 et u_2 , on a :

$$\forall \tau \in \mathbb{Z}, u_1(t) = u_2(t) \text{ pour } t \leq \tau \Rightarrow [\mathbb{S}(u_1)](t) = [\mathbb{S}(u_2)](t) \text{ pour } t \leq \tau.$$

1.2.2 Systèmes élémentaires

On présente dans cette partie deux systèmes élémentaires linéaires :

- Système gain Γ^v défini de $\overline{\mathbb{Z}}_{min}^{\mathbb{Z}}$ vers $\overline{\mathbb{Z}}_{min}^{\mathbb{Z}}$ par :

$$\forall t \in \mathbb{Z}, \forall u, y \in \overline{\mathbb{Z}}_{min}^{\mathbb{Z}}, v \in \overline{\mathbb{Z}}, y(t) = [\Gamma^v(u)](t) = v \otimes u(t).$$

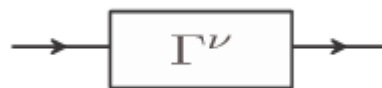


Figure 1.2. Système « Γ^v ».

- Système retard Δ^τ défini de $\overline{\mathbb{Z}}_{min}^{\mathbb{Z}}$ vers $\overline{\mathbb{Z}}_{min}^{\mathbb{Z}}$ par :

$$\forall t \in \mathbb{Z}, \forall u, y \in \overline{\mathbb{Z}}_{min}^{\mathbb{Z}}, \tau \in \overline{\mathbb{Z}}, y(t) = [\Delta^\tau(u)](t) = u(t - \tau).$$



Figure 1.3 Système « Δ^τ ».

1.2.3 Opérations sur les systèmes

Il existe trois opérations élémentaires permettant de mettre en relation des systèmes : la composition série, la composition parallèle et la mise en feedback.

- Composition série : $\mathbb{S} = \mathbb{S}_1 \otimes \mathbb{S}_2$ est définie par :



Figure 1.4 Composition série.

$$[\mathbb{S}(u)](t) = [\mathbb{S}_2(\mathbb{S}_1(u))](t).$$

- Composition parallèle : $\mathbb{S} = \mathbb{S}_1 \oplus \mathbb{S}_2$ est définie par :

Figure 1.4 Composition série.

$$[\mathbb{S}(u)](t) = [\mathbb{S}_1(u)](t) \oplus [\mathbb{S}_2(u)](t).$$

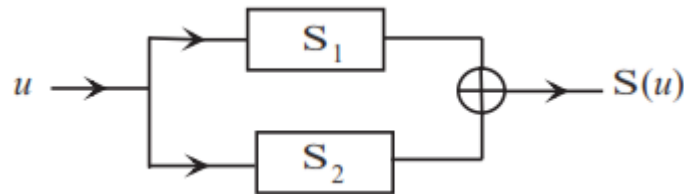


Figure 1.5 Composition parallèle.

- Mise en feedback : On considère le schéma suivant, où \mathbb{S}_1 et \mathbb{S}_2 sont deux opérateurs linéaires.



Figure 1.6 Mise en feedback.

La relation entre la sortie y et l'entrée u satisfait :

$$y = \mathbb{S}_1(\mathbb{S}_2(y) \oplus u) = \mathbb{S}_1(\mathbb{S}_2(y)) \oplus \mathbb{S}_1(u).$$

1.3 Conclusion

Nous avons présenté dans ce chapitre les outils mathématiques utilisés par la suite. Après un bref rappel sur la structure algébrique des dioïdes, et nous avons donné un ensemble de définitions sur la théorie des systèmes utiles pour aborder, dans le chapitre suivant, la représentation de certains systèmes dynamique, modélisables dans un dioïde d'opérateurs.

Chapitre II

Introduction aux GETG

Il y'a des nombreuses classes de modèles sont proposées pour l'étude des systèmes de production. Le choix d'un type de modèle est en fonction des objectifs recherchés. Dans le cas des systèmes de production, des concepts de modélisation ont été élaborés : par exemple, les chaîne de Karkov [Baynat, 2000], les files d'attentes [Bayant, 2000] ou bien les réseaux de Petri [Murata, 1989, Porth and Xie, 1995].

Certains systèmes à évènement discrets (SED) qui mettent en jeu uniquement des phénomènes de synchronisation peuvent être décrits de façon linéaire dans certains dioïdes. Ces systèmes peuvent également être représentés par un modèle graphique appelé graphe d'évènements temporisé (GET). Cette représentation découle directement de la théorie des réseaux de pétri.

2.1 Les réseaux de Petri

On présente sommairement dans la première section de ce chapitre les réseaux de Petri, qui sont un langage graphique permettant de décrire les Systèmes à Evènements Discrets (SED). L'emploi des réseaux de Petri pour la spécification, la simulation et l'analyse de ces systèmes remonte aux années 60 et s'est maintenant généralisé. La littérature sur les réseaux de Petri généraux est très abondante. On s'est limité aux préliminaires indispensables dans cette partie.

Les réseaux de Petri (RdP) constituent un formalisme graphique propre à la modélisation des Systèmes à Evènements Discrets (SED) introduit en 1962 par Carl Adam Petri [Petri 1962]. Ils sont particulièrement adaptés à l'étude des processus complexes mettant en jeu des propriétés de synchronisme et de partage de ressources. Leur support mathématique a permis en outre d'établir de nombreux résultats analytiques. Le lecteur intéressé par plus

amples développements trouvera une présentation générale et des références bibliographiques dans [Murata 1989], [Proth 1995] et [David 1992]. Pour l'étude des SED dans l'algèbre des dioïdes, les RdP sont souvent utilisés comme un outil de modélisation intermédiaire.

2.1.1 Définitions et notations

Définition 2.1 : Un réseau de Petri est un graphe biparti constitué de deux types de sommets : places (représentées par des cercles) et transitions (représentées par des barres). Des arcs orientés relient certaines places à certaines transitions, ou certaines transitions à certaines places. A chaque arc on associe un poids (entier positif). Le nombre de places (resp. de transitions) est fini, et non nul. Chaque place, peut contenir un ou plusieurs jetons (représentés par des points) qui modélisent la dynamique du système. A chaque place on associe une temporisation. Cette dernière correspond au temps de séjour des jetons dans une place.

D'une façon plus formelle, un RdP est un 5-tuplet $\mathcal{R} = (\mathcal{P}, \mathcal{T}, \text{Pré}, \text{Post}, \tau)$, où :

- \mathcal{P} (resp, \mathcal{T}) est un ensemble non vide et fini de place (resp, transitions) ;
- $\text{Pré} : \mathcal{P} \times \mathcal{T} \rightarrow \mathbb{N}$ une application d'incidence avant : $\text{Pré}(p ; c_q)$ contient la valeur entière M_{qp} associée à l'arc allant de la place p à la transition c_q ;
- $\text{Post} : \mathcal{P} \times \mathcal{T} \rightarrow \mathbb{N}$ une application d'incidence arrière : $\text{Post}(p ; c_q)$ contient la valeur entière M_{pq} , associée à l'arc allant de la transition c_q à la place p ;
- $\tau \in \mathbb{N}^{\mathcal{P}}$ définit les temporisations associées aux places : τ_p désigne la durée minimale de séjour d'un jeton dans la place p .

Le marquage \mathcal{M} d'un réseau de Petri est une application $\mathcal{M} : \mathcal{P} \rightarrow \mathcal{N}$. La quantité $\mathcal{M}(p)$ détermine le marquage de la place p . un réseau marqué est déterminé par le couple $\mathcal{N} = (\mathcal{R}, \mathcal{M}_0)$ formé d'un réseau de Petri \mathcal{R} est d'un marquage initial \mathcal{M}_0 .

Une transition sans place en amont est dite transition source, une transition sans place en aval est dite transition puits. La figure 2.1 représente un RdP ordinaire (tous les poids sont unitaires) dont les places sont représentées par des cercles et les transitions par des barres.

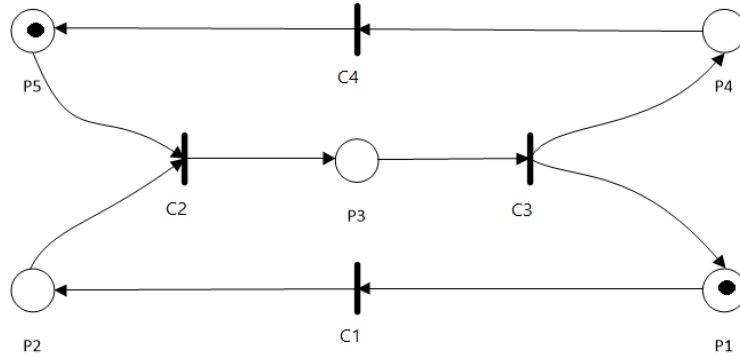


Figure 2.1 Représentation graphique d'un RdP

2.1.2 Equation fondamentale

L'évolution d'un RdP est décrite par le franchissement des transitions par des jetons séjournant dans les places du graphe.

Considérant un RdP et un marquage m , on dit que la transition $c_q \in \mathcal{T}$ est franchissable pour le marquage \mathcal{M} si l'on a :

$$\forall p \in \mathcal{P}, \mathcal{M}(p) \geq \text{Pré}(p, c_q).$$

Lorsque cette condition est validée, le *franchissement* de la transition c_q conduit à nouveau marquage, \mathcal{M}' , qui est nouvel état du modèle, défini par :

$$\mathcal{M}'(p) = \mathcal{M}(p) - \text{Pré}(p; c_q) + \text{Post}(p; c_q).$$

Exemple 2.1 Considérons le RdP de la figure 2.1. En partant du marquage $\mathcal{M}_0^t = (1, 0, 0, 0, 1)$ ($\mathcal{M}(p_1)=1, \mathcal{M}(p_2)=0, \mathcal{M}(p_3)=0, \mathcal{M}(p_4)=0, \mathcal{M}(p_5)=1$), On atteint le marquage $\mathcal{M}_1^t = (0, 1, 0, 0, 1)$, par franchissement de la transition c_1 .

Plus généralement, on note $[\mathcal{C}^+] = [\text{Post}(p; c_q)]$ (élément de la matrice d'incidence arrière) et $[\mathcal{C}^-] = [\text{Pré}(p; c_q)]$ (élément de la matrice d'incidence avant), $\mathcal{C} = \mathcal{C}^+ - \mathcal{C}^-$ (la matrice d'incidence) et en considérant \mathcal{F} une séquence franchissement réalisable à partir d'un marquage \mathcal{M}_i , on a l'équation fondamentale suivante, qui traduit la dynamique de fonctionnement du RdP :

$$\mathcal{M}_k = \mathcal{M}_i + \mathcal{C} \times \overline{\mathcal{F}}. \quad (2.1)$$

$\overline{\mathcal{F}}$ est le vecteur caractéristique de la séquence de franchissement \mathcal{F} , c'est-à-dire un vecteur de dimension égale à $|\mathcal{T}|$; $\overline{\mathcal{F}}_q$ correspond au nombre de franchissement de la transition c_q dans la séquence \mathcal{F} . Dans la figure 2.1 où la séquence de tir était $\mathcal{F} = \{c_1\}$, le vecteur caractéristique est égal à $\overline{\mathcal{F}}^t = (1,0,0,0)$.

2.1.3 Marquages accessibles

Soit $\mathcal{N} = (\mathcal{R}; \mathcal{M}_0)$ un réseau de Petri. L'ensemble des marquages accessibles $\mathcal{A}(\mathcal{R}; \mathcal{M}_0)$ d'un réseau de Petri marqué est l'ensemble des marquages que l'on peut atteindre à partir du marquage initial \mathcal{M}_0 par une séquence de franchissement, après un temps τ , c-à-d :

$$\mathcal{A}(\mathcal{R}; \mathcal{M}_0) = \{\mathcal{M}_i, \exists c_q, \mathcal{M}_0 \xrightarrow{c_q/\tau} \mathcal{M}_i\}.$$

On peut, lorsque cet ensemble est fini, le représenter sous la forme d'un graphe. Les sommets de ce graphe correspondent aux marquages accessibles de $\mathcal{A}(\mathcal{R}; \mathcal{M}_0)$.

Un arc orienté relie deux sommets \mathcal{M}_i et \mathcal{M}_j s'il existe une transition c_q franchissable permettant de passer d'un marquage à un autre après un temps τ : $\mathcal{M}_i \xrightarrow{c_q} \mathcal{M}_j$.

La figure 2.2 représente le graphe des marquages accessibles pour le réseau de Petri de la figure 2.1 avec un marquage initial $\mathcal{M}_0 = (1,0,0,0,1)$.

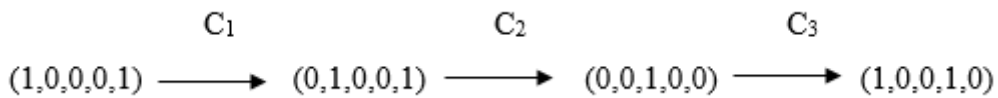


Figure 2.2 Graphe des marquages atteignables du réseau de Petri de la figure 2.1

2.1.4 Concurrence et synchronisation dans les RdP

Le formalisme des réseaux de Petri est très puissant, il permet de représenter une grande variété de comportement des systèmes réels mettant en jeu des phénomènes de concurrence et de synchronisation. Sur la figure 2.3, nous avons représenté quatre structures de réseaux de Petri permettant de décrire ces phénomènes :

- La figure 2.3 (a) représente une structure de choix. La place p_1 a deux transition en aval, notées c_1 et c_2 . Le tir de c_1 ou c_2 consommera le jeton de place p_1 et exclura donc le tir de l'autre. Une telle configuration permet de modéliser un phénomène de concurrence à la consommation comme par exemple le partage d'un processeur entre tâches concurrentes dans un système informatique ;
- La configuration de la figure 2.3(b) permet de représenter une concurrence à l'apport de jetons dans une place, ici p_1 ;
- La configuration de la figure 2.3(c) permet de modéliser un phénomène de synchronisation. La transition c_1 n'est franchissable que lorsque les places p_1 et p_2 contiennent toutes les deux au moins un jeton ;
- Dans la structure de la figure 2.3(d), l'apport de jetons dans les places p_1 et p_2 est synchronisé par le tir de la transition c_1 et permet, par exemple, de modéliser l'initiation simultanée de différentes tâches dans un système à

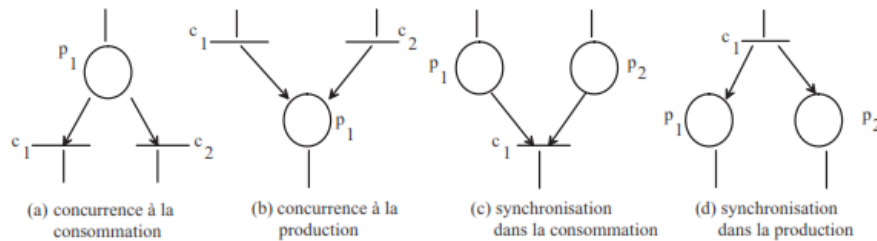


Figure 2.3 Concurrence et synchronisation dans les RdP

événements discrets.

2.1.5 Quelques propriétés des RdP

Le graphe des marquages associé à un RdP fournit des indications essentielles sur le fonctionnement du système qu'il représente. Suivant que le graphe des marquages est fini ou infini, qu'il présente ou non des circuits, il reflète certaines des propriétés caractéristiques du système modélisé.

Définition 2.2 (Bornitude). Une place $p \in \mathcal{P}$ d'un réseau de Petri marqué $(\mathcal{R} ; \mathcal{M}_0)$ est k -bornée ($k \in \mathcal{N}$) si pour tout marquage accessible $\mathcal{M} \in \mathcal{A}(\mathcal{R} ; \mathcal{M}_0)$, le marquage de cette place vérifie $\mathcal{M} \leq k$. dans le cas contraire, nous dirons que p est non-bornée. Enfin si p est 1-bornée, on dit que p est binaire. Si on considère le réseau de Petri de la figure 2.1, on voit que

pour le marquage initial \mathcal{M}_0 , les places p_1 , p_2 et p_3 sont binaires (1-bornée) et le réseau est dit sauf.

Lorsqu'un RdP modélise un système manufacturier, certaines places représentent des convoyeurs ou bien des zones de stockages intermédiaires. La bornitude du marquage du modèle est alors synonyme de limitation de la taille des stocks internes du système. La bornitude du RdP reflète alors en quelque sorte une propriété de "stabilité" du système de production modélisé.

Définition 2.3 (Vivacité). Etant donné un réseau de Petri \mathcal{R} et un marquage initial \mathcal{M}_0 , une transition c est vivante pour le réseau marqué $(\mathcal{R}; \mathcal{M}_0)$ est dit vivant si toutes ses transitions sont vivantes.

Remarque 2.1 (L'intérêt d'étude de la vivacité d'un modèle). L'évolution du marquage d'un RdP se fait par franchissement de transitions. Lorsqu'au cours de son évolution, certaines transitions ne sont jamais franchies, cela indique que l'événement associé à la transition ne se produit pas et que le marquage d'une partie du RdP n'évolue pas. Cela indique que le sous-système modélisé par cette partie-là ne fonctionnera pas. Il y a donc un problème au niveau de la conception du système.

L'idée est d'être capable de détecter systématiquement ce phénomène par l'analyse de propriétés du modèle RdP, afin de disposer d'un outil d'aide à la conception des systèmes.

2.1.6 Invariants d'un réseau de Petri

Il existe deux types d'invariants (appelés *semiflots*) dans un RdP [David 1992] : les invariants de marquage (notés également P-invariant) et les invariants de franchissement (notés également T-invariant).

Définition 2.4 Les invariants de marquage, appelés P-invariant, ou encore *P-semiflots*, illustrent la conservation du nombre de jetons dans un sous ensemble de places du RdP. Un vecteur, noté \mathcal{K} , de dimension égale au nombre de places du RdP est un P-invariant si, et seulement s'il vérifie l'équation suivante :

$$\mathcal{K}^t \times \mathcal{C} = \vec{0}, \quad \mathcal{K} \neq \vec{0}.$$

Où \mathcal{C} correspond à la matrice d'incidence du RdP.

D'après l'équation (2.1), on déduit que si \mathcal{K} est un P-invariant, alors pour un marquage donné, noté m_i , obtenu à partir d'un marquage initial m_0 , on a :

$$\mathcal{K}^t \times \mathcal{M}_i = \mathcal{K}^t \times \mathcal{M}_0, = n, \quad n \in \mathbb{N}^*.$$

Cette équation représente un invariant de marquage, elle signifie que si \mathcal{K} est un P-invariant du RdP alors la transposé du vecteur \mathcal{K} pondéré par le vecteur marquage m_i du RdP est une constante entière quel que soit le marquage m_i atteignable à partir du marquage initial m_0 . L'ensemble des places pour lesquelles la composante associée dans le P-invariant est non nulle, est appelée la composante conservative du RdP.

Définition 2.5 Un vecteur non nul d'entiers Θ de dimension $|\mathcal{T}| \times 1$ est un T-invariant, ou encore T-*semiflows*, du RdP si, et seulement s'il vérifie l'équation suivante :

$$\mathcal{C} \times \Theta = \vec{0}$$

D'après l'équation (2.1), le franchissement à partir d'un marquage m_i d'une séquence dont le vecteur caractéristique est Θ ramène le graphe au même marquage $m_k = m_i$.

Un T-invariant correspondant à une séquence de franchissement réalisable est appelé *composante répétitive*. L'ensemble des transitions pour lesquelles la composante associée dans le T-invariant est non nulle est appelé le support du T-invariant.

Définition 2.6 Un RdP est dit consistant s'il possède un T-invariant Θ couvrant toutes les transitions du réseau. Un RdP qui possède cette propriété est dit répétitif ou réinitialisable. Le réseau atteint un régime périodique dès lors qu'il existe une séquence de franchissement réalisable avec Θ , comme vecteur caractéristique.

Définition 2.7 Un RdP est dit conservatif si toutes les places du graphe forment une composante conservative.

Remarque 2.2

- Un réseau de Petri consistant est dit fortement connexe si, et seulement s'il est conservatif.
- Si toutes les places d'un RdP forment une composante conservative, le graphe est dit conservatif.
- Toute combinaison de T-invariants (resp, P-invariants) est un T-invariant (resp, P-invariant).

- Un réseau de Petri consistant à un unique T-invariant élémentaire.
- Le produit des poids d'un circuit d'un réseau de Petri conservatif est égal à 1.

Exemple 2.2 Considérons le RdP de la figure 2.4 qui suit, un P-invariant est le vecteur \mathcal{K} vérifiant :

$$\mathcal{K}^t \times \mathcal{C} = (k_1, k_2, k_3, k_4, k_5) \times \begin{pmatrix} -1 & 0 & 2 \\ 1 & -3 & 0 \\ 0 & 3 & -3 \\ 0 & -3 & 2 \\ 0 & 0 & 0 \end{pmatrix} = \vec{0}$$

Il existe un P- invariant : $\mathcal{K}^t = (0, 0, 1, 1, 1)$. L'invariant de marquage est :

$$1 \times m(p_3) + 1 \times m(p_4) + \times m(p_5) = 6.$$

Ceci signifie que les places p_1 et p_2 forment une composante conservative du graphe.

Le RdP de la figure 2.4 admet un T- invariant, $\Theta^t = (6, 2, 3)$; vérifiant :

$$\mathcal{C} \times \Theta = \begin{pmatrix} -1 & 0 & 2 \\ 1 & -3 & 0 \\ 0 & 3 & -3 \\ 0 & -3 & 2 \\ 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ 3 \end{pmatrix} = \vec{0}.$$

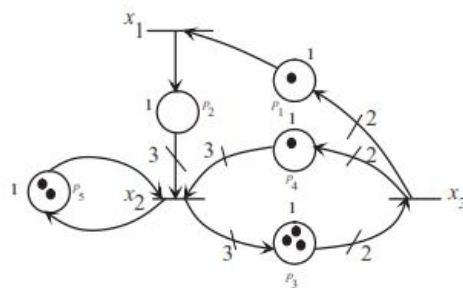


Figure 2.4 Réseau de Petri à arcs pondérés.

Définition 2.8 Un RdP est dit k-serveur si, pour chaque transition, il ne peut y avoir au maximum que k franchissements simultanés. Une boucle de réentrance implicite contenant k jetons est alors associée à chaque transition du graphe :

- Dans le cas où $k = 1$, le RdP est dit mono-serveur.

- Dans le cas où le nombre de franchissement de transitions n'est pas limité, le RdP est dit infinité -serveur.

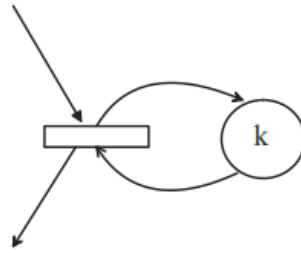


Figure 2.5 Boucle de ré-entrance.

2.2 Graphes d'Événements Temporisés

Suite à cette brève présentation succincte des réseaux de Petri, notre intérêt va maintenant se porter sur une structure particulière des réseaux de Petri qui permet de modéliser les systèmes où interviennent des phénomènes de synchronisation et de délai (les phénomènes de conflits étant supposés résolus au préalable).

Définition 2.9 Un Graphe d'Événements Temporisé (GET) est un RdP tel que toute place a exactement une transition en amont et une transition en aval.

Pour illustrer ce type de modèles temporisés, nous proposons l'exemple d'un graphe d'événements temporisé représentant le fonctionnement d'un atelier de coupe de bois.

Exemple 2.3 Le GET de la figure 2.6 modélise un atelier constitué d'une machine de coupe de bois. Quand une pièce arrive et que la machine de coupe est disponible, cette dernière est traitée (découpée).

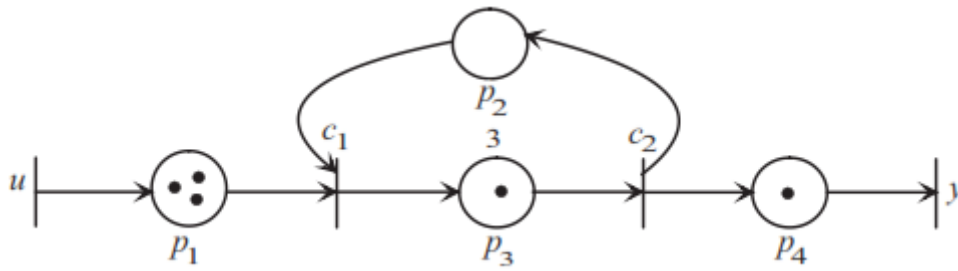


Figure 2.6 Modèle GET d'un atelier de coupe de bois

Le nombre de jetons dans une place s'interprète comme le nombre de *ressources* disponible. Par exemple, le nombre de jetons dans la place p_1 correspond au nombre de pièces en attente d'être traitées par la machine de coupe, c'est-à-dire le nombre de « ressources » qui vont être consommées. Un jeton dans la place p_3 indique qu'une pièce est en train de traitée par la machine de coupe. La durée de ce traitement est de $2 u.t$ (unités de temps). La fin de la coupe par la machine se traduit par l'enlèvement d'un jeton dans la place p_3 et par l'ajout d'un jeton dans la place p_2 et dans la place p_4 . Le nombre de jetons dans la place p_4 correspond au nombre de pièces qui ont été traitées. La présence d'un jeton dans la place p_2 indique que la machine est disponible.

2.2.1 Conditions initiales

Définition 2.10 (Conditions initiales canoniques). On dira que les conditions initiales d'un GET sont canoniques si tous les jetons du marquage initial sont considérés comme disponibles depuis l'instant $-\infty$.

Considérer des conditions initiales canoniques, implique de prendre pour hypothèse que le marquage initial du graphe n'est pas nécessairement égal au marquage à l'instant initial d'observation, (égal à 0 par convention). En effet, si les jetons initiaux sont supposés disponibles depuis $-\infty$, ils sont susceptibles d'avoir validé et tiré des transitions également à $-\infty$, aussi le marquage peut avoir évolué avant l'instant initial d'observation.

Exemple 2.4 La figure 2.7 représente un GET. Si on considère des conditions initiales canoniques, le marquage à l'instant initial d'observation est différent du marquage initial du graphe. En effet, le jeton contenu dans la place p_3 a validé et tiré la transition c_3 à l'instant $t = -\infty$. Un jeton a alors été ajouté dans les places p_4 et p_5 . Finalement, le marquage, après qu'il

ait évolué librement depuis l'instant $t = -\infty$, à l'instant d'observation est celui représenté sur la figure 2.8.

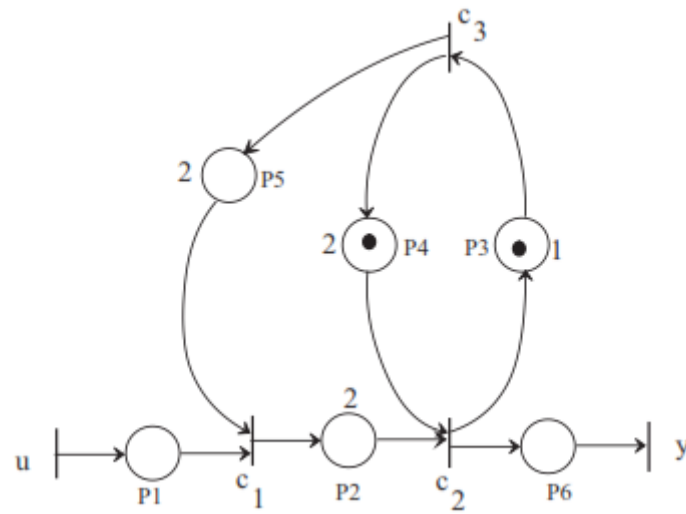


Figure 2.7 Marquage initial ($t = -\infty$).

Définition 2.11 (Conditions initiales faiblement compatibles non canoniques). On dira que les conditions initiales d'un GET sont faiblement compatibles si tout jeton du marquage initial est disponible qu'à partir de à l'instant $t = 0$. Pour $t < 0$, le graphe est considéré comme étant "gelé".

Les jetons visibles à l'instant initial d'observation $t = 0$ sont alors ceux du marquage initial. Ces jetons sont supposés avoir été introduits dans le graphe avant l'instant 0, de telle sorte qu'à l'instant $t = 0$ un jeton du marquage initial est :

- Soit en train de subir la temporisation de la place dans laquelle il a été introduit,
- Soit tout juste disponible pour le tir de la transition aval.

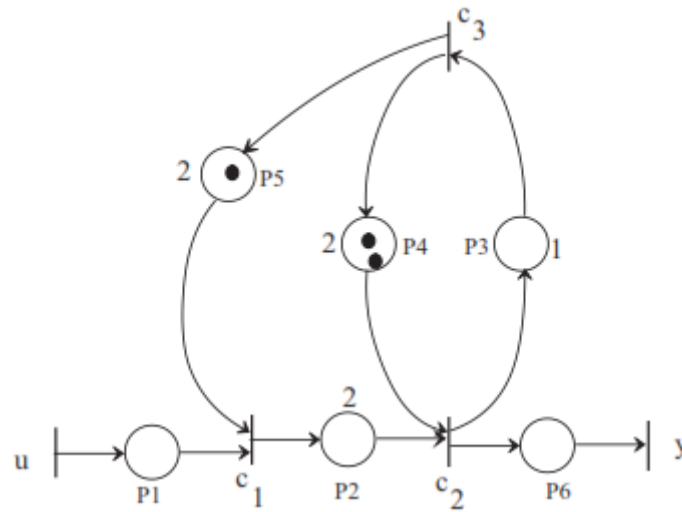


Figure 2.8 Marquage à l'instant initial d'observation ($t=0$)

2.3 Graphes d'Événements Temporisés Généralisés

De nombreux systèmes sont tels que les entités qui les traversent subissent des phases d'assemblage ou de désassemblage. La classe des Graphes d'Événements Temporisés Généralisés GETG permet une modélisation simple et intuitive de tels systèmes.

Définition 2.12 Un GETG est un GET dans lequel des poids (nombres entiers strictement positifs) sont associés aux arcs. Ces graphes sont aussi appelés graphes d'événements *avec multiplicateurs* ou *valués*.

Dans ce type de graphe, dont un exemple élémentaire est donné par la figure 2.9, le poids M_{pq} rattaché à l'arc $x_q \rightarrow p$ est tel que lors d'un franchissement de la transition x_q , M_{pq} jetons sont ajoutés à la place p , le poids M_{qp} associé à l'arc $p \rightarrow x_q$ implique que la transition x_q ne sera validée que si p contient au moins M_{qp} jetons. Lors du franchissement de cette transition, après un séjour τ_p unités de temps dans la place p , M_{qp} jetons y seront retirés.

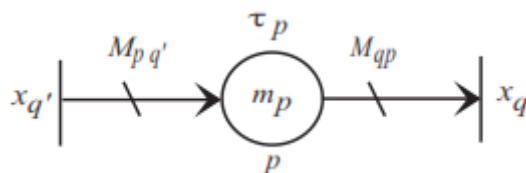


Figure 2.9 GETG élémentaire.

La figure 2.10 représente un GETG tel que l'arc $p_1 \rightarrow x_1$ a un poids de 2, et l'arc $x_1 \rightarrow p_4$ a un poids de 3. Les arcs dont le poids n'est pas explicitement spécifié ont un poids égal à 1. La transition x_1 est validée puisque la place p_2 contient un jeton. Après un séjour de deux unités de temps des jetons dans la place p_1 , le franchissement de x_1 consiste à retirer deux jetons de p_1 et un jeton de p_2 , et à ajouter un jeton à p_3 et trois jetons à p_4 . On obtient le marquage de la figure 2.10.b.

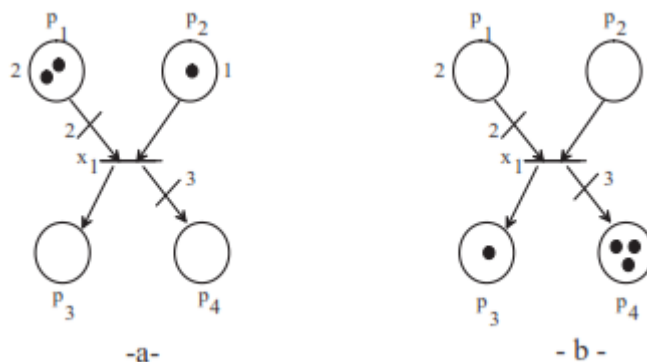


Figure 2.10 Evolution du marquage

2.3.1 Gain d'un circuit :

Définition 2.13 (Gain d'un circuit [Teruel 1992]). Le gain d'un circuit, noté $G(\rho)$, est le produit du quotient des poids des arcs entrants par le produit des poids des arcs sortants de toutes les places du circuit ρ . Formellement, ce gain est défini comme suit :

$$G(\rho) = \prod_{p_i \in \rho} \frac{a_i}{b_i}$$

Où :

- a_i est égal au poids de l'arc entrant dans la place p_i ,
- b_i est égal au poids de l'arc sortant dans la place p_i ,

Selon la valeur de son gain, un circuit ρ est classé dans l'une des trois catégories suivantes :

- $G(\rho) > 1$: circuit générateur,

- $G(\varrho) = 1$: circuit neutre,
- $G(\varrho) < 1$: circuit absorbant.

La valeur de $G(\varrho)$ informe sur la conservation des marques dans le circuit ϱ . Si un circuit est générateur, alors le nombre de marques dans le circuit augmente avec l'évolution du marquage. Si le circuit est neutre, ce gain reste constant au cours de l'évolution du marquage. Enfin, si le circuit est absorbant, le nombre de marque diminue.

Dans le cas des GET ordinaires, le nombre total de jetons dans un circuit est constant. De par la *valuation* des arcs dans \mathbb{N}^* , ceci n'est plus vrai dans le cas des GETG. Chao et al. [Chao 1993] ont utilisé un principe similaire en introduisant la notion de marques *valuées* des places d'un circuit. La marque *valuée*, notée \mathcal{N}_p , d'une place p d'un circuit peut se calculer comme suit :

$$\mathcal{N}_p = \frac{m_p}{M_{qp}\Theta_q}$$

Où :

- m_p correspond au marquage initial de la place p ,
- M_{qp} correspond au poids de l'arc sortant de la place p ,
- Θ_q correspond à la composant de T-invariant associé à la transition en aval de la place p .

La marque *valuée totale* d'un circuit neutre ϱ , est donnée par la somme des marques valuées des places de ce circuit. Elle ne dépend pas du temps et est constante quel que soit l'évolution du circuit. Par conséquent, elle est généralement calculé pour le marquage initial M_0 et est donnée par :

$$\mathcal{N}(\varrho) = \sum_{p \in \varrho} \mathcal{N}_p$$

L'étude de la vivacité des GETG est difficile. Des résultats ont été obtenus dans ce domaine par D.T. Chao [Chao 1993] : un circuit d'un GETG est vivant s'il contient initialement au moins une place marquée dont le marquage value est supérieur à un ($\mathcal{N}(\varrho) \geq 1$).

Contrairement aux GET ordinaires, il n'existe pas de condition nécessaire et suffisante pour vérifier la vivacité des GETG.

3.1 Conclusion :

Dans ce chapitre nous avons présenté le langage graphique (les réseaux de petri) qui permettent de décrire les systèmes à événements discrets (SED). Qu'on a besoins pour modéliser notre ligne d'embouteillage d'eau par la suit dans le chapitre suivant.

Chapitre III

Modélisation des GETG dans l'algèbre des dioïdes

3.1 Modélisation des GET dans l'algèbre (min, +)

Dans le cas des Rdp, l'état modélisant l'évolution du modèle est associé aux marquages des places. Si l'on se réfère à la figure 2.6, la seule caractéristique que traduit une telle représentation concerne l'état de la machine M (au sens logique) : machine libre ou occupée. Pour pouvoir discuter des performances du système temporisé, c'est-à-dire déterminer son régime transitoire et/ou son régime permanent, l'équipe (max, +) de l'INRIA a proposé dans [Cohen 1985] une modélisation considérant non pas l'état du marquage mais les compteurs et les dates de franchissements des transitions.

Pour étudier les GET dans l'algèbre (min, +), Les variables d'état considérées sont de type compteurs d'événements. De telles variables permettent d'aboutir à une représentation linéaire de ces modèles dans les dioïdes. On a privilégié la représentation compteur à celle de dateur (*cf.* remarque 3.2) du fait des dioïdes inspirés de l'algèbre (min, +) qui seront utilisés par la suite pour représenter des Graphes d'Événements Temporisés Généralisés.

Définition 3.1 Un compteur associé à une transition c_j est une application croissante, notée c_j de $\mathbb{Z} \rightarrow \mathbb{Z} \cup \{\pm\infty\}$, $t \rightarrow c_j(t)$, où $c_j(t) \in \mathbb{Z} \cup \{\pm\infty\}$ correspond au nombre de tirs de la transition q_j ayant lieu jusqu'à la date t .

Le vecteur d'état pour un GET, est défini par :

$$C(t) = (c_1(t) \dots c_{|\mathcal{T}|}(t))^t$$

Contrairement aux marquages, ces compteurs ont une propriété de monotonie :

$$\forall t \in \mathbb{R}, \forall \beta \geq 0, \forall c_j \in \mathcal{T}, c_j(t + \beta) \geq c_j(t).$$

Remarque 3.1 Comme le franchissement des transitions est ici supposé être instantané, il n'y a pas à formuler d'hypothèses sur les tirs des transitions, leur fonctionnement est évidemment de type FIFO (First-In, First-Out).

Pour illustrer l'évolution d'un compteur associé à la transition c_j d'un GET considérons le GET élémentaire suivant :

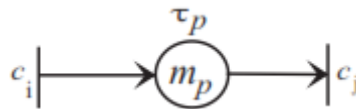


Figure 3.1 GET élémentaire

Sous l'hypothèse de fonctionnement au plus tôt, qui est un comportement limite du GET, l'évolution du compteur $c_j(t)$, associé à la transition c_j du GET élémentaire de la figure 3.1, est modélisé par l'équation suivant :

$$c_j(t) = m_p + c_i(t - \tau_p).$$

Dans un GET, le phénomène de synchronisation se produit lorsque plusieurs arcs

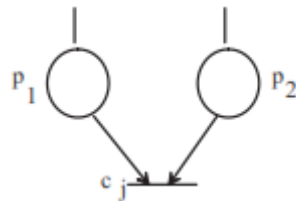


Figure 3.2 Phénomène de synchronisation dans un GET.

convergent sur une même transition (cf. figure 3.2).

En général, l'évolution d'un compteur associé à une transition de synchronisation d'un GET est donnée par l'équation suivante :

$$C_j(t) = \min_{p \in \circ_j, i \in \circ_p} (m_p + c_i(t - \tau_p)).$$

On note $\bullet q$ (resp, $q \bullet$) l'ensemble des places situées immédiatement en amont (resp aval) de la transition c_q . Similairement, $\bullet p$ (resp, $p \bullet$) représente l'ensemble des transitions situées immédiatement en amont (resp, en aval) de la place p .

On remarque que cette équation est non linéaire dans l'algèbre usuelle de par la présence de l'opérateur min qui modélise le phénomène de synchronisation au niveau de la transition c_j . Par contre, elle se décrit de façon linéaire dans le dioïde $\overline{\mathbb{Z}}_{min}$. Le compteur $c_j(t)$ vérifie l'équation récurrente suivante :

$$C_j(t) = \bigoplus_{p \in \circ_j, i \in \circ_p} (m_p \otimes c_i(t - \tau_p)).$$

L'opération \bigoplus du dioïde $\overline{\mathbb{Z}}_{min}$ représente un phénomène de synchronisation au niveau de transition c_j , et l'opération \otimes représente un décalage *événementiel* du compteur $c_j(t)$ par rapport à tout compteur associé à une transition en amont de la transition c_j .

D'une manière générale, les équations modélisant le fonctionnement au plus tôt des GET ont la forme suivante dans le dioïde $\overline{\mathbb{Z}}_{min}$.

$$\begin{cases} C(t) = \bigoplus_{a=0}^T A_a C(t-a) \bigoplus \bigoplus_{a=0}^T B_a U(t-a), \\ Y(t) = \bigoplus_{a=0}^T C_a C(t-a), \end{cases} \quad (3.1)$$

Avec $T = \max_{p_a \in \mathcal{P}} (\tau(p_a))$. $C(t)$ est le vecteur d'état, composé des compteurs associés aux transitions internes du graphe. $U(t)$ et $Y(t)$ sont respectivement les vecteurs d'entrées (commandes) et de sorties, composés des transitions sources, respectivement puits, du graphe.

Exemple 3.1 Dans le dioïde $\overline{\mathbb{Z}}_{min}$, l'évolution des compteurs du GET de la figure 3.3 sont régies par les relations suivantes :

$$\begin{cases} c_1(t) = 1 \otimes c_1(t-1) \bigoplus 2 \otimes c_2(t) \bigoplus u(t), \\ c_2(t) = c_1(t-2) \\ y(t) = c_2(t) \end{cases}$$

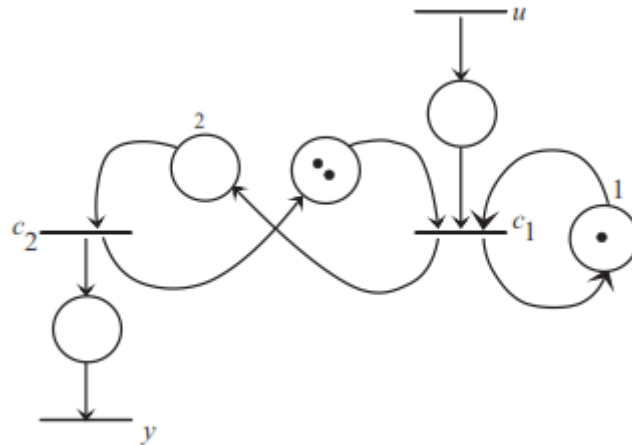


Figure 3.3 Exemple d'un GET.

Il en résulte la représentation d'état suivante :

$$\begin{cases} C(t) = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix} C(t) \oplus \begin{pmatrix} 1 & \varepsilon \\ \varepsilon & \varepsilon \end{pmatrix} C(t-1) \oplus \begin{pmatrix} \varepsilon & \varepsilon \\ e & \varepsilon \end{pmatrix} C(t-2) \oplus \begin{pmatrix} e \\ \varepsilon \end{pmatrix} U(t), \\ Y(t) = (\varepsilon \quad e)C(t), \end{cases}$$

Avec $C=(c_1 \quad c_2)t$, $U=u$ et $Y=y$.

3.1.1 Forme récurrente explicite

La représentation d'état sous une forme ARMA (Auto Regressive Moving Average) donnée par la représentation (3.1) peut se ramener, moyennant une extension de la dimension du vecteur d'état, à une forme récurrente markovienne du type :

$$\begin{cases} C(t) = AC(t-1) \oplus BU(t), \\ Y(t) = CC(t). \end{cases} \quad (3.2)$$

Le GET équivalent auquel va correspondre une représentation de la forme (3.2) est tel que :

- La temporisation d'une place située entre deux transitions internes doit être égale à un ;
- La temporisation d'une place située entre une transition source et une transition interne, ou entre une transition interne et une transition puits, doit être nulle.

Exemple 3.2 La figure 3.4 représente un GET dont le comportement entrée-sortie est équivalent à celui de la figure 3.3. En introduisant un nouvel état c'_1 , on aboutit à la représentation d'état suivante sur le dioïde $\overline{\mathbb{Z}}_{min}$:

$$\begin{cases} C(t) = \begin{pmatrix} \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} C(t) \oplus \begin{pmatrix} 1 & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{pmatrix} C(t-1) \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} U(t), \\ Y(t) = (\varepsilon \quad \varepsilon \quad e)C(t). \end{cases}$$

Avec $C=(c_1 \quad c'_1 \quad c_2)t$, $U=u$ et $Y=y$.

On obtient, après utilisation du théorème 1.3 pour supprimer la partie implicite de l'équation d'état, la forme récurrente markovienne suivante :

$$\begin{cases} C(t) = \begin{pmatrix} 1 & 2 & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{pmatrix} C(t-1) \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} U(t), \\ Y(t) = (\varepsilon \quad \varepsilon \quad e)C(t). \end{cases}$$

Remarque 3.2 (Equations aux dateurs). Au lieu de considérer les nombres cumulés de tirs d'une transition d'un GET à chaque instant, on s'intéresse aux différentes dates de tirs de cette transition. A chaque transition c_i , on associe l'application $d_i : \mathbb{Z} \rightarrow \overline{\mathbb{Z}}_{max}$, $k \rightarrow i(k)$ où $d_i(k)$ désigne la date du k -ième tir de la transition d_i . Cette application est non décroissante et est appelée dateur.

Par un raisonnement analogue à celui effectué pour les compteurs, on montre que la description du comportement d'un GET peut se représenter *via* des équations (max, +) linéaires.

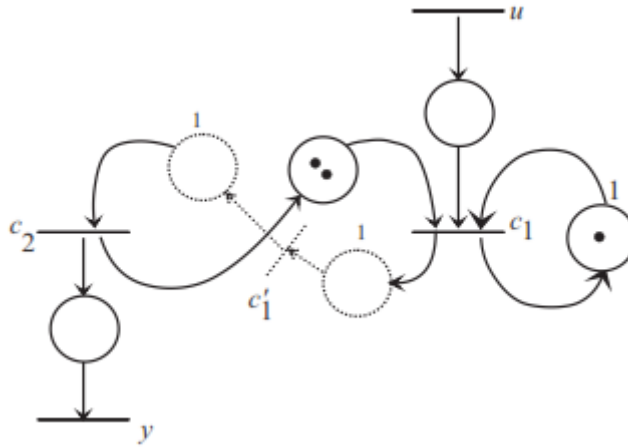


Figure 3.4 GET équivalent à celui de la figure 3.3 en vue d'une représentation markovienne.

Le graphe de la figure 3.4 peut être modélisé par les équations $(\max, +)$ linéaires suivantes :

$$\begin{cases} d_1(k) = 1 \otimes d_1(k-1) \oplus d_2(k-2) \oplus u(k), \\ d_2(k) = 2 \otimes d_1(k), \\ y(k) = d_2(k). \end{cases}$$

3.2 Modélisation des GETG

3.2.1 Equations récurrentes des GETG

Comme dans le cas des GET ordinaires, on se propose d'étudier les GETG à travers leurs représentations d'état de type compteur.

Assertion 1 L'évolution du compteur associé à la transition x_q du GETG élémentaire de la figure 2.9 est donnée par l'équation suivante :

$$x_q(t) = \lfloor M_{qp}^{-1}(m_p + M_{pq'}x_{q'}(t - \tau_p)) \rfloor \quad (3.3)$$

La présence de la partie entière inférieure dans l'équation (3.3) assure la valeur discrète du compteur indépendamment de la valeur M_{qp} . En général, une transition x_q peut avoir plusieurs transitions en amont $\{x_{q'} \in \bullet\bullet q\}$, ce qui implique que l'équation compteur associée fasse intervenir le min entre les compteurs associés aux transitions en amont.

Exemple 3.3 La variable compteur associée à la transition x_q du GETG représenté par la figure 3.5 satisfait l'équation suivante :

$$x_q(t) = \left\lfloor \frac{2 + 3x_{q'}(t-1)}{2} \right\rfloor$$



Figure 3.5 Exemple de GETG élémentaire.

Exemple 3.4 L'évolution des compteurs associés aux transitions du GETG représenté par la figure 3.6, est décrite par les équations récurrentes suivantes :

$$\begin{cases} x_1(t) = \min(3 + x_3(t-2), u(t)), \\ x_2(t) = \min\left(\left\lfloor \frac{2x_1(t-2)}{3} \right\rfloor, \left\lfloor \frac{3 + 2x_3(t-2)}{3} \right\rfloor\right), \\ x_3(t) = 1 + 3x_3(t-1), \\ y(t) = x_2(t). \end{cases}$$

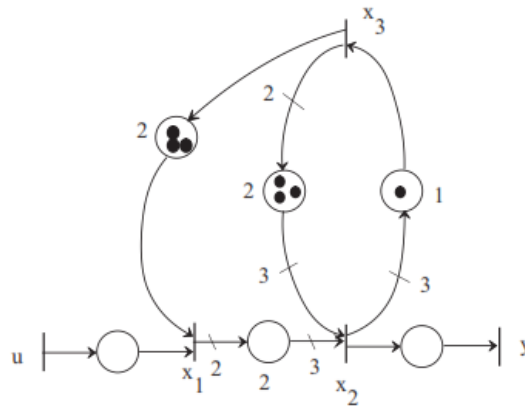


Figure 3.6 Exemple de GETG

3.2.2 Modélisation à base d'opérateurs des GETG

Le modèle mathématique représentant l'évolution de la dynamique des GETG n'admet pas une représentation linéaire dans l'algèbre $(\min, +)$. Cette non linéarité est due aux parties entières engendrées par la présence des poids sur les arcs. De ce fait, il s'avère difficile d'utiliser l'algèbre $(\min, +)$ pour aborder, par exemple, le problème de commande

d'évaluation de performances des GETG. Pour pallier à ce problème de non linéarité, nous utilisons un autre type de modélisation qui se base sur un dioïde d'opérateurs développé par Guy Cohen dans [Cohen 1998a].

Trois types d'opérateur définis de $\overline{\mathbb{Z}^Z}$ vers $\overline{\mathbb{Z}^Z}$ sont utilisés pour modéliser les GETG.

- Opérateur de stock γ^v : Il représente un décalage événementiel de v événement ($v \in \overline{\mathbb{Z}}$) et modélise un système gain élémentaire.

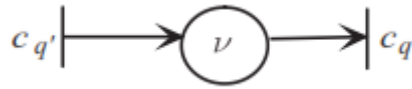


Figure 3.7 Opérateur « γ^v »

Il est défini comme suit :

$$\forall t \in \mathbb{Z}, \forall c_{q'} \in \overline{\mathbb{Z}^Z}, c_q(t) = \gamma^v c_{q'}(t) = c_{q'}(t) + v$$

L'opérateur γ^v satisfait les règles suivantes :

- $(\gamma^v \oplus \gamma^{v'})c_q(t) = \gamma^{\min(v, v')}c_q(t)$.
- $(\gamma^v \otimes \gamma^{v'})c_q(t) = \gamma^{v+v'}c_q(t)$.

En effet, on a

$$(\gamma^v \oplus \gamma^{v'})c_q(t) = \min(c_q(t) + v, c_q(t) + v') = c_q(t) + \min(v, v') = \gamma^{\min(v, v')}c_q(t)$$

$$(\gamma^v \otimes \gamma^{v'})c_q(t) = \gamma^v(c_q(t) + v') = c_q(t) + v + v' = \gamma^{v+v'}c_q(t)$$

- Opérateur de retard δ^τ : Il représente un retard temporel de τ unité de temps ($\tau \in \overline{\mathbb{Z}}$), et modélise un système retard élémentaire.

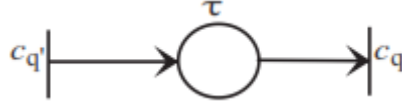


Figure 3.8 Opérateur « δ^τ »

Il est défini comme suite :

$$\forall t \in \mathbb{Z}, \forall c_{q'} \in \overline{\mathbb{Z}}, c_q(t) = \delta^\tau c_{q'}(t) = c_{q'}(t - \tau)$$

L'opérateur δ^τ satisfait les règles suivantes :

- $(\delta^\tau \oplus \delta^{\tau'}) c_q(t) = \delta^{\max(\tau, \tau')} c_q(t)$.
- $(\delta^\tau \otimes \delta^{\tau'}) c_q(t) = \delta^{\tau + \tau'} c_q(t)$.

Du fait que le signal $c_q(t)$ soit non décroissant, on a :

$$(\delta^\tau \oplus \delta^{\tau'}) c_q(t) = \min(c_q(t - \tau), c_q(t - \tau')) = c_q(t - \max(\tau, \tau')) = \delta^{\max(\tau, \tau')} c_q(t).$$

$$(\delta^\tau \otimes \delta^{\tau'}) c_q(t) = \delta^\tau c_q(t - \tau') = c_q(t - \tau' - \tau) = \delta^{\tau + \tau'} c_q(t).$$

- Opérateur multiplicateur μ_m : Il représente un facteur de mise à l'échelle m .



Figure 3.9 Opérateur μ_m , avec $m = \frac{a}{b}$

Il est défini comme suit :

$$\forall t \in \mathbb{Z}, \forall c_{q'} \in \overline{\mathbb{Z}}, c_q(t) = \mu_m c_{q'}(t) = \lfloor m \times c_{q'}(t) \rfloor,$$

Avec $m \in \mathbb{Q}^+$ (« m » est égal au quotient de deux nombres naturels strictement positifs).

Remarque 3.3 contrairement aux opérateurs γ^V et δ^τ , l'opérateur μ_m n'admet pas de règles de simplification pour l'opération \otimes :

$$\forall m, m' \in \mathbb{Q}^+, \forall c_q \in \overline{\mathbb{Z}}, (\mu_m \otimes \mu_{m'}) c_q(t) \text{ peut être différent de } (\mu_{m \times m'}) c_q(t).$$

L'opérateur μ_m satisfait les règles suivante lorsqu'il composé avec les opérateurs γ^V et δ^τ :

- $(\mu_m \otimes \gamma^V)c(t) = (\gamma^{V \times m} \otimes \mu_m)c(t)$, pour $v \in \mathbb{N}$.
- $(\mu_m \otimes \delta^\tau)c(t) = (\delta^\tau \otimes \mu_m)c(t)$

En effet, on a :

$$\forall v \in m^{-1} \times \mathbb{N}, (\mu_m \otimes \gamma^V)c(t) = [m \times v + m \times c(t)] = m \times v + [m \times c(t)] = (\gamma^{v \times m} \otimes \mu_m)c(t), \text{ puisque } v \times m \in \mathbb{N}.$$

$$(\mu_m \otimes \delta^\tau)c(t) = [m \times c(t - \tau)] = (\delta^\tau \otimes \mu_m)c(t)$$

Si les opérateurs, γ^V et δ^τ , habituellement utilisé dans le cas des GET ordinaires, sont linéaires, l'opérateur μ_m est, par contre, seulement additif, en effet, on a :

$$\forall t \in \mathbb{Z}, \forall c_1, c_2 \in \overline{\mathbb{Z}},$$

$$\mu_m(c_1(t) \oplus c_2(t)) = [m \times (\min(c_1(t), c_2(t)))] = \min([m \times c_1(t)], [m \times c_2(t)]) = \mu_m c_1(t) \oplus \mu_m c_2(t).$$

La non linéarité de l'opérateur μ_m est due à sa non homogénéité, en effet : $\forall \lambda \in \mathbb{N}$,

$$\lambda \otimes (\mu_m c_1(t)) = \lambda + [m \times c_1(t)]$$

Est différent de

$$\mu_m(\lambda \otimes c_1(t)) = [m \times (\lambda + c_1(t))]$$

On note $\mathcal{D}_{\min}[\delta]$ le dioïde d'opérateur où la loi additive \oplus correspond à l'opération *min*, et la loi multiplicative \otimes est équivalente à la loi composition usuelle. Ce dioïde a pou élément neutres $\varepsilon = \mu_{+\infty} \gamma^{+\infty} \delta^{-\infty}$ et $e = \mu_1 \gamma^0 \delta^0$. Pour des détails [Hamaci et al, 2004].

Un élément de $\mathcal{D}_{\min}[\delta]$ permet de coder un signal $c(t)$ sous forme d'une série formelle en δ , et il permet également de modéliser la relation entrée-sortie liant deux transitions d'un GETG.

Exemple 3.5 Dans le dioïde $\mathcal{D}_{\min}[\delta]$, le compteur $x_q(t)$, associé à la transition x_q du GETG élémentaire de la figure 2.9, satisfait l'équation suivante :

$$X_q(\delta) = \mu_{M_{qp}^{-1}} \gamma^{m_p} \delta^{\tau_p} \mu_{M_{pq}} X_{q'}(\delta)$$

Pour $X_{q'}$ correspondant à une impulsion, notée $E(\delta)$, et égal à $e = \mu_1 \gamma^0 \delta^0$, on obtient la relation liant la transition x_q et $x_{q'}$ sous forme d'une composition d'opérateurs donnée par : $\mu_m \gamma^v \delta^\tau$.

Assertion 2 L'évolution de la dynamique d'un GETG est donnée par la représentation d'état suivante :

$$\begin{cases} X(\delta) = AX(\delta) \oplus BU(\delta), \\ Y(\delta) = CX(\delta) \oplus DU(\delta). \end{cases} \quad (3.4)$$

Où $X(\delta)$ est le vecteur d'état composé des compteurs associés aux transitions internes du graphe, codé dans le dioïde $\mathcal{D}_{\min}[\delta]$. $U(\delta)$ et $Y(\delta)$ sont les vecteurs d'entrées et de sorties codés dans $\mathcal{D}_{\min}[\delta]$.

Les matrices A , B , C et D sont composées des monômes de formelles en δ , chaque monôme modélise une relation liant deux transitions du graphe dans $\mathcal{D}_{\min}[\delta]$.

Exemple 3.6 Dans le dioïde $\mathcal{D}_{\min}[\delta]$, le GETG de la figure 3.6 admet la représentation d'état suivante :

$$\begin{cases} \begin{pmatrix} x_1(\delta) \\ x_2(\delta) \\ x_3(\delta) \end{pmatrix} = \begin{pmatrix} \varepsilon & \varepsilon & \gamma^3 \delta^2 \\ \mu_1 \delta^2 \mu_2 & \varepsilon & \mu_1 \gamma^3 \delta^2 \mu_2 \\ \varepsilon & \gamma^1 \delta^1 \mu_3 & \varepsilon \end{pmatrix} \begin{pmatrix} x_1(\delta) \\ x_2(\delta) \\ x_3(\delta) \end{pmatrix} \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} U(\delta), \\ Y(\delta) = (\varepsilon \quad e \quad \varepsilon) \begin{pmatrix} x_1(\delta) \\ x_2(\delta) \\ x_3(\delta) \end{pmatrix}. \end{cases} \quad (3.5)$$

3.3. Modélisation de la chaîne de production de beurre :

Les systèmes considérés, sont ceux modélisables par les GETG. Un système réel sera modélisé dans ce chapitre, il s'agit bien d'une chaîne de conditionnement de beurre.

3.3.1 Description du processus de fabrication

Le flux de production sont illustré sur la figure 3.10 suivante :

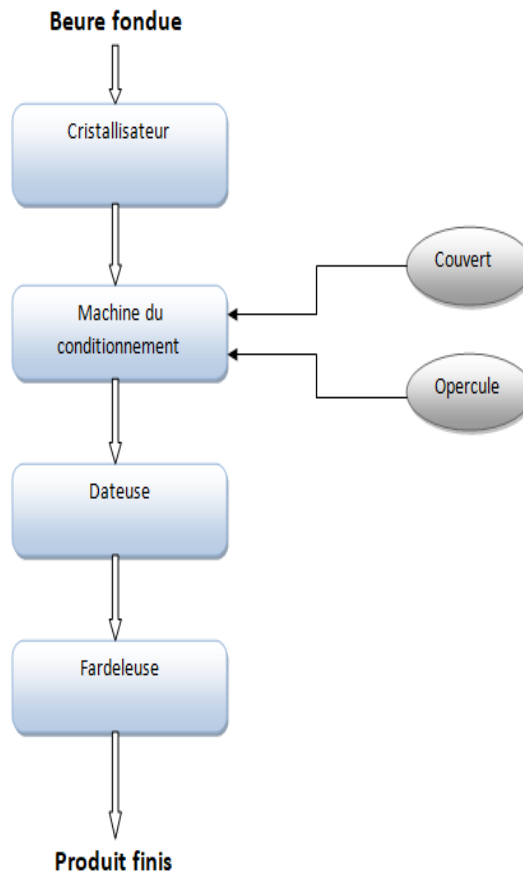


Figure 3.10 - Description du processus de fabrication de beurre.

- **Poste 1 :** Le cristallisateur (capacité 15kg, 120/130s), ce dernier est composé de deux procédés :
 - Le premier est l'échangeur principale de température son rôle est des baisser la température du produit pour passer au remplissage,
 - le deuxième procédé est le mélangeur ou le mixeur son rôle est de mixer la beure en utilisant un agitateur afin d'éliminer les grains du beur non écraser
- **Poste 2 :** La machine du conditionnement (3boite/7s),
 - *Le remplissage des boite :* la beure arrivé du poste 1 elle rentre dans un doseur principale de 3kg pour distribue le produit sur 3 doseurs secondaires de 500g/dose, la machine remplit 3 boite à la fois avec une durée de 3s.

- *La fermeture des boites* : Ce procédé passe sur 4 étapes de fermeture avec une durée total de 4s et 3 boite simultanément,
- *Positionnement de l'opercule* (aluminium) avec un vérin 1s,
- *Soudure* de l'opercule avec une résistance chauffante d'une durée de 1s,
- *Positionnement* du couvert sa durée une 1s,
- *Fixation* du couvert sa durée une 1s,
- **Poste 3** : Dateuse (1boite/1s)

Ce poste recevant les boites de poste 2 a laide d'un convoyeur dans le quel un système dateur jet l'encre dis qu'un capteur envoyé un signalé de présente d'une boite dans la bonne position pour écrire la date récente de fabrication sur la boite.

- **Poste 4** : Fardeleuse (24boite dans chaque carton//8-10s)

A la sortie du poste précédent les boites passent sur une balance électronique pour trier les boites qui on le poids de 500g avant d'être mise en carton. Les carton est de capacité 12kg soit 24boites /carton.

3.3.2 Modélisation graphique :

Dans cette partie, en basant sur cahier des charges décrivant, pour une gamme de fabrication, le fonctionnement global de la chaîne de production, on a établi le modèle réseau de Pétri qui modélise le fonctionnement de cette chaîne.

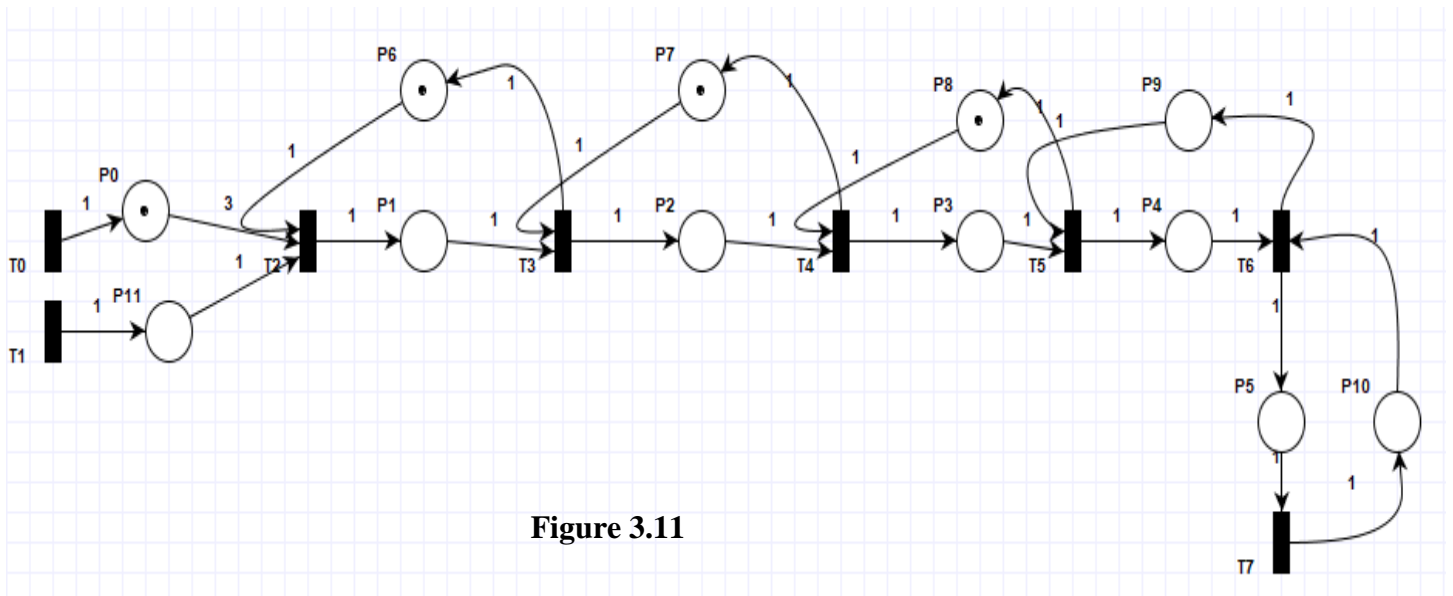


Figure 3.11 – Modèle proposé réseau de Pétri de la chaîne de production de beurre.

3.3.3 Modélisation mathématique

Cette modélisation est donnée par les équations récurrentes suivantes :

$$\left\{ \begin{array}{l} N1(t) = 3 + U1(t - 120) \\ N2(t) = 3N1(t - 3) \\ N3(t) = 3 + 3N2(t - 2) \\ N4(t) = 3N3(t - 2) \\ N5(t) = N4(t - 5) \\ N6(t) = 24 + N5(t - 5) \\ y(t) = N6(t) \end{array} \right. \quad (3.6)$$

Le modèle mathématique (3.6) est représenté les équation récurrente de système modélisé indique la dynamique de l'unité de fabrication en fonction de temps.

3.4 Conclusion

Ce modèle mathématique a une représentation linéaire dans l'algèbre $(\min, +)$, pour palier à ce problème de non linéarité et pour pouvoir appliquer certains résultats développés dans le cadre de l'analyse de performances des systèmes linéaire dans l'algèbre des dioïdes et qui ne sera pas traité dans ce mémoire, une approche de modélisation de ces graphes a été utilisée. Cette approche est permet de décrire la dynamique de modèle et les relations reliant les différentes transitions du graphe. Le modèle d'équations récurrentes sera utilisé par la suite pour évaluer les performances des GETG et sera simulé dans un environnement de logiciel réseau de Pétri PIPE2.

Chapitre IV

Simulation de la chaîne de production de beure dans PIPE2

4. 1. Introduction :

Les Réseaux de Petri sont un outil de modélisation universellement connu et reconnu pour les possibilités d'analyse, de validation et de vérification dont ils font preuve. L'exploitation de la théorie associée aux RdP permet, par la recherche des P et T invariants de répondre à de nombreux problèmes. La diversité des configurations des systèmes de production et la demande de flexibilité de ces systèmes en temps réel, nous amènent à considérer l'évolutivité possible des modèles en vue de la simulation et de la commande éventuelle.

Cet outil de modélisation est particulièrement bien adapté à la représentation des systèmes de production. Il se prête en effet à une construction modulaire et permet, dans la globalité du modèle obtenu, de repérer tous les sous-ensembles.

Avec cet outil de modélisation, on peut proposer également une simulation des fonctionnements. Ainsi, nous retrouvons cette dualité entre le système et son modèle, où l'élaboration du système se fait conjointement à l'observation préventive des comportements à travers le modèle. En toute logique, on doit concevoir alors le système par une série d'aller-retour entre le système en construction et son modèle en simulation.

Les Réseaux de Petri peuvent en outre apporter une contribution dans le domaine de la flexibilité : ainsi, comme nous allons le voir, les modèles sont tout à fait aptes à évoluer en fonction des événements et des contextes et ainsi répondre en temps réel aux contraintes imposées en termes de flexibilité. Dans ce chapitre nous nous sommes intéressés à la simulation et l'analyse des résultats dans un environnement PIPE2 (PIPE2 est un logiciel de simulations des systèmes de productions modélisés par les réseaux de Petri). Le modèle exploité dans la simulation est le modèle de la chaîne de production de beure démontré dans le chapitre précédent.

4. 2. Définition du logiciel de simulation (PIPE 2) :

Le formalisme de modélisation du Réseau de Pétri permet une visualisation graphique pratique des modèles de système, comme ainsi que l'analyse des propriétés d'exactitude et de

performance. La théorie des Réseaux de Pétri a été largement utilisée pour mettre en œuvre une variété d'outils de modélisation et d'évaluation. Cet article passe en revue divers de ces outils et décrit certaines extensions récentes d'un outil de Réseau de Pétri gratuit et open-source indépendant de la plate-forme appelé PIPE.

Les extensions comprennent : une augmentation de la puissance de modélisation grâce à l'introduction d'arcs inhibiteurs, un nouveau module d'analyse pour générer des siphons et des pièges, quelques nouvelles fonctionnalités d'interface et diverses présentations améliorées. Nous illustrerons la modélisation et l'analyse du système avec des arcs inhibiteurs dans une étude de cas d'un jeton réseau en anneau.

4. 3. Simulation :

4.3.1. Classification :

La classification est basé sur la connectivité entre les lieux et les transitions, ce module classe un réseau Pétri en un ou plusieurs types suivants : State Machine, Marked Graph, FC-Nets, EFC-Nets, Réseaux SPL, Réseaux ESPL.

Nous allons utiliser l'outil «Pétri net classification results» pour afficher la classification pour ce faire on va double cliquer sur cet outil :

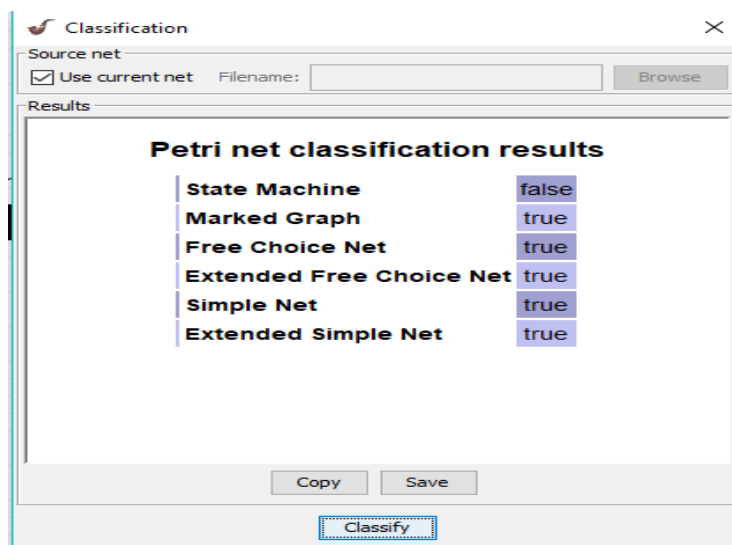


Figure 4.1- l'outil «Pétri net classification results» .

4.3.2. Les résultats d'analyse des invariants dans les Réseaux de Pétri :

Ce module calcule avec précision les vecteurs invariant de lieu et invariant de transition et efficacement. Ces vecteurs (appelés net-invariants) ne sont pas difficiles à calculer puisque la complexité de le calcul ne dépend que du nombre de places et de transitions dans le réseau et non de la taille de l'ensemble d'accessibilité. Il fournit également les équations de marquage et des informations sur la délimitation et vivacité.

Nous allons utiliser l'outil «Pétri net invariant analysis results» pour afficher les résultats des invariants analyses pour ce faire on va double cliquer sur cet outil :

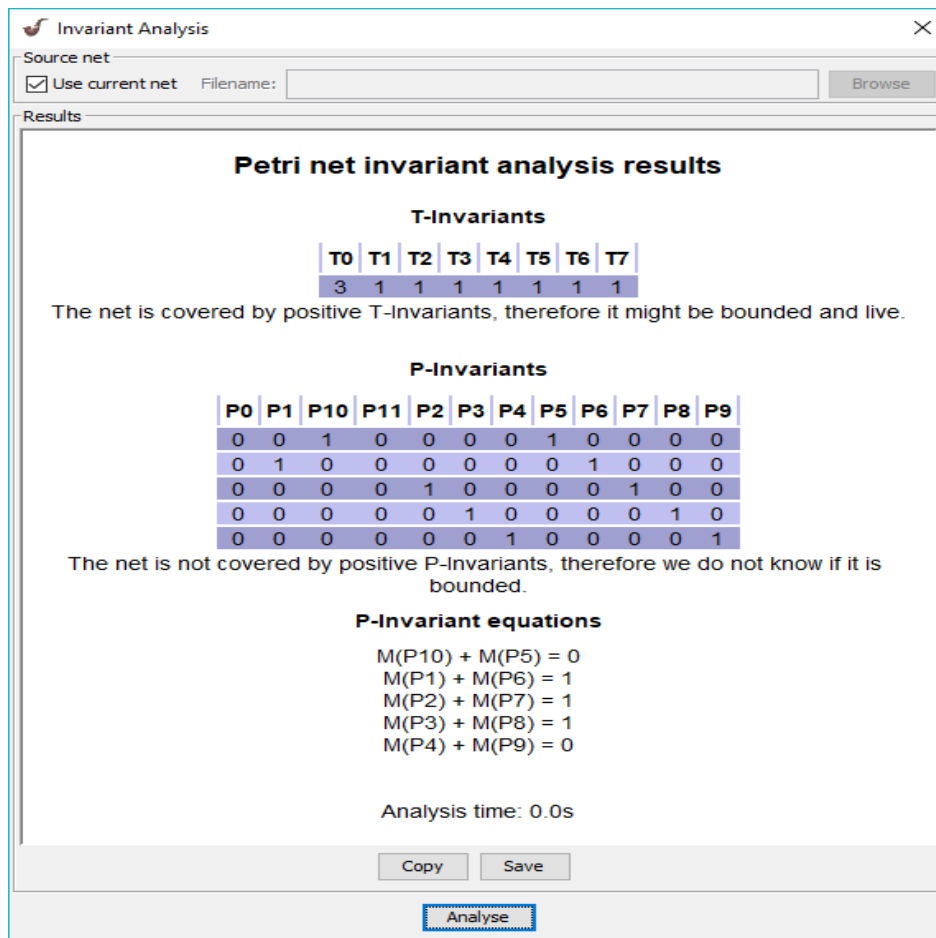


Figure 4.2 - l'outils «Pétri net invariant analysis results».

4.3.3. Incidence et marquage du Réseaux de Pétri:

Ce module affiche les matrices d'incidence avant, arrière et combine et la matrice de marquage et l'ensemble des transitions activées. Il permet à l'utilisateur de se sentir plus en confiance sur la validité du Réseau de Pétri car les résultats intermédiaires utilisés pour générer les plus avances les résultats peuvent être consultés.

Nous allons utiliser l'outil «Pétri net incidence and marking», pour ce faire on va double cliquer sur cet outil :

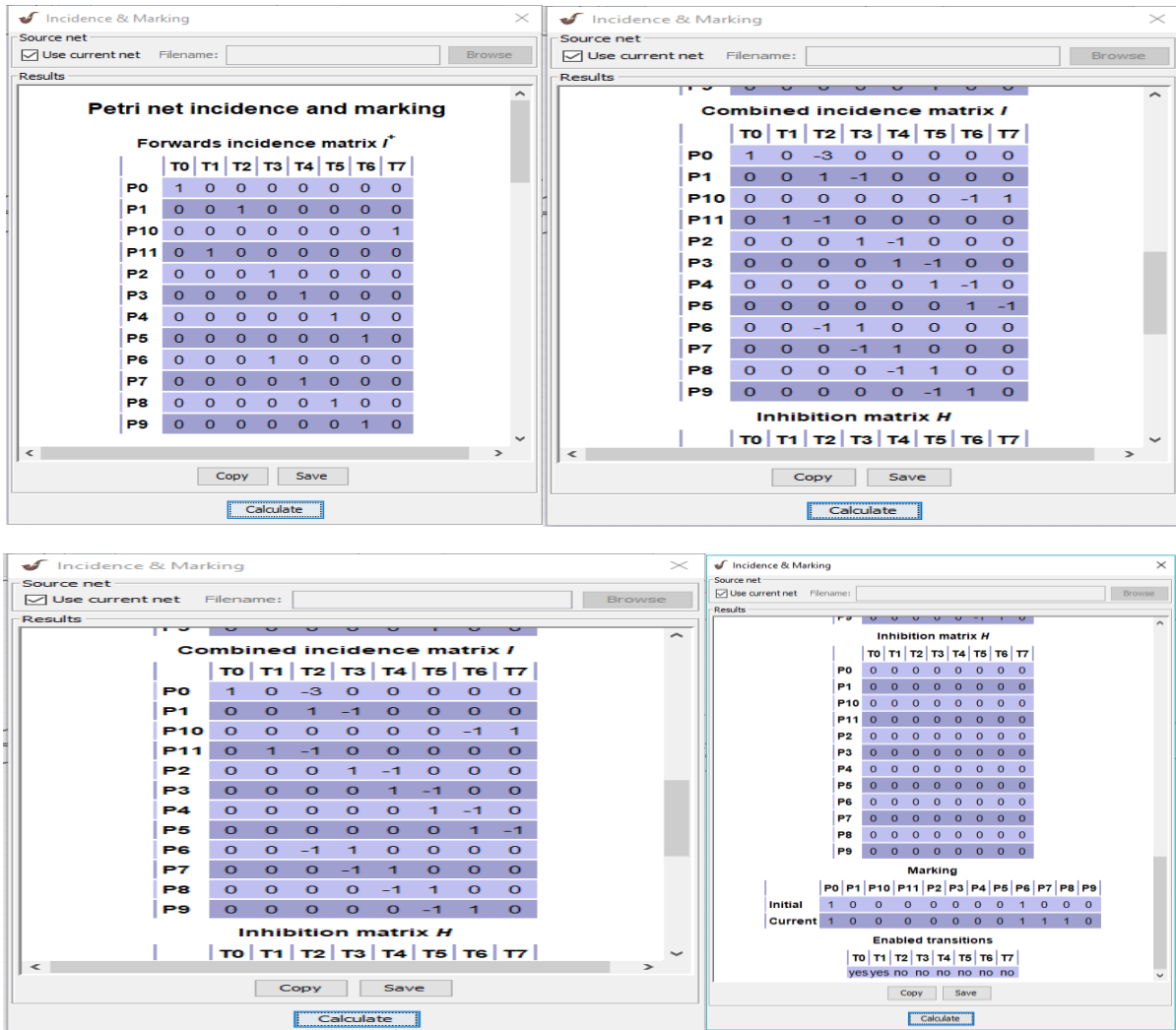


Figure 4.3 - l'outil «Petri net incidence and marking».

4.3.4. Siphons minimaux et pièges minimaux :

Nous allons utiliser l'outil «Minimal siphons and minimal traps» pour ce faire on va double cliquer sur cet outil:

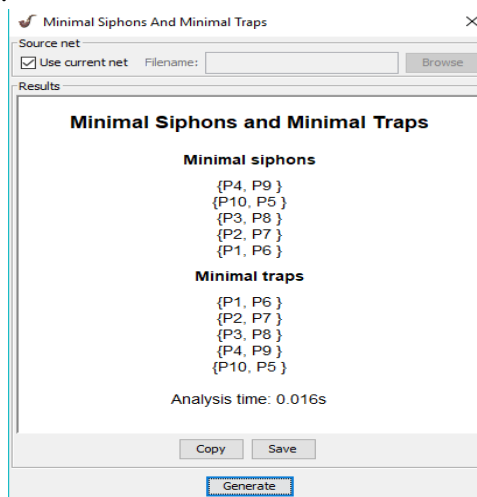


Figure 4.4 - l'outil «Minimal siphons and minimal traps».

4.4. Graphe d'accessibilité/couverture:

Ce module fournit une représentation visuelle de toutes les séquences de tir possibles pour le Réseau de Pétri donné, comme décrit dans la section 2. Il donne également des informations sur la délimitation, la sécurité et des propriétés sans blocage.

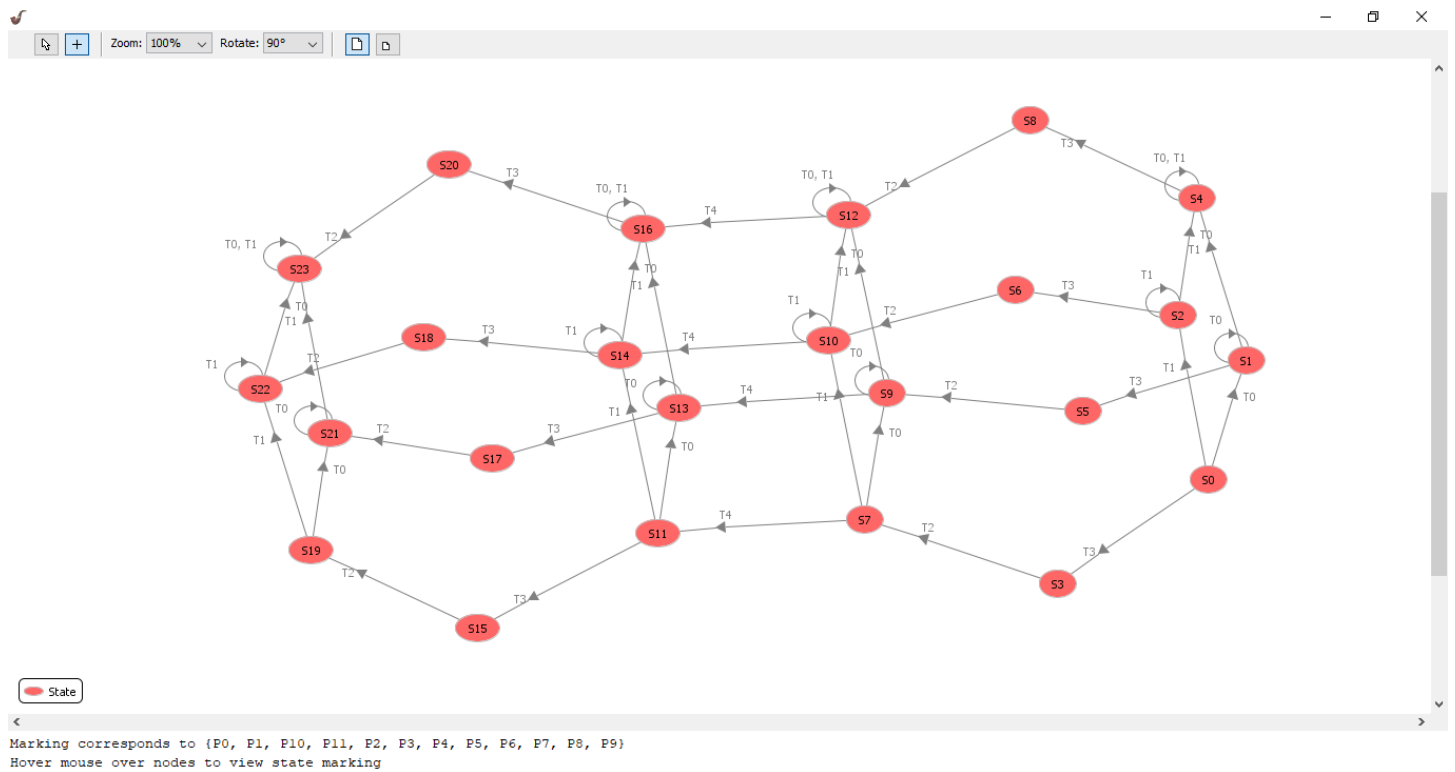


Figure 4.5 - l'outil de graphe d'accessibilité/couverture.

4.5. Simulation :

Par simulation, ce module calcule le nombre moyen de jetons par place le long avec l'intervalle de confiance à 95% pour chaque place dans le filet. Une approche analytique complète d'un problème souvent demande une énorme quantité de ressources, tandis que la simulation peut fournir une méthode alternative pour surmonter les problèmes de performance. Ce module fonctionne sans aucune intervention de l'utilisateur et fonctionne en conjonction avec les modules d'analyse afin que les résultats de performance du système simulé puissent être calculés.

Nous allons maintenant procéder à la première simulation de ce modèle. Pour cela, cliquez sur l'outil simulation :

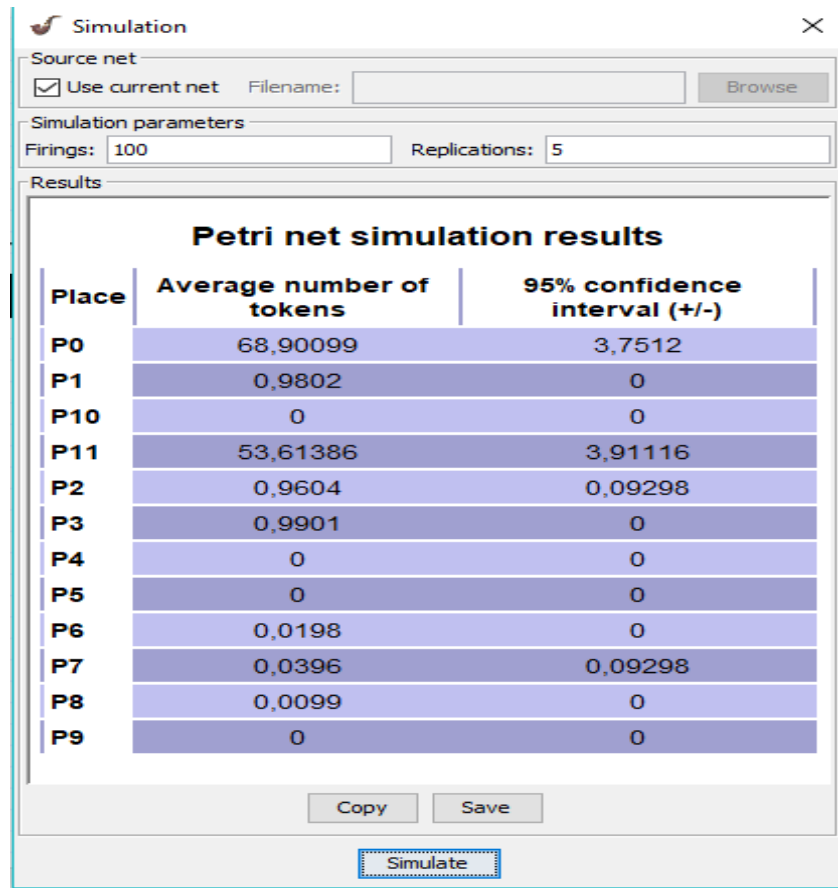


Figure 4.6 – Résultats de simulation la ligne de production de beure.

4.6. Analyse de l'espace d'état :

Ce module construit une arborescence de tous les marquages atteignables qui sert à déterminer les propriétés qualitatives du Réseau de Pétri donné : délimitation, sans impasse et sécurité. Ça aussi fournit le chemin le plus court vers le blocage dans le cas où il en existe un.

Nous allons ensuite utiliser l'outil « State Space Analysis» pour vérifier qu'il n'y a pas de deadlock (blocage) dans notre système.

Pour ce faire on va double cliquer sur cet outil :

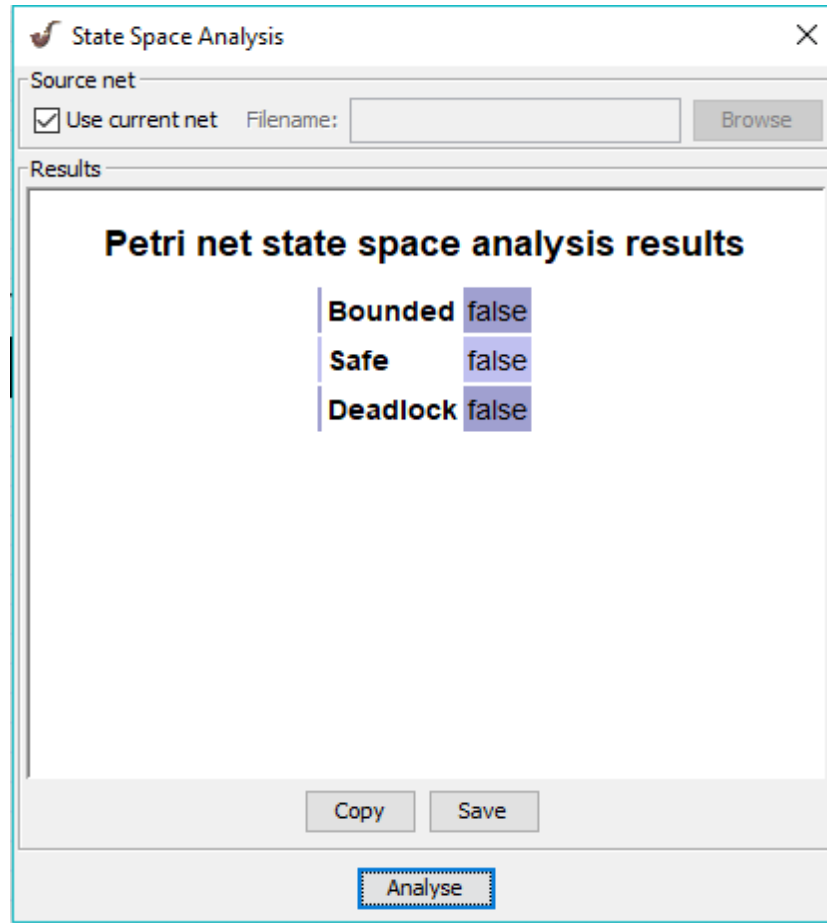


Figure 4.5 - Analyse de modèle de la ligne de production de beurre dans l'espace d'état.

4.7. Conclusion :

Dans ce chapitre, nous nous sommes intéressés à la simulation de la chaîne de production de Beurre modélisé par l'outil graphique Réseau de Pétri dans un environnement PIPE2.

Conclusion Générale

Ce rapport de mémoire a pour objet une présentation rapide des différentes techniques d'étude des SED, mais ne prétend pas à l'exhaustivité. On a pu voir que des approches très différentes sont possibles. Chaque approche a été développée dans un but précis : commande de système, évaluation de performances, vérification (temporelle)... cependant, deux axes principaux se dégagent, le premier est la modélisation par réseaux de Pétri et le deuxième utilise les automates.

Dans le cas des Réseaux de Pétri la structure du modèle inclut les spécifications de fonctionnement. De ce fait l'évaluation de performances paraît être une approche plus naturelle. On a vu par exemple que dans le cas des Réseaux de Pétri temporisés on peut caractériser les trajectoires d'un système et borner son taux de production. Cependant, s'il est plus ardu, le contrôle des SED sous forme Réseaux de Pétri est une des directions actuelles de recherche, on peut en particulier citer la thèse de L. Libeaut [13] et les travaux effectués au LISA , regroupé sous le thème : « Contribution à la commande des Systèmes à Evénements Discrets », en particulier la thèse de B. Cottenceau [6]. Le cas particulier des graphes d'événements temporisés fait aussi l'objet de nombreux travaux très aboutis, il faut cependant noter que sa structure lui interdit la représentation des systèmes permettant des divergences en « ou ».

En utilisant les graphes d'événements temporisés, nous arrivons à modéliser un système de la ligne de production de beurre et ensuite simuler et évaluer ses performances en utilisant le logiciel de simulation PIPE2. Le GET avec ajout /retrait de jetons nous permet de minimiser le temps de correspondance au niveau de traitement de produit de en jouons sur la ligne qui représente le circuit critique du linge de production. La valeur propre maximale de la matrice du système d'équations linéaires nous permet d'obtenir le modèle mathématique représenté par des équations récurrentes.

Références Bibliographiques

- [Baccelli 1992] F. Baccelli, G. Cohen, G.-J. Olsder et J.-P. Quadrat. Synchronization and linearity : An Algebra for Discrete Event Systems. Wiley and Sons, 1992.
- [Baynat 2000] B. Baynat. Théorie des files d'attente : des chaines de markov aux réseaux à forme produit. Ed Hermès, 2000.
- [Boimond and Ferrier, 1996] Boimond, J.-L. and Ferrier, J.-L. (1996). Internal Model Control and MaxAlgebra. *IEEE Trans. on Automatic Control*, 41(3):457–461.
- [Cohen 1985] G. Cohen, D. Dubois, J.-P. Quadrat et M. Viot. A linear system theoretic view of discrete event processes and its use for performance evaluation in manufacturing. *IEEE Trans. on Automatic Control*, vol. 30, no. 3, pages 210–220, March 1985.
- [Cohen et al., 1989] Cohen, G., Moller, P., Quadrat, J.-P., and Viot, M. (1989). Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *IEEE Proceedings: Special issue on Discrete Event Systems*, 77(1):39–58.
- [Cohen 1998a] G. Cohen, S. Gaubert et J.-P. Quadrat. Timed-Event Graphs with Multipliers and Homogeneous Min-Plus Systems. *IEEE Transaction on Automatic Control*, vol. 43, no. 9, pages 1296–1302, 1998.
- [Cottenceau, 1999] Cottenceau, B. (1999). *Contribution à la commande de systèmes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les dioïdes*. Thèse, LISA - Université d'Angers.
- [Cottenceau et al., 2000] Cottenceau, B., Lhommeau, M., Hardouin, L., and Boimond, J.-L. (2000). Data Processing Tool for Calculation in Dioid. In *Workshop On Discrete Event Systems (WODES'2000)*, Ghent, Belgium.
- [Cottenceau et al., 2001] Cottenceau, B., Hardouin, L., Boimond, J.-L., and Ferrier, J.-L. (2001). Model Reference Control for Timed Event Graphs in Dioid. *Automatica*, 37:1451–1458.
- [David 1992] R. David et H. Alla. Du grafcet aux réseaux de Pétri. Editions Hermès, Paris, 1992.
- [Gaubert 1992] S. Gaubert. Théorie des systèmes linéaires dans les dioïdes. Thèse, Ecole des Mines de Paris, July 1992.
- [Gunawardena 1998] J. Gunawardena, éditeur. Idempotency. Publications of the Newton Institute. Cambridge University Press, 1998.
- [Hamaci 2004] S. Hamaci, J.-L. Boimond, S. Lahaye et Mostefaoui.M. Sur la linéarisation des graphes d'événements temporisés discrets avec multiplieurs dans l'algèbre (Min,+). *Revue e-STA, Sciences et Technologies de l'Automatique*, vol. 1, no. 3, 2004.
- [Maia et al., 2003] Maia, C.-A., Hardouin, L., Santos-Mendes, R., and Cottenceau, B. (2003). Optimal Closed-Loop Control of Timed-Event Graphs in

- [Menguy et al., 2000a] Dioids. *IEEE Trans. on Automatic Control*, 48:2284–2287.
- [Menguy et al., 2000a] Menguy, E., Boimond, J.-L., Hardouin, L., and Ferrier, J.-L. (2000a). A First Step Towards Adaptative Control for Linear in Max Algebra. *Journal of Discrete Event Dynamic Systems*, 10(4):347–367.
- [Murata 1989] T. Murata. Pétri Nets: Properties, Analysis and Applications. In *IEEE Proceedings: Special issue on Discrete Event Systems*, volume 77, pages 541–581, 1989.
- [Pétri 1962] C.-A. Pétri. Kommunikation mit Automaten. Thèse, Institut für Instrumentelle Mathematik, Bonn, Germany, 1962.
- [Proth 1995] J.M. Proth et X. Xie. Les réseaux de Pétri pour la conception et la gestion des systèmes de production. Edition MASSON, 1995.