



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et télécommunication

Par :

NADJI Louiza
DEHMI Kheira

Sur le thème

Etude de propriétés qualitatives du protocole :Bit Alterné avec les réseaux de Petri

Soutenu publiquement le 04/ 10 /2021 à Tiaret devant le jury composé de :

Mr. CHENINE Abdelkader	Grade Université MA	Président
Mme. HAMDANI Abdia	Grade Université MA	Encadreur
Mr.KHARROUBI Sahraoui	Grade Université MC	Examineur

2020-2021

Remerciements

Je tiens tout d'abord à remercier infiniment notre encadreur Madame HAMDANI Abdia, pour sa grande disponibilité et ses encouragements tout au long de ce travail.

Je remercie également Mr. CHENINE Abdelkader et Mr. KHROUBI Sahraoui d'avoir accepté d'évaluer et de juger ce travail.

Après j'adresse mes plus sincères remerciements à toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail. Mes remerciements vont particulièrement à mes parents qui n'ont jamais cessé de me soutenir.

En fin, nous ne pouvons achever ce travail sans exprimer mes gratitude à tous les enseignants de département informatique,

Université Ibn-khaldoun-Tiaret,

Pour leur dévouement et leur assistance tout au long de cette Année.

Dédicaces

*C*est avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à :
Mes très chers parents, aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consentis pour mon instruction et mon bien-être.

*J*e vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

*Q*ue ce modeste travail soit l'exaucement de vos vœux tant formules, le fruit de vos innombrables sacrifices. Puisse Dieu, le très haut vous accorder santé, bonheur et longue vie.

*M*es chères familles, petites ou grandes, proches ou lointaines. Tous mes amis, sans qui la vie me semblerait bien fade, je vous souhaite la prospérité et le succès.

*T*ous ceux qui nous ont assistés, dans la réalisation et le bon déroulement de ce travail.

Louiza

Dédicaces

*A*tous ceux qui m'ont appris une lettre dans ce monde

*C*eux qui n'ont pas épargné une âme dans mon éducation, ma tendre mère

*M*on père, pour son soutien et la confiance qu'il m'accorder

J'espère que mon Dieu remplit leurs cœurs de contentement et de sérénité et les garde au-dessus de nos têtes

*M*es chers frère et sœurs

*A*oute ma famille et mes amis

Kheira

Résumé

Les réseaux de Petri (Rdp) sont très connus pour la modélisation, l'analyse et la vérification des systèmes complexes. Plus-encore, leur aspect graphique et leur sémantique permettent de détecter facilement leurs propriétés les plus intéressantes (exp : les états accessibles, le blocage, ... etc). Tina (Time Petri Net analyzer) étant un des plus puissants environnements logiciels d'analyse et de vérification des systèmes complexes basés sur les Rdp et développé au LAAS par B. Berthomieu et le groupe OLC. L'analyse et la vérification des protocoles de communication est une tâche très importante car cela permet d'assurer *fiabilité* et *sûreté* de fonctionnement des protocoles citons : le protocole du Bit Alterné (BAP).

Mots clés: Rdp, RdpT, Propriétés, TINA, Protocole du bit alterné.

Abstract

Petri nets (TPN) are a well known tool for modeling, analyzing and verifying complex systems. Moreover, their graphical aspect and their semantics make it possible to easily detect their properties of interest (eg: accessible states, blocking, etc.). TINA (Time Petri Net Analyser) is one of the most powerful software to analyze and verify complex systems based on PN and developed at LAAS by B. Berthomieu and the OLC group. Analyzing and verifying the communication protocols is an important task thus, permit to insure *correctness* and *safety* of the protocol as: the Alternating Bit Protocol (ABP).

Key words: PN, TPN, Properties, TINA, alternating bit protocol.

Table des matières

Remerciements

Dédicaces

Résumé

Liste des figures

Liste des tableaux

Liste des Abréviations

Introduction générale

Introduction générale02

Chapitre – I –

Les réseaux de Petri

1. Introduction06

2. Concepts de base06

2.1. Condition06

2.2. Événement06

2.3. Déclenchement, pré-condition, post-condition06

2.4 Modélisation d'un système événement - condition07

3. Définition des réseaux de Petri.....07

3.1. Définition informelle07

3.2. Définition formelle08

4. Les Caractéristiques des RdP08

5. Marquage08

6. Les principales propriétés qualitatives des RdP09

6.1. Les propriétés structurelles09

6.2. Les propriétés comportementales09

6.2.1. RdP Borné10

6.2.2. Blocage10

6.2.3. Réinitialisable (propre)11

6.2.4. Quasi-vivacité12

6.2.5. Vivant.....13

6.2.6. Infiniment actif13

7. Franchissement d'une Transition14

8. Exemple de modélisation en RDP15

9. Les différentes extensions des RdP	16
9.1. Le RdP à capacité limitée	16
9.2. Le RdP généralisé	16
9.3. Le RdP coloré	17
9.4. Le RdP continu	17
9.5. Le RdP hybride	17
9.6. Les RdP non autonomes	17
9.6.1. Le RdP à prédicats	17
9.6.2. Le RdP synchronisé	17
9.6.3. Le RdP T-temporisé	17
9.6.4. Le RdP P-temporisé	18
9.6.5. Le RdP stochastique	18
9.6.6. Le RdP à objet	18
10. Conclusion	18

Chapitre – II–

Les réseaux de Petri temporels

1. Introduction	20
2. Les réseaux de Petri T-temporel :	21
2.1. Définition	21
2.2. Définition formelle des t –RdP	21
3. Les autres extensions temporelles	22
3.1. Les réseaux de Petri p-temporels	22
3.1.1. Définition	22
3.1.2. Définition formelle des p-RdP	22
3.2. Les réseaux de Petri a-temporel :	23
3.2.1. Définition	23
3.2.2. Définition formelle des a-RdP	22
4. Comparaison entre les RdP t-temporels et p-temporels et a-temporels	24
5. Distinction par rapport aux places, aux transitions et arc :	25
6. Domaines d'application des réseaux de Petri temporels	25
7. Conclusion	25

Chapitre – III –

L'espace d'état des réseaux de Petri temporels

1. Introduction.....	27
2. Classe d'état	27
2.1. Définition	27
2.2. Condition de tir d'une transition :	28
2.2.1. Définition	28
3. Graphe des classes d'états :.....	28
3.1. Définition	28
3.2. Algorithme de calcul du SCG	28
4. Graphe de classes d'états fortes [22].....	29
4.1. Classe d'état	29
4.1.1. Définition	30
4.1.2. Transition de classes d'états fortes	30
5. Graphes de classes d'états atomiques [22, 23].....	31
6. La méthode des matrices de distances de Boucheneb	32
6.1. Classe d'état	33
6.2. Transition d'état.....	33
6.3. Construction du graphe des classes d'états.....	34
7. Conclusion	36

Chapitre – IV –

Cas d'étude : Protocole du bit alterné (ABP)

1. Introduction.....	38
2. Le protocole TCP/IP	38
2.1. Les caractéristiques de protocoles TCP/IP	38
3. Le Modèle OSI	38
4. Le protocole du bit alterné.....	39
4.1. Avantages.....	40
4.2. Inconvénients.....	40
4.3. Principe de fonctionnement du BA.....	41
5. Conclusion	45

Chapitre – V –

Cas d'étude : Analyse et vérification du protocole Bit Alterné

1. Introduction	47
2. L'outil Tina	47
2.1. Fonctionnalités générales	47
2.2. Interface avec Tina	48
2.3. Graphe de classes sous Tina	48
2.3.1. Préservation des propriétés LTL, classes d'états linéaires	48
2.3.1.1. Classes d'états linéaires, mode W	48
2.3.1.2. Classes d'états linéaires fortes, mode S	48
2.4. Les interfaces de Tina	49
2.4.1. L'interface officielle	49
2.4.2. Les extensions de TINA	49
2.4.2.1. Les entrées	49
2.4.2.2. Les sorties	52
3. Expérimentation	53
3.1. Discussion	69
3.2. Comparaison	70
3.3. Discussion	70
4. Conclusion	70

Conclusion générale

Conclusion générale	72
Références bibliographique	74
Annexes	

Liste des figures

Figure 1 : Exemple simple d'un réseau de Petri.	08
Figure 2 : Exemple de marquage.....	09
Figure 3 : Les propriétés structurelles.	09
Figure 4 : Un réseau de Petri non borné.	10
Figure 5 : Exemple de réseau non bloquant.....	10
Figure 6 :Exemple de réseau bloquant.	11
Figure 7 :Exemple de réseau propre.....	12
Figure 8 : Exemple de réseau non propre.	12
Figure 9 : Réseau quasi-vivacité.	12
Figure 10 : Réseau non réinitialisable mais vivant.	13
Figure 11 : Réseau bloquant et infiniment actif.....	13
Figure 12 : Réseau non infiniment actif.	14
Figure 13 : Exemple de franchissement d'une Transition.	15
Figure 14 : Exemple de modélisation en RDP d'une piscine.	15
Figure 15 : Exemple de réseau temporel de Merlin.	20
Figure 16 : Un réseau temporel.....	22
Figure 17 : Exemple d'un réseau p-temporel.....	23
Figure 18 : Exemple du a-RdP.....	24
Figure 19 : Modèles des réseaux de Petri temporel.....	24
Figure 20 : Distinction par rapport aux places, aux transitions et arcs.	25
Figure 21 : RdPT et son SCG.....	29
Figure 22 : Un réseau de Petri et son SSCG.....	31
Figure 23 : a) Réseau de Petri. b) graphe de classe LTL. c) graphe de classe CTL.	32
Figure 24 : a) RDP t-temporel b) Son graphe de classes.....	34
Figure 25 : Le protocole TCP/IP et les couches OSI.	39
Figure 26 : Exemple bit alterné Situation idéale.....	42
Figure 27 : Exemple bit alterné Situation réelle.	43
Figure 28 : Exemple bit alterné Situation naïve.....	43
Figure 29 : Exemple bit alterné Situation pas efficace.....	44
Figure 30 : La page d'accueil de TINA.....	49
Figure 31 : Création graphique d'un réseau (.ndr).....	49
Figure 32 : Graphe sous (.ndr).....	50
Figure 33 : Création textuelle d'un réseau (.net).....	50

Figure 34 : Réseau textuel (.net)	51
Figure 35 :Traduction du graphe en texte.....	51
Figure 36 :création de graphe depuis un texte	52
Figure 37 : La sortie graphique (.adr).....	52
Figure 38 : La sortie textuelle (.aut).....	53
Figure 39 : Un réseau temporel pour le Protocole du Bit Alterné (.ndr).....	53
Figure 40 : Le texte correspond au graphe de BA(.net)	54
Figure41 : Les étapes d'analyse fondamentale de l'outil Tina	55
Figure 42 : Le SCG du protocole BA.....	56
Figure 43 :La suite de la figure 42	57
Figure 44 :La forme textuelle du SCG (.aut).....	58
Figure 45 : Le Graphe de classes SCG de BA (.adr).....	58
Figure 46 : Le SSCG de protocole BA.....	59
Figure 47 : La suite de la figure 46	59
Figure 48 :Le texte du SSCG (.aut).....	60
Figure 49 : Le graphe SSCG (.adr)	60
Figure50 : Le ASCG du protocole BA	61
Figure 51 :La suite de la figure 50	62
Figure 52 : Le texte d'algorithme ASCG (.aut)	63
Figure 53 : Le graphe d'algorithme ASCG	63
Figure 54 : Les classes de graphe atomisé mode (-U).....	64
Figure 55 : Le texte de classes atomisé mode (-U)	64
Figure 56 : Le graphe de classes atomisé mode (-U)	65
Figure 57 : Les classes d'états mode (E).....	66
Figure 58 : Le texte de classes d'états mode (E).....	66
Figure59 : Le graphe de classes d'états mode (E).....	67
Figure 60 : Les classes d'algorithme de marquage	68
Figure 61 : Le texte du graphe de marquage.....	68
Figure 62 : Le graphe de marquage mode (M)	69

Liste des tableaux

Tableau 1 : Les classes, marquage et domaines de tir du graphe de figure23.....	30
Tableau 2 : Comparaison entre les différents graphes des classes.....	36
Tableau 3 : Comparaison entre les différents types des graphes.	69


Annexes

Annexe Le détail des classes

Tableau 4 : Résultat détaillé de SCGduBA	
Tableau 5 : Résultat détaillé de SSCG du BA	
Tableau 6 : Résultat détaillé de ASCG du BA	
Tableau 7 : Résultat détaillé de graphe d'états	
Tableau 8 : Résultat détaillé du graphe de Marquage	
Tableau 9 : Résultat détaillé Graphe atomisé mode (U)	

Liste des Abréviations

RdP :	Réseaux de Petri.
BA:	Bit alterné.
TCP:	Transmission Control Protocol.
IP:	Internet Protocol.
ARPANet:	Advanced Research Projects Agency.
LAN:	Local Area Network.
WAN:	Wide Area Network.
UDP:	User Datagram Protocol.
OSI:	Open System Inter connexion.
ISO:	International Standard Organisation.
ACK:	ACKnowledgement.
LTL:	Linear Temporal Logic
CTL:	Computing Temporal Logic
SCG:	Stat Class Graph.
SSCG:	Strong Stat Class Graph.
ASCG:	Atomic Stat Class graph.
DBM:	Difference Bound Matrix.
TPN:	Time Petri Nets.
RdpT:	Réseaux de Petri Temporel.
A-RdP:	Réseaux de Petri a-temporel.
P-RdP:	Réseaux de Petri p-temporel.
T-RdP:	Réseaux de Petri t-temporel.
TAPN:	Time Arc Petri Nets.
TINA :	Time Petri Net Analyser.



Introduction générale

Introduction générale

L'analyse et la vérification des systèmes permet de d'assurer fiabilité et sûreté des systèmes complexes. Les réseaux de Petri (Rdp) est l'un des premiers outils permettant la modélisation, l'analyse et la vérification des systèmes

Les réseaux de Petri sont des outils graphiques et mathématiques permettant de modéliser le comportement dynamique des systèmes à événements discrets comme les systèmes manufacturiers, les systèmes de télécommunication, les réseaux de transport, ...etc. Leur représentation graphique permet de visualiser d'une manière naturelle le parallélisme, la synchronisation, le partage de ressources, les choix, ...etc. Leur représentation mathématique permet d'établir les équations d'états, à partir desquelles il est possible d'apprécier les propriétés du modèle et de les comparer avec le comportement du système modélisé.

En tant que formalisme de spécification, les réseaux de Petri présentent un certain nombre de qualités très importantes :

- Ils disposent d'une définition formelle : ce caractère formel permet de produire des spécifications exemptes d'ambiguïté ; chaque construction des modèles possède une sémantique parfaitement définie.
- Ils sont exécutables : les modèles peuvent être interprétés par un programme construit à partir de la définition formelle de la notation.
- Ils disposent de nombreuses techniques de vérification automatique des propriétés des modèles. Il est possible de rechercher des propriétés génériques telles que le caractère borné, vivant ou réinitialisable.
- Ils disposent d'une représentation graphique attrayante, qui accroît la lisibilité et facilite la compréhension des modèles.

Les réseaux de Petri sont également étendus avec des informations temporelles (LTL), (CTL) et(CTL*). Les deux extensions temporelles sont les réseaux de Petri temporisés et les réseaux de Petri temporels [14]. Les premiers considèrent des durées minimales de tir pour les transitions alors que les seconds ajoutent des intervalles temporels de tir aux modèles. Il existe également plusieurs manières d'intégrer le temps dans les réseaux de Petri, en plaçant les informations temporelles soit sur les transitions, soit sur les places, soit sur les arcs.

La logique temporelle linéaire LTL permet l'expression des propriétés sur les comportements linéaires d'un système, en outre la logique temporelle arborescente CTL permet d'exprimer les propriétés sur arbres d'exécution et tardivement la logique CTL* constituant une classe générale des deux dernières et permettant d'exprimer des propriétés sur les chemins aussi bien que sur les arbres d'exécution .on peut écrire $LTL + CTL \subseteq CTL^*$.

Introduction générale

Les réseaux de Petri T-temporels [14, 15] ajoutent un intervalle temporel aux transitions du réseau. En utilisant une sémantique de tir dite « forte », ils sont capables de modéliser l'urgence de certains événements. Cela permet de représenter des motifs tels que le « chien de garde » très utilisés dans les systèmes temps réel. Pour représenter l'espace d'états de ces modèles on utilise également des méthodes symboliques. La méthode du *graphe des classes d'états*[15], et celle du *graphe des zones*, sont les principales méthodes utilisées. Les principaux outils manipulant des réseaux de Petri T temporels sont Tina [21], RT-Studio, Oris, Romeo.....etc.

Pour communiquer ensemble, 2 ordinateurs en réseau doivent utiliser des règles communes, l'ensemble de ces règles qui permettent à 2 ordinateurs de communiquer ensemble s'appelle un protocole. Il existe de nombreux protocoles réseau, nous allons en étudier 2 : le protocole TCP/IP, Le protocole de bit alterné (BAP) qui est implémenté au niveau de la couche de liaison de données du modèle OSI, il ne concerne donc pas les paquets, mais les trames. Le principe de ce protocole est simple, considérons 2 ordinateurs en réseau : un ordinateur A qui sera l'émetteur des trames et un ordinateur B qui sera le destinataire des trames.

Problématique :

Nous considérons maintenant des systèmes où les canaux de communications sont asynchrones et sujets à des pertes équitables de messages. Le but du protocole Bit Alterné est de fournir des primitives d'envois et de réception de message fiable afin de rendre la gestion des pertes de messages transportent à l'application.

Ainsi, en remplaçant les primitives de communications basse niveau par le Bit Alterné.

Objective :

L'objectif de ce mémoire est l'analyse et la vérification d'un protocole de communication : 'le protocole du Bit Alterné'. En utilisant les réseaux de Petri comme outil de modélisation et TINA comme environnement logiciel.

Organisation du manuscrit :

Ce mémoire est organisé comme suit, après une **introduction générale** discutant le cadre générale de ce mémoire.

➤ **Le premier chapitre** : Ce chapitre présente quelques notions de base des «Les réseaux de Petri». Ensuite, nous donnons quelques généralités sur les Rdp en commençant par des

Introduction générale

définitions et quelques structures particulières des réseaux de Petri. Après en spécifiant les différentes propriétés de réseau, ainsi que des exemples dans le cadre de ces études.

➤ **Le deuxième chapitre** : Ce chapitre porte sur les différents types et concepts de réseau de Petri temporels «Les réseaux de Petri temporels». Une comparaison entre les différents types Rdpt est ensuite présentée, ainsi que des exemples dans ce cadre.

➤ **Le troisième chapitre** : Ce chapitre présente les différents graphes et algorithmes de calcul de L'espace d'état des réseaux de Petri temporels les plus connus dans la littérature.

➤ **Le quatrième chapitre** : Ce chapitre parle sur le protocole TCP/IP en général et sur le protocole bit alterné (ABP) comme cas spécial. .

➤ **Le cinquième chapitre** : Ce chapitre est consacré à l'analyse et à la vérification du protocole du bit alterné (ABP), notre cas d'étude.

Nous concluons notre travail avec une le dernier chapitre représentant **la conclusion générale** avec ses perspectives sur les travaux futurs à envisager.

Chapitre – I -

Les réseaux de Petri

1. Introduction

Ce chapitre vous présentera un outil de modélisation très puissant. Ceci est particulièrement vrai dans le domaine de la recherche et développement: RdP. Cet outil permet essentiellement de modéliser le système, quel que soit son domaine d'Applications (informatique, télécommunications, production, etc.) Nous allons d'abord vous présenter quelques définitions et formalisme à propos de cet outil. Ensuite, nous discuterons de certaines propriétés très utiles dans l'analyse des techniques de RdP et de simplification peuvent facilement les identifier. Nous vous présenterons également quelques structures de base couramment utilisées, en particulier dans Système de production.

En 1962, Carl Adam Petri définit le réseau qui porte son nom. Depuis deux décennies, cet outil de modélisation n'a pas été utilisé, et il jouit d'une grande réputation dans tous les laboratoires de recherche. Il a pénétré avec succès l'industrie sous une forme similaire mais conceptuellement différente de GRAFCET. [1]

Il ne fait aucun doute que cet échec est en partie dû au manque de normes dans les écoles d'ingénieurs et les universités, et l'utilisation dans les écoles d'ingénieurs et les universités est davantage axée sur la recherche que sur l'industrie. Mais la raison principale peut être que les concepteurs de systèmes d'automatisation n'avaient pas besoin d'outils de modélisation aussi puissants que les réseaux Petri jusqu'à récemment.

2. Concepts de base :

2.1. Condition:

Une condition est un prédicat logique d'un état du système. Elle est soit vraie, soit fausse.

2.2. Événement :

Les événements sont des actions se déroulant dans le système. Le déclenchement d'un événement dépend de l'état du système.

Un état du système peut être décrit comme un ensemble de conditions.

2.3. Déclenchement, pré-condition, post-condition:

Les conditions nécessaires au déclenchement d'un événement sont les pré-conditions de l'événement.

Lorsqu'un événement se produit, certaines de ses pré-conditions peuvent cesser d'être vraies alors que d'autres conditions, appelées post-conditions de l'événement deviennent vraies. [5]

2.4. Modélisation d'un système événement - condition :

Condition: modélisée à l'aide d'une place.

Événement: modélisé à l'aide d'une transition

Satisfaction d'une condition: modélisée à l'aide d'un jeton



3. Définition des réseaux de Petri

3.1. Définitions Informelles :

Un RdP est un graphe composé de 2 types de nœuds :

- Les places (P_i) qui permettent de décrire les états du système modélisé. L'ensemble de ces places est noté $P = \{P_1, P_2, \dots\}$.
- Les transitions (T_i) qui représentent les changements d'états. L'ensemble de ces transitions est noté $T = \{T_1, T_2, \dots\}$.

Les Places et transitions sont reliées par des arcs orientés. On dira qu'un RdP est un graphe biparti orienté.

A chaque arc, on attribue un poids (nombre entier). Par défaut ce nombre est égal à 1. [6]

☒ Exemple

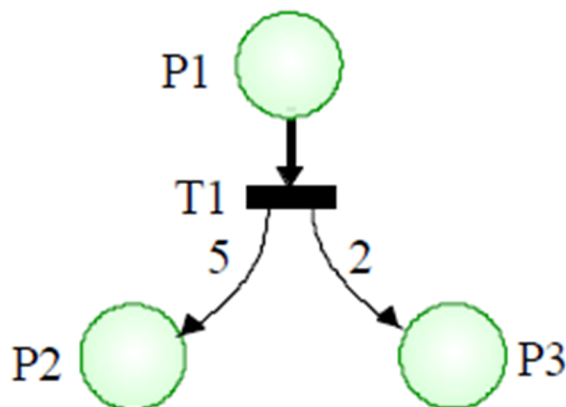


Figure 1 : Exemple simple d'un réseau de Petri

Pour le RdP ci-dessus, le poids de l'arc reliant P1 à T1 est égal à 1, alors que celui reliant T1 à P2 est égal à 5.

3.2. Définition formelle :

Un réseau de Petri N est un graphe biparti dont les nœuds sont des places et des transitions, et dans lequel les arcs relient les places aux transitions ou les transitions aux places-

Un réseau de Petri est souvent représenté par un quadruple $N = (P, T, I, O)$, où :

- (1) $P = \{p_1, p_2, \dots, p_m\}$ est l'ensemble des places.
- (2) $T = \{t_1, t_2, \dots, t_n\}$ est l'ensemble des transitions.
- (3) I est une fonction avant de type $P \times T \rightarrow \mathbb{N}$ correspondant aux arcs allant d'une place à une transition. $I(p, t) > 0$ signifie qu'il y a un arc allant de la place p vers la transition t ; $I(p, t) = 0$ signifie qu'un tel arc n'existe pas.
- (4) O est une fonction arrière de type $T \times P \rightarrow \mathbb{N}$ et correspond aux arcs allant d'une transition à une place. $O(t, p) > 0$ signifie qu'il y a un arc allant de la transition vers la place p ; $O(t, p) = 0$ indique l'absence d'un tel arc.

4. Les Caractéristiques des RdP :

Les Caractéristiques principales des RdP sont :

- Distribution des états et des changements d'états dans le réseau.
- Dépendance et indépendance d'ensembles d'événements représentées explicitement (relations de causalité).
- Représentation à différents niveaux d'abstraction (i.e. détaillés comme abstraits).
- Vérification des propriétés possibles car basées sur un formalisme mathématique rigoureux.
- Modélisation simulable.
- Représentation graphique.

5. Marquage

Chaque place contient un nombre entier (positif ou nul) de marques ou jetons. Le nombre de marque contenu dans une place P_i sera noté soit $M(P_i)$ soit M_i .

Le marquage du réseau à l'instant i , M_i est défini par le vecteur de ces marquages M_i c.-à-d. $M_i = (m_1, m_2, \dots, m_n)$.

Le marquage dit initial décrit l'état initial du système modélisé (M_0).

☒ Exemple

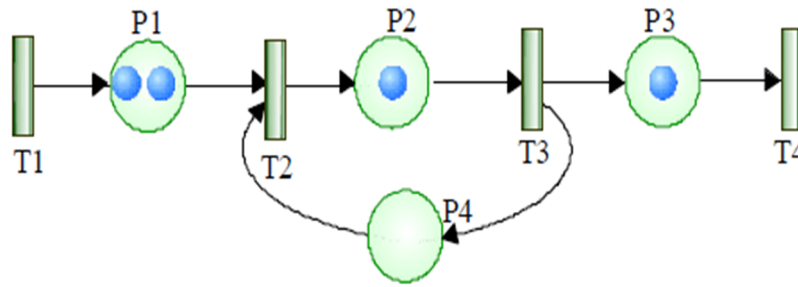


Figure 2 : Exemple de marquage

Ce RdP possède 4 places, 4 transitions et 8 arcs orientés. Soit donc :

$P = \{P_1, P_2, P_3, P_4\}$ et $T = \{T_1, T_2, T_3, T_4\}$;

Le marquage initial est $M_0 = (2, 1, 1, 0)$;

La place P_1 est en amont (une entrée) de la transition T_2 et elle est en aval (une sortie) de la transition T_1 .

T_1 est une transition sans place d'entrée: transition source.

T_2 est une transition sans place de sortie: transition puits.

6. Les principales propriétés qualitatives des RdP :

6.1. Les propriétés structurelles :

Les propriétés dépendent uniquement de la structure du réseau. Il s'agit ici de faire ressortir les propriétés statiques du système étudié. Les différentes propriétés sont indépendantes du marquage. Ainsi, il est possible de faire apparaître, entre autres, les caractéristiques de synchronisation (a) ou de précédence (b)



Figure 3 : Les propriétés structurelles

6.2. Les propriétés comportementales :

Ces propriétés dépendent à la fois du marquage initial M_0 et de la structure du réseau.

Il s'agit ici de faire ressortir les propriétés dynamiques du système étudié. En se ramenant à un système de production, l'étude des propriétés comportementales permet de mettre en avant l'influence de la réparation des ressources (matière première, machine, opérateur) sur son évolution.

6.2.1. RdP Borné :

Cette propriété répond à la question de savoir si le nombre de jetons circulant dans le réseau reste borné ou non.

Soit un réseau R et un marquage M_0 . Une place P_j du réseau marqué (R, M_0) est k -bornée si pour tout marquage M_i accessible depuis M_0 , $M_i(P_j) \leq k$.

P_j est k -borné $\Leftrightarrow \forall M_i$ accessible depuis M_0 , et $P_j \in P$, $M_i(P_j) \leq k$

Un RdP marqué est borné si toutes ses places sont bornées.

Dans le cas contraire la place P_j est dite non bornée. Il s'ensuit que le RdP est qualifié de non borné. [7]

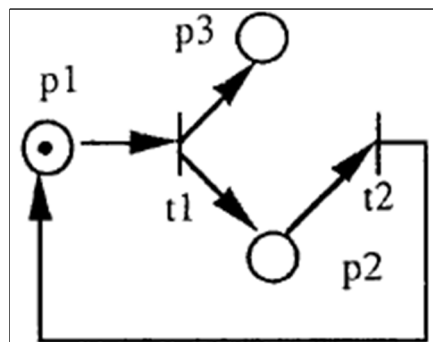


Figure 4 : .Un réseau de Petri non borné

6.2.2. Blocage

Un réseau de Petri $\langle N, M_0 \rangle$ est sans blocage si, pour tout marquage atteignable $M \in R(M_0)$ il existe toujours au moins une transition franchissable.

Un réseau de Petri $\langle N, M_0 \rangle$ est bloqué si, pour un marquage atteignable $M \in R(M_0)$, il n'y a aucune transition franchissable.

☒ **Exemple** de réseau non bloquant



Figure 5: Exemple de réseau non bloquant

☒ Exemple de réseau bloquant

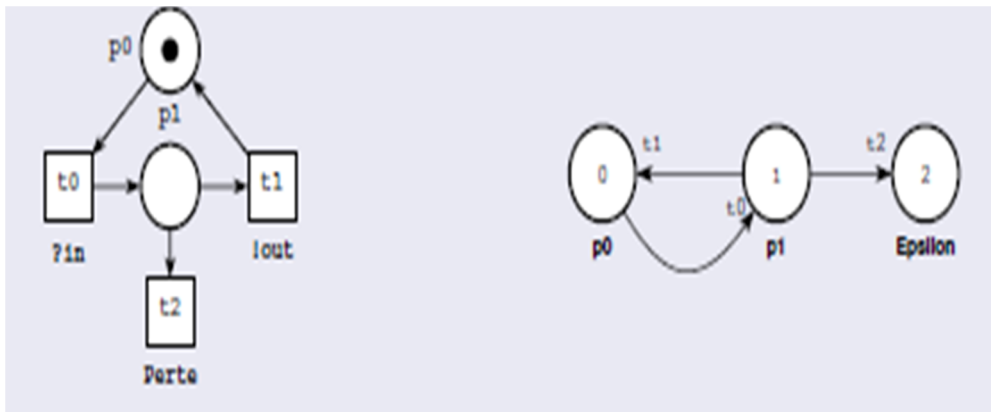


Figure 6: Exemple de réseau bloquant

6.2.3. Réinitialisable (propre)

Un Rdp est propre si et seulement le marquage M_i accessible depuis M_0 , une séquence de tirs conduisant à M_0 .

Un réseau est propre si quel que soit le marquage atteignable à partir de M_0 , il existe une séquence permettant de revenir à M_0 .

Réinitialisable (propre) si $\forall M \in A(R, M_0) M_0 \in A(R, M)$ [8]

☒ Exemple de réseau propre

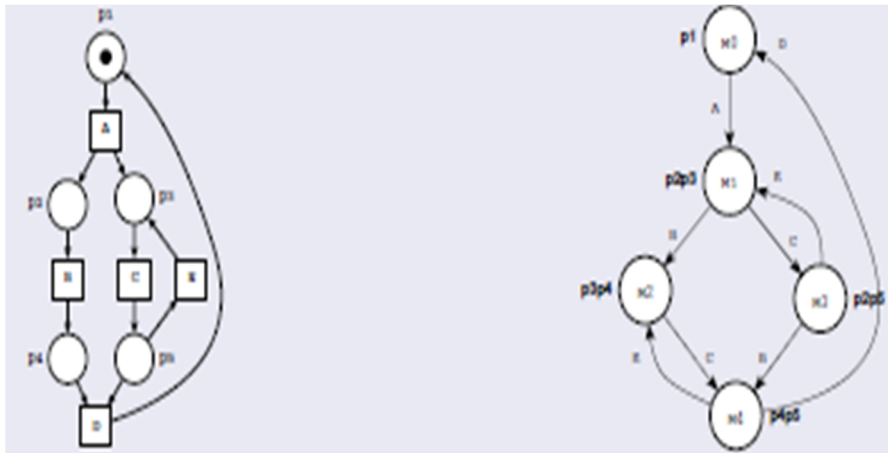


Figure 7: Exemple de réseau propre

Exemple de réseau non propre

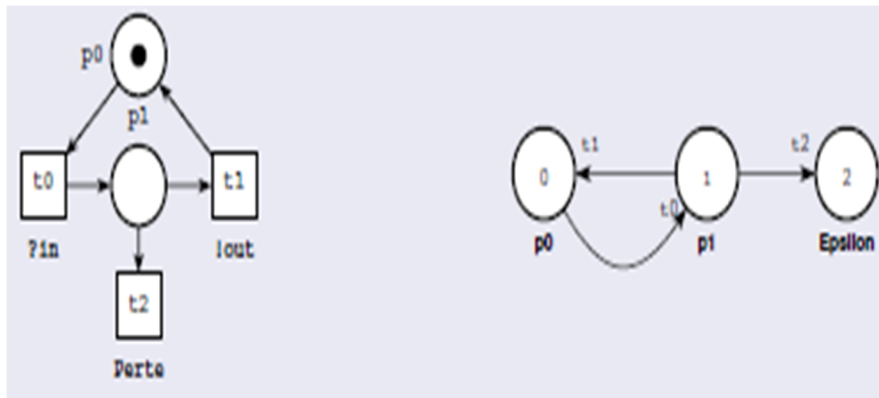


Figure 8: Exemple de réseau non propre

6.2.4. Quasi-vivacité :

Une transition t est quasi-vivante si $\exists m \in G(R, m_0) : m \rightarrow^t$

Le réseau est quasi-vivant si chacune de ses transitions est quasi-vivante.

Une transition non vivante est dite morte

Réseau quasi-vivacité :

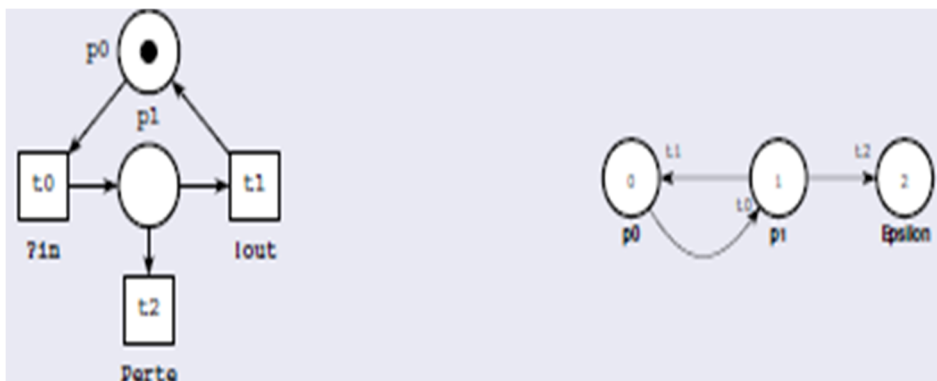


Figure 9: Réseau quasi-vivacité

6.2.5. Vivant

Une transition t est vivante si $\forall m \in G(R; m_0) : \exists w \in T^* : m \xrightarrow{w}$

Le réseau est vivant si chacune de ses transitions est vivante.

Réseau non réinitialisable mais vivant

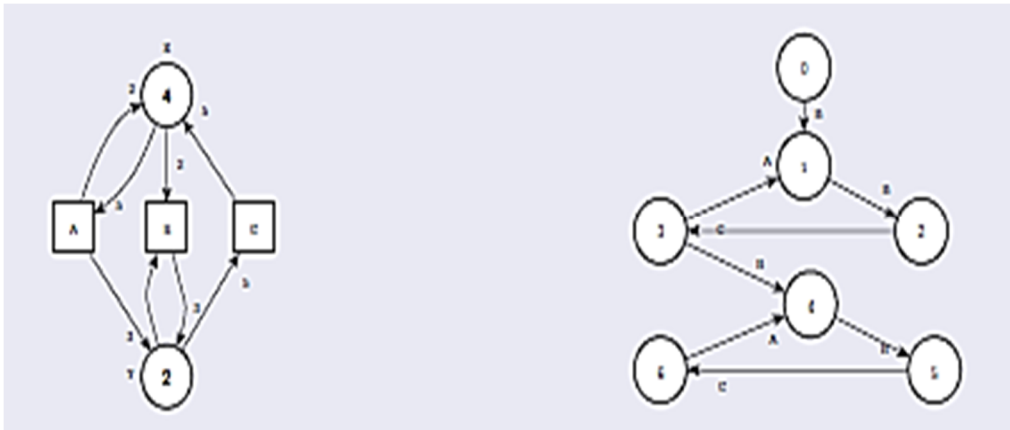


Figure 10: Réseau non réinitialisable mais vivant

6.2.6. infiniment actif

Un réseau est infiniment actif si $\forall x \in \mathbb{N}; \exists w \in T^* : m \xrightarrow{w}$

Le réseau admet au moins une séquence de franchissement infinie.

Réseau bloquant et infiniment actif

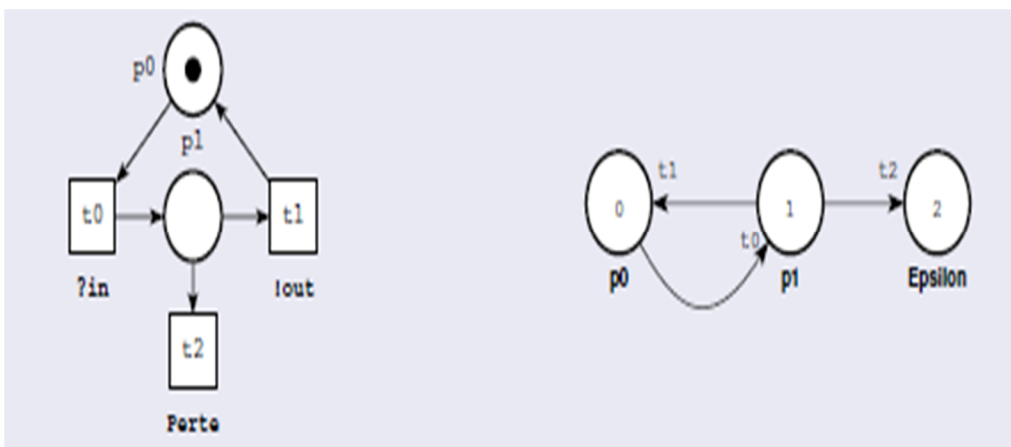


Figure 11: Réseau bloquant et infiniment actif

Réseau non infiniment actif

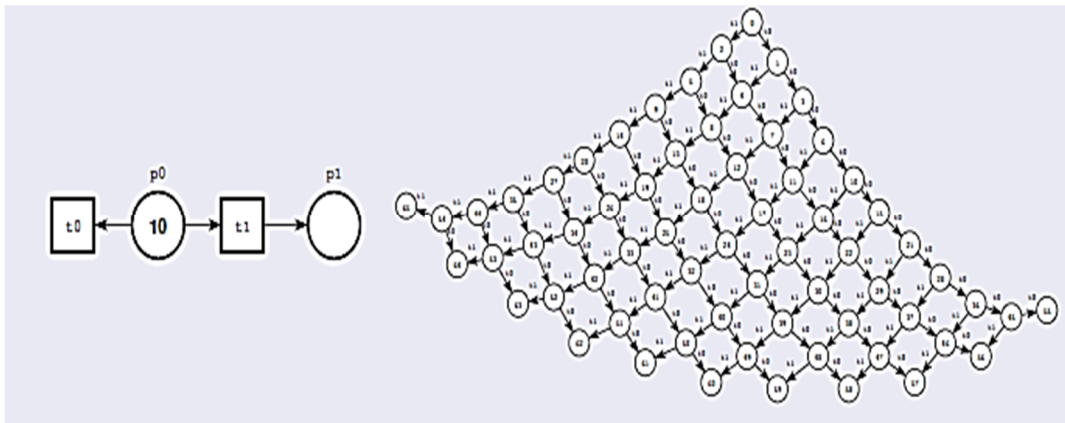


Figure 12: Réseau non infiniment actif

Toutes les séquences sont de longueur bornées (ici par 10)

☞ **Remarque :**

- ☞ Réinitialisable => Sans blocage
- ☞ Vivant => Sans blocage
- ☞ Réinitialisable + Quasi-Vivant => Vivant
- ☞ Sans blocage => Infiniment actif
- ☞ Pas de rapport entre Infiniment actif et quasi-vivacité, vivacité, propre
- ☞ Pas de rapport entre borné et bloquant, quasi-vivacité, vivacité, propre
- ☞ Non borné => Infiniment actif

7. Franchissement d'une Transition :

Pour rendre compte de l'évolution du système modélisé, les réseaux de Pétri intègrent un formalisme permettant de passer d'un marquage à un autre : c'est le franchissement des transitions.

Le franchissement (ou le tir) d'une transition ne peut s'effectuer que si chacune des places en amont (en entrée) de cette transition contient suffisamment de jetons (\geq au poids de l'arc correspondant). On dit alors que la transition est franchissable ou validée. [9]

☞ **Exemple**

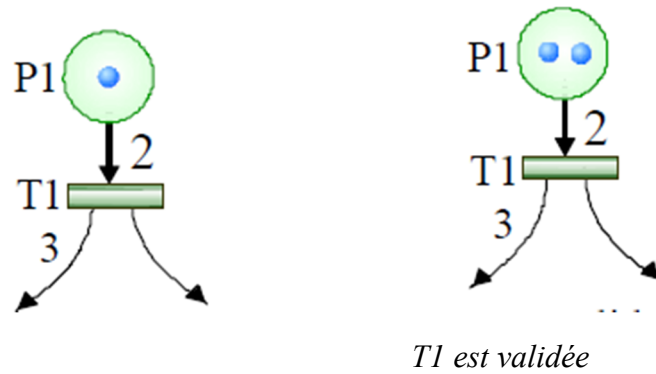


Figure 13: Exemple Franchissement d'une Transition

Pour le premier marquage, T1 n'est pas validée car le nombre de jetons dans P1 (1) est inférieur au poids de l'arc reliant P1 à T1 (2) !

8. Exemple de modélisation en RDP :

Une piscine comporte C cabines pour se changer et P paniers pour déposer ses vêtements.

- On n'entre dans la piscine que si une cabine est libre. On attend un panier pour se changer et déposer ses vêtements. On libère la cabine et on pénètre dans le bassin.
- On ne quitte le bassin que si une cabine est libre. On se change et on restitue cabine et panier. On quitte la piscine.

Un modèle possible :

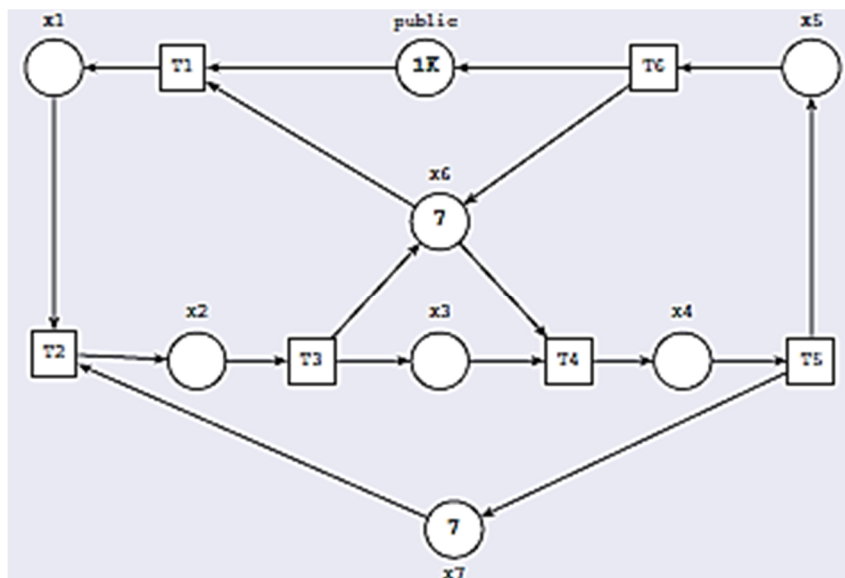


Figure 14 : Exemple de modélisation en RDP d'une piscine

T1 : le client entre _a la piscine (si 9 cabine libre (x6))	
T2 : le client se déshabille (si 9 panier libre (x7))	x1 : client en attente d'un panier x2 : client se déshabille
T3 : le client pénètre dans le bassin (et restitue la cabine (x6))	x3 : client dans le bassin x4 : client s'habille
T4 : le client quitte le bassin (si 9 cabine libre (x6))	x5 : client habille prêt à sortir x6 : Compteur de cabines libres
T5 : le client s'habille (et restitue le panier (x7))	x7 : Compteur de paniers libres
T6 : le client quitte la piscine.	

9. Les différentes extensions des RdP :

9.1. Le RdP à capacité limitée :

Dans un RdP ordinaires, la capacité des places n'est pas limitée. Ici, on affecte une capacité, défini par un nombre entier positif associé aux places.

Ainsi, le franchissement d'une transition est conditionné par la capacité des places en aval.

Cette limitation peut illustrer la capacité d'un stock par exemple.

9.2. Le RdP généralisé :

On affecte un poids. Ce poids indique le nombre de jetons consommés ou créés lors du franchissement d'une transition. On retrouve ces poids dans la matrice d'incidence. Il s'agit d'une simplification par agrégation d'un RdP ordinaire.

La pondération des arcs permet, entre autres, de mettre en avant le processus de consommation de la matière.

9.3. Le RdP coloré :

Dans un RdP coloré, les marques peuvent être différenciées par des couleurs. Si le nombre de couleurs est fini, alors on peut se ramener à un RdP classique.

Le RdP coloré est un quintuplé $R = (P, T, \text{Pré}, \text{Post}, C)$

Où $C = \{c_1, c_2, \dots, c_i, \dots, c_k\}$ = Ensemble des couleurs avec $c_i = \langle c_i, c_i, c_i, \dots, c_i \rangle$

La coloration des jetons permet de différencier les produits circulant dans un SdP. La couleur peut être soit un attribut unique, soit un n -uplet d'attributs formant ainsi un agrégat.

9.4. Le RdP continu :

Dans un RdP continu le marquage des places n'est plus un nombre entier mais un nombre réel positif. Ce type de RdP permet d'analyser des performances en termes de débit. Il est fortement utile lorsque le nombre de marquages dans un RdP classique devient trop important ou pour représenter des processus continus. [2]

Les SdP de type flux (pétrochimie, sidérurgie,...) ou de grand débit (production de masse : visserie,...) demandent une approche par les RdP continus.

9.5. Le RdP hybride :

Ces RdP sont utilisés pour représenter des SdP ayant à la fois des caractéristiques discrètes et continues. Une usine de soda par exemple possède cette double approche par la matière première (fluide continu) et son conditionnement (bouteilles, canettes).

9.6. Les RdP non autonomes :

9.6.1. Le RdP à prédicats :

Le prédicat permet d'affecter des actions de transformation sur les jetons lors du franchissement d'une transition.

Utilisé avec la coloration, le RdP à prédicat permet de mettre en évidence le processus de production et de transformation de la matière première en produit fini.

9.6.2. Le RdP synchronisé :

On associe en plus des conditions de franchissement d'un RdP ordinaire (présence de marques dans les places en amont) un événement de franchissement. Ces événements sont externes au système du RdP.

9.6.3. Le RdP T-temporisé :

Dans les RdP T-temporisés, une durée de franchissement est affectée aux transitions.

A chaque transition T_j est associée une temporisation d_j constante.

Cette particularité permet de décrire des systèmes dont le fonctionnement est dépendant du temps. Ces RdP sont utiles pour l'évaluation de performance.

9.6.4. Le RdP P-temporisé :

L'aspect temporel est ici affecté aux places. Aussi, à chaque place P_i est associée une temporisation d_i constante.

9.6.5. Le RdP stochastique :

Dans un RdPS, la durée affectée n'est plus déterministe mais aléatoire. On peut grâce à cette particularité prendre en compte des événements probabilistes comme la panne d'une machine. On utilise généralement une temporisation suivant une loi exponentielle.

9.6.6. Le RdP objet :

L'introduction de l'approche orientée objet dans les RdP [3], permet de prendre en compte une grande variété d'informations tout en évitant une explosion combinatoire.

On peut faire ressortir deux grandes tendances, les objets encapsulés dans des RdP et les RdP encapsulés dans les objets [4]. Ces RdP appartiennent à la catégorie des « RdP de haut niveau ».

10. Conclusion

Dans ce chapitre, nous comprenons que les réseaux de Petri sont une forme de formalisme:

- Facile à comprendre et à manipuler
- Dispose d'une forte sémantique de concurrence
- Les caractéristiques structurelles permettent une analyse directe du réseau sans (dans certains cas) par l'utilisation de la logique temporelle
- Idéal pour la modélisation.

Dans le chapitre suivant on va parler sur les réseaux de petri temporel comme un cas spécial.



Chapitre – II -

Les réseaux de Petri temporels

1. Introduction

L'objet de ce chapitre est analyse du fonctionnement de systèmes dans lesquels le temps apparait comme un paramètre quantifiable et continu. Les protocoles de communication sont au nombre de ce système : les mécanismes qui s'assurent la configuration après une perte de message ou un changement de topologie du réseau sont souvent implantés à l'aide de temporisations.

Plusieurs techniques ont été proposées pour spécifier vérifier de tels systèmes.

Deux de ces techniques ont été développées à partir des réseaux de Petri les réseaux temporisés (Timed Petri Nets) et les réseaux de Petri temporels (Time Petri Nets).

Les réseaux temporisés sont obtenus, à partir des réseaux de Petri, en associant une durée de tir à chaque transition. De plus, les transitions doivent être tirées dès qu'elles sont sensibilisées. Ces réseaux sont essentiellement utilisés pour l'analyse de performances.

Merlin a défini les réseaux temporels comme des réseaux de Petri dans lesquels deux temps min et max (avec $0 \leq \min \leq \max$, min fini et max éventuellement infini) sont associés à chaque transition du réseau. Ces temps, pour une transition t du réseau, sont relatifs à la date à laquelle la transition t a été sensibilisée pour la dernière fois. Supposons qu'à une date θ la transition t devienne sensibilisée ; la transition t ne peut alors être tirée avant la date $\theta + \min$ et doit l'être, au plus tard, à la date $\theta + \max$.

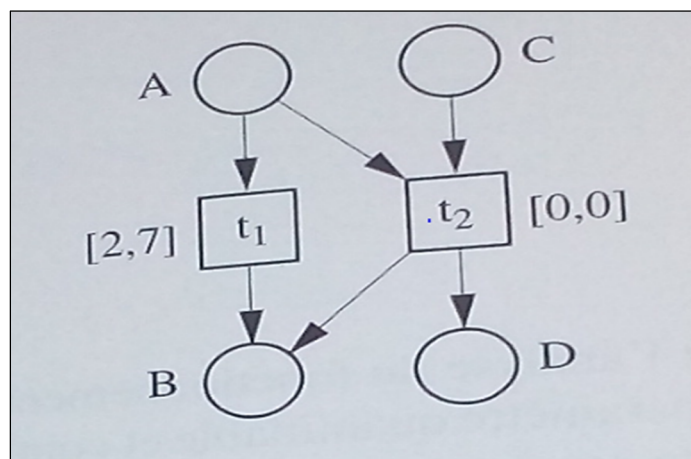


Figure 15 : Exemple de réseau temporel de Merlin

Il existe aussi plusieurs façons d'incorporer le temps dans les réseaux Petri, en plaçant les informations temporelles soit sur les transitions, soit sur les places, ou bien sur les arcs.

Dans le cas des réseaux de Petri temporels les modèles correspondants sont respectivement les réseaux de Petri T-temporels [14,15], P-temporels [12, 13] et A-temporels [16, 17, 18]. Pour chacun de ces modèles, le modèle de réseau Petri temporisé qui lui est associé est moins expressif [19].

Nous nous intéressons plus précisément aux réseaux de Petri T-temporels, ou *time Petri nets* (TPN) en anglais, dans lesquels un intervalle temporel est associé à chaque transition.

2. Les réseaux de petri T-temporel :

2.1 Définition

Les RdP t-temporels sont destinés principalement à l'étude des systèmes de télécommunication dont les évolutions dépendent des contraintes de type temps de réponse (time-out) [14] [15].

Dans ce modèle, un intervalle $[a, b]$ de temps est associé à chaque transition du réseau. L'intervalle associé à la transition t est relatif au moment où la transition devient validée. Supposons que t est validée à l'instant c , alors elle peut être franchie seulement entre $a+c$ et $b+c$, sauf si elle devient non-validée à cause du franchissement d'une autre transition avec laquelle elle était en conflit.

2.2 Définition formelle des t –RdP

Un réseau de Petri t-temporel est un tuple $\langle P, T, \text{Pré}, \text{Post}, M_0, IS \rangle$ dans lequel : $\langle P, T, \text{Pré}, \text{Post}, M_0 \rangle$ est un réseau de Petri

$IS : T \rightarrow Q^+ \times (Q^+ \cup \{\infty\})$ la fonction intervalle statique (Q^+ est l'ensemble des nombres rationnels).

$T_i \rightarrow IS_i = [a_i, b_i]$ avec $0 \leq a_i \leq b_i$

La fonction IS associe, à chaque transition t_i du réseau, un intervalle à bornes rationnelles $IS_i = [a_i, b_i]$ avec $0 \leq a_i \leq b_i$; (b_i peut être infini). a_i est la date statique de tir au plus tôt et b_i est la date de tir au plus tard de T_i .

Une transition doit être sensibilisée (validée) pendant le délai minimum a_i avant de pouvoir être tirée (franchie) et ne peut rester validée au-delà du délai maximum b_i sans être tirée.

☒ Exemple :

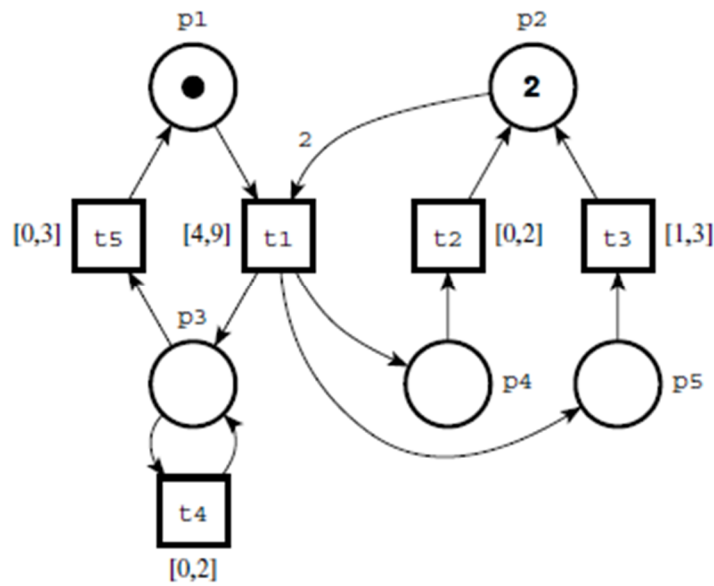


Figure 16 : Un réseau temporel.

3. Les autres extensions temporelles

On peut aussi plaçons les informations temporelles sur les places et sur les arcs séparément.

3.1. Les réseaux de Petri p-temporels

3.1.1. Définition

Développé dans [14] et [8], les réseaux de Petri p-temporels (p-RdP), ont pour objectif de modéliser des systèmes dans lesquels le temps prend un aspect non déterministe. A cet effet, le temps de séjour associé à chaque place p_i n'est plus une constante fixe et déterminé comme dans les RdPT, mais prend la forme d'un intervalle de temps IS_i . On peut citer, les unités de productions alimentaires [15], les cellules de traitements [16], et les réseaux de transport public [17].

3.1.2 Définition formelle des p-RdP

Un p-RdP est défini par le doublet $\langle R; IS \rangle$ où :

R est un réseau de Petri marqué

$$IS : P \rightarrow (\mathbb{R}^+ \cup \{0\}) \times (\mathbb{R}^+ \cup \{0, +\infty\}),$$

$$P_i \rightarrow IS_i = [a_i, b_i] \text{ avec } 0 \leq a_i \leq b_i,$$

Où $[a_i, b_i]$ définit l'intervalle statique de temps de séjour d'une marque dans la place p_i .

Le temps de séjour d'un jeton est compris dans un intervalle de temps. Le jeton qui arrive dans une place est indisponible (il ne peut pas permettre le franchissement de la transition située en aval) pendant un certain temps. Il doit passer un temps de séjour minimum correspondant à la borne minimale de l'intervalle. Le jeton atteint un âge de maturité (qui lui permet de franchir une transition) après a_i unités de temps après son arrivée dans la place. Il reste dans cet état de disponibilité durant $b_i - a_i$ unités de temps.

Une des spécificités des réseaux de Petri p-temporels est la possibilité de la mort de jetons, le mode de fonctionnement est le suivant. Après un séjour de b_i (borne max de l'intervalle) unités de temps dans la place, le jeton se trouve dans l'obligation de quitter cette dernière, sinon, il se retrouve dans un état de mort. Autrement dit, le jeton ne peut plus participer aux franchissements, et cela peut générer d'éventuels dysfonctionnements du système dans le futur si, par exemple, le jeton représente une ressource importante. Par conséquent, ce modèle va nécessiter un contrôle des tirs des transitions afin d'éviter la mort des jetons.

Exemple

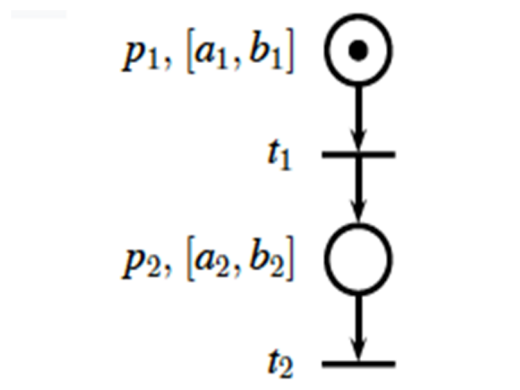


Figure17 : Exemple d'un réseau p-temporel

3.2. Les réseaux de Petri a-temporel :

3.2.1 Définition

Les réseaux de Petri à arc chronométré, dont les jetons sont annotés avec un âge (une valeur entière indiquant le temps écoulé depuis sa création), et où les arcs reliant des lieux avec des transitions ont associé un intervalle de temps, qui limite l'âge des jetons consommés pour la transition adjacente. Cependant, une transition n'est pas forcée d'être rouge lorsqu'elle a toutes ses conditions préalables contiennent des jetons avec un âge adéquat, et il en va de même si l'âge de l'un de ces jetons est sur le point d'expirer. Plus en général, dans le

modèle il y a aucune urgence, qui peut être interprétée dans le sens où le modèle est réactif, car les transitions ne seront rouges que lorsque le contexte externe l'exige. [20]

3.2.2 Définition formelle des a-RdP

On définit un a-RdP(TAPN) comme un tuple ${}^3N = (P; T; F; \text{temps})$, où P est un ensemble fini de places, T est un ensemble fini de transitions ($P \cap T = \emptyset$), F est la relation de flux ($F \subseteq (P \times T) \cup (T \times P)$), et temps est une fonction qui associe à chaque arc (p, t) dans F une paire d'entiers naturels dont le second peut être infini i.e. : $\text{temps} : F \mid P \times T \rightarrow N \times (N \cup \{\infty\})$.

☒ Exemple :

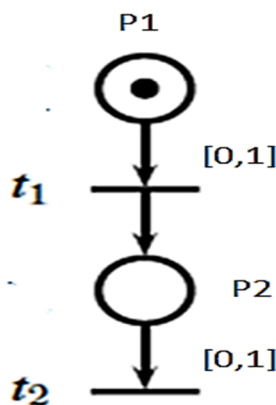


Figure 18: Exemple d'un a-RdP

4. Comparaison entre les RdP t-temporels et p-temporels et a-temporels

A partir des définitions données pour les RdP t-temporels et p-temporels nous donnons, dans les lignes qui suivent, une comparaison pour la structure de synchronisation.

Dans les réseaux de Petri temporel, le temps de synchronisation n'est pas lié aux tâches ou aux processus qui s'exécutent, mais à une attente après leur synchronisation logique sur la transition.

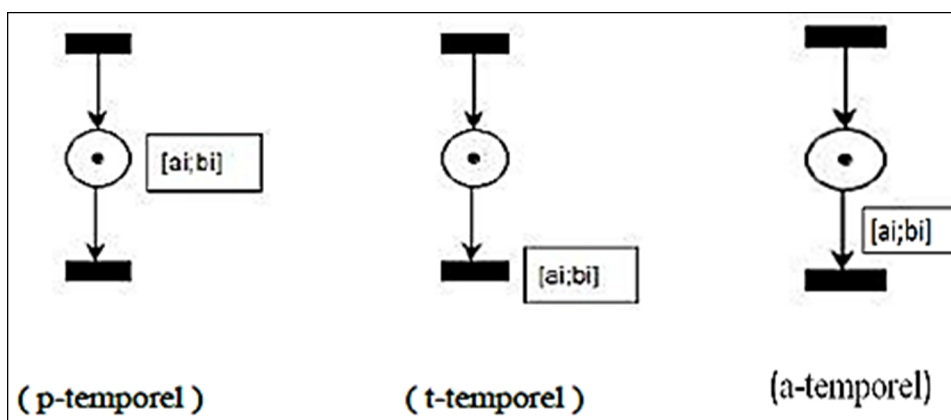


Figure 19: Modèles des réseaux de Petri temporel

5. Distinction par rapport aux places, aux transitions et arc :

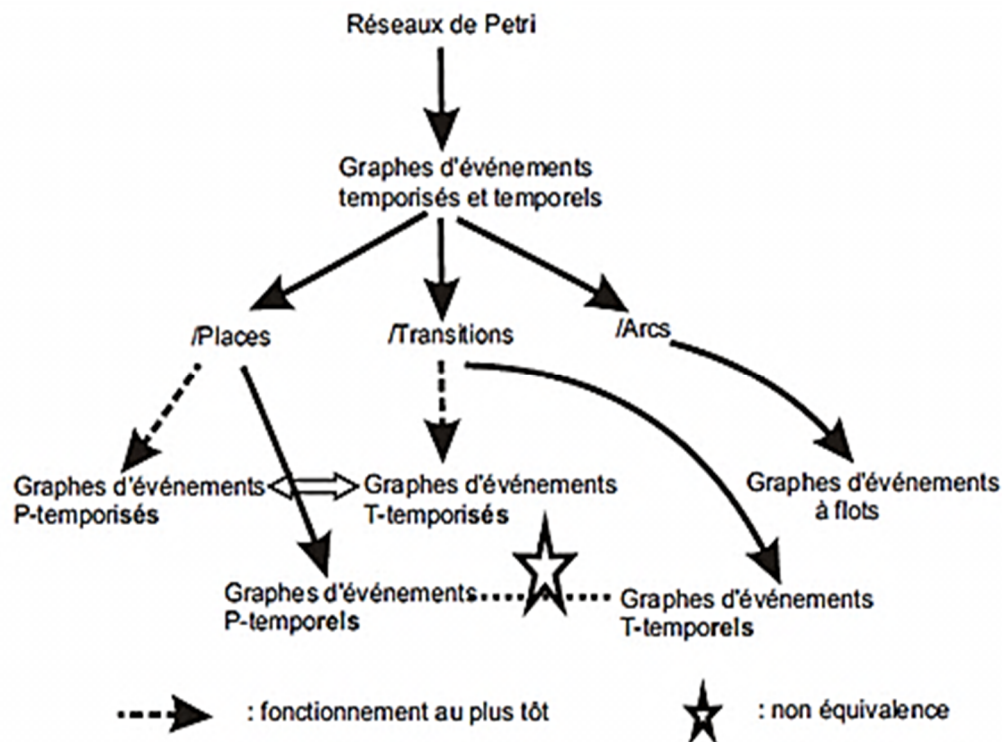


Figure20: Distinction par rapport aux places, aux transition et arcs

6. Domaines d'application des réseaux de petri temporels

Plusieurs extensions proposées pour prendre en compte le temps

- Protocoles de communications.
- Performances des systèmes.
- Ordonnancement (temps-réel).

7. Conclusion

A l'heure où les systèmes temps réel sont de plus en plus complexes et où la demande de vérification est toujours plus importante, il est intéressant d'avoir avec soi un outil permettant d'en diminuer la complexité en permettant leur décomposition en éléments plus facilement manipulables.

Nous avons présenté brièvement les principaux modèles des réseaux de Petri qui permettent une intégration simple et efficace du temps. Ces modèles présentés (RdP temporisés et RdP temporels) sont donc bien adaptés à l'évaluation des performances des systèmes, à contraintes temporelles. Plusieurs chercheurs ont utilisé cette notion de temps de cycle pour l'évaluation analytique des performances d'ateliers de fabrication [10], [11]. Le modèle considéré est dans ce cas un graphe d'événements.



Chapitre – III -

L'espace d'état des réseaux de Petri temporels

1. Introduction

Parmi les techniques proposées pour spécifier et vérifier des systèmes dans les quels le temps apparaît comme paramètre, deux sont largement utilisées : Les Automates Temporisés (voir le chapitre 5 de ce volume, consacré à ce modèle) et les Réseaux de Petri Temporels (ou *Time Petri Nets*), introduits dans [14].

Dans ce chapitre un panorama des méthodes d'analyse disponibles pour les réseaux temporels et nous discutons de leur mise en œuvre. Ces méthodes, basées sur la technique des classes d'états, ont été initiées par [15]. Ces graphes constituent des abstractions finies du comportement des réseaux temporels bornés. Différentes abstractions ont été définies, préservant diverses classes de propriétés. De plus nous discuterons du problème pratique de la vérification de propriétés (*model-checking*) de certaines logiques sur les graphes de classes construits. La mise en œuvre de ces techniques nécessite des outils logiciels, tant pour la construction des abstractions requises que pour leur vérification elle-même. Les exemples discutés dans cet article ont été traités avec les outils de l'environnement.

2. Classe d'état

Formellement, Depuis l'état initial, tous les états accessibles après franchissement de la même séquence de tir sont de la même classe, mais dans des différentes dates.

2.1. Définition

Une classe d'états est un couple $E = (M, D)$ dans lequel :

- ▀ M , est le marquage de la classe (marquage commun à tous les états de la classe) ;
- ▀ D , est le domaine de tir de la classe, défini par l'union des intervalles de tir associé à chaque état de la classe. La classe initiale E_0 contient seulement l'état initial E_0 . D peut être représenté par l'ensemble des solutions d'un système d'inéquations linéaires, comportant une variable pour chaque transition sensibilisée par le marquage M de la classe. Les inéquations de D sont de deux types ; on parle alors d'un système mis sous forme DBM (DifferenceBound Matrix) :

$$\text{▀ } \forall t_i \in E(M), 0 \leq \text{MIN}_{e \in E} \{ \text{Eft}_e(t_i) \} \leq \underline{t}_i \leq \text{MAX}_{e \in E} \{ \text{Lft}_e(t_i) \} ;$$

- ▀ $\forall t_j, t_i \in E(M) \wedge (t_i \neq t_j), \underline{t}_j - \underline{t}_i \leq \text{MAX}_{e \in E} \{ \text{Lft}_e(t_j) - \text{Eft}_e(t_i) \}$. Où \underline{t}_j et \underline{t}_i sont des variables associées respectivement aux transitions t_j et t_i .

2.2. Condition de tir d'une transition :

2.2.1 Définition (Transition tirable depuis une classe d'états)

Soit une classe d'états $E = (M, D)$, une transition t_i est tirable depuis E ,ssi :

- (i) elle est sensibilisée et non inhibée par M au sens de RDP.
- (ii) $D \wedge (\forall t_i \in E(M), \underline{t}_i \leq \underline{t})$ est consistant.

La condition (ii) interdit le tir de toute transition validée (au sens des Rdp) dont l'intervalle de tir (courant) est strictement précédé par celui d'une autre transition validée. Le tir de t_i conduit à la classe $E' = (M', D')$ suivante, caractérisée par :

▀ Le nouveau marquage M' :

$$\forall p \in P, M'(p) = M(p) - \text{Pre}(p, t) + \text{Post}(p, t);$$

▀ Le nouveau domaine D' est obtenu selon la procédure suivante:

a) Ajouter au système D , les conditions de tir de t_i exprimant qu'elle soit la première tirée parmi l'ensemble des transitions validés, i.e.

$$D \wedge (\forall t_j \in E(M), \underline{t}_i \leq \underline{t}_j);$$

b) Supprimer dans le système obtenu les inéquations incluant les variables θ_j associées aux transitions en conflit avec t_i , ces transitions sont désensibilisées par le tir de t_i ;

c) Dans le système ainsi réduit, effectuer le changement de variable suivant: $\forall i \neq j, \underline{t}_j = \underline{t}'_j + \underline{t}_i$, et éliminer par substitution toutes les occurrences de la variable \underline{t}_i , pour ne garder que les nouvelles \underline{t}'_j .

d) Compléter ce dernier système par une variable supplémentaire pour chaque transition nouvellement sensibilisée. Associer à ces transitions leurs intervalles de tir statiques.

3. Graphe des classes d'états :

3.1 Définition

Le State Class Graph (SCG en abrégé) est une abstraction finie du graphe d'état SG qui fait abstraction des transitions qui s'écoulent dans le temps mais préserve ses marques et ses traces ; le SCG embarque le graphe d'accessibilité du TPN.

3.2 Algorithme de calcul du SCG

Pour toute séquence δ tirable, L_δ peut être calculée comme suit. Calculer le plus petit ensemble de classes contenant L_e et tel que, lorsque $L_\delta \in C$ et $\delta.t$ est tirable, alors $(\exists X \in C) (X \cong L_{\delta.t})$.

– La classe initiale est $L_e = (m_0, \{\text{Eft}_s(t) \leq \underline{\theta}_t \leq \text{Lft}_s(t) \mid m_0 \geq \text{Pre}(t)\})$

-Si δ est tirable et $L_\delta = (m, D)$, alors $\delta.t$ est tirable si et seulement si :

- (i) $m \geq \text{Pre}(t)$ (t est sensibilisée par m) et ;
- (ii) le système $D \wedge \{\underline{\vartheta}_t \leq \underline{\vartheta}_i \mid i \neq t \wedge m \geq \text{Pre}(i)\}$ est consistant.

– Si $\delta.t$ est tirable, alors $L_{\delta.t} = (m', D')$ est obtenue depuis $L_\delta = (m, D)$ par :

$$m' = m - \text{Pre}(t) + \text{Post}(t),$$

D' obtenu comme suit :

a) Les contraintes (ii) ci-dessus de tirabilité de t depuis L_δ sont ajoutées à D ;

b) Pour chaque k sensibilisée par m' , une variable $\underline{\vartheta}'_k$ est introduite par :

$$\underline{\vartheta}'_k = \underline{\vartheta}_k - \underline{\vartheta}_t, \text{ si } k \neq t \text{ et } m - \text{Pre}(t) \geq \text{Pre}(k),$$

$$\text{Eft}_s(k) \leq \underline{\vartheta}'_k \leq \text{Lft}_s(k), \text{ sinon;}$$

c) Les variables $\underline{\vartheta}$ sont éliminées.

☒ Exemple

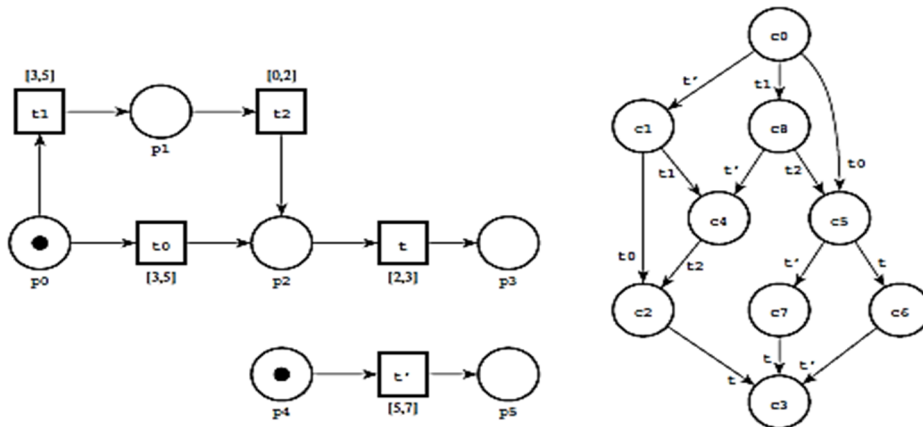


Figure21 : RdpT et son SCG

La variable $\underline{\vartheta}_t$ est notée t

Classe	marquage	Domaine de tir
C0	P0, p4	$5 \leq t' \leq 7$ $3 \leq t0 \leq 5$ $3 \leq t1 \leq 5$
C1	P0, p5	$0 \leq t0 \leq 0$ $0 \leq t1 \leq 0$
C2	P2, p5	$2 \leq t \leq 3$
C3	P3, p5	
C4	P1, p5	$0 \leq t2 \leq 2$

C5	P2, p4	$2 \leq t \leq 3$ $0 \leq t' \leq 4$
C6	P3, p4	$0 \leq t' \leq 2$
C7	P2, p5	$0 \leq t \leq 3$
C8	P1, p4	$0 \leq t' \leq 4$ $0 \leq t_2 \leq 2$

Tableau 1: Les classes, marquage et domaines de tir du graphe de figure23

4. Graphe de classes d'états fortes [22]

Le graphe connu sous le nom le graphe des classes d'états fortes ou (Strong Stat Class Graph : SSCG), préserve les états et les propriétés linéaires (LTL).

4.1 Classe d'état

4.1.1 Définition

Une classe d'états forte est un couple

$S = (M, H)$ dans laquelle :

- ▀ M , est le marquage de la classe (marquage commun à tous les états de la classe) ;
- ▀ H , un domaine d'horloges décrit par un système d'inéquations où chaque variable \bar{t}_i , est associée à une transition sensibilisée t_i qui définit le temps écoulé depuis la dernière sensibilisation de t_i . La classe initiale S_0 contient seulement l'état initial s_0 . H peut être représenté par :

$$\forall t_i \in E(M), 0 \leq \bar{t}_i \leq 0;$$

$$\forall t_j, t_i \in E(M) \wedge 0 \leq \theta, 0 \leq \text{MIN}_{e \in E} \{ \text{Eft}_e(t_i) - \bar{t}_i \} \leq \theta \leq \text{MAX}_{e \in E} \{ \text{Lft}_e(t_j) - \bar{t}_j \};$$

Où \bar{t}_j et \bar{t}_i sont des variables associées respectivement aux transitions t_j et t_i .

4.1.2 Transition de classes d'états fortes

Une transition t_i est tirable depuis la classe $S = (M, H)$, ssi :

- (i) t_i est sensibilisée par M au sens des RdP ;
- (ii) $H \wedge 0 \leq \theta \wedge 0 \leq \text{MIN}_{e \in E} \{ \text{Eft}_e(t_i) - \bar{t}_i \} \leq \theta \leq \text{MAX}_{e \in E} \{ \text{Lft}_e(t_j) - \bar{t}_j \}$; est consistant. (θ est une nouvelle variable).

Le tir de t_i conduit à la classe $S' = (M', S')$, caractérisée par :

- ▀ Le nouveau marquage $M' : \forall p \in P, M'(p) = M(p) - \text{Pre}(p, t) + \text{Post}(p, t)$;
- ▀ Le nouveau domaine d'horloge H' est obtenu selon la procédure suivante (4 étapes) :

- a) Ajouter au système H , les conditions de tir (ii) (une nouvelle variable θ est ajoutée).
- b) Supprimer dans le système obtenu les inéquations incluant les variable \bar{t}_j associées aux transitions en conflit avec t_i , ces transitions sont désensibilisées par le tir de t_i ;
- c) Dans le système ainsi réduit, effectuer le changement de variable suivant : $\forall i \neq j, \bar{t}'_j = \bar{t}_j + \theta$, et éliminer par substitution toutes les occurrences de la variable \bar{t}_j et la variable θ , pour ne garder que les nouvelles \bar{t}'_j ;
- d) Compléter ce dernier système par une variable supplémentaire pour chaque transition nouvellement sensibilisée t_i , tel que $0 \leq \bar{t}'_i \leq \theta$.

La variable θ décrit les dates de tir possibles de t_i de puis les états de la classe de départ.

A partir de cette représentation des états accessibles et de la relation de transition entre classes, on peut aborder la notion de graphe des classes d'états fortes.

☞ Exemple

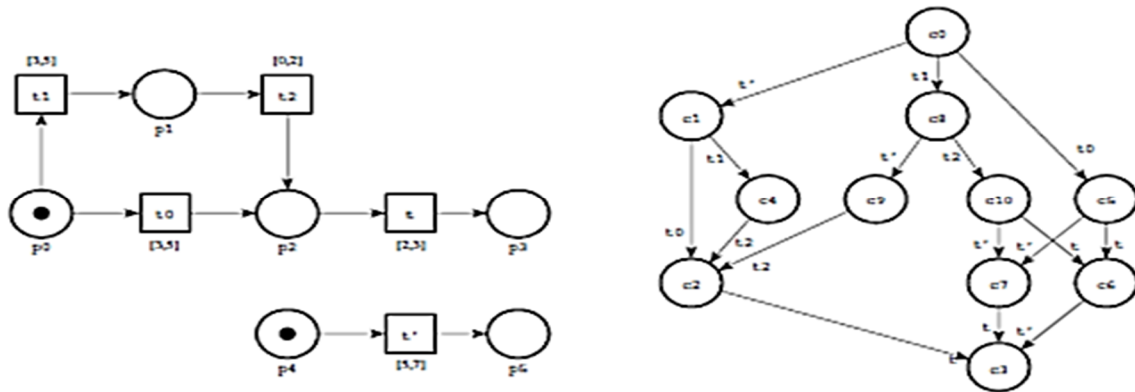


Figure 22: Un réseau de Petri et son SSCG

5. Graphes de classes d'états atomiques [22, 23]

La première construction pour un graphe de classes atomique, ou (Atomic Stat Class graph : ASCG) a été proposée par Yoneda et al [23] pour les réseaux de Petri avec intervalle statique borné supérieurement. Cependant, le graphe obtenu peut être infini même si le modèle est borné. Ultérieurement, la technique a été étendue pour tout type de RdPT dans [22] (le RdPT n'est pas forcément borné supérieurement), que nous rappelons dans cette sous-section. Les variantes de cette technique ne sont rien d'autres que celles proposées pour la technique du (SSCG) étant donné que cette méthode se repose sur la construction du graphe des classes d'états fortes d'abord. Suivant [23], nous appelons atomique une classe stable par rapport à toutes ses classes suivantes, c'est-à-dire dont chaque état a un successeur dans chacune de ses classes suivantes. Notons que les états dans une classe d'états forte (horloges) peut être distinguée une à une, alors que c'est impossible pour les états dans une classe d'état

(délai exprimant le domaine de tir). La raison principale est que le domaine de tir d'une classe d'état est une union des domaines de tir de tous ses états, et l'union est connue comme étant une opération irréversible.

Les classes fortes sont adéquates comme partition initiale (contrairement aux classes du graphe des classes). Toutefois, le fait que cet ensemble soit un recouvrement plutôt qu'une partition, implique que le résultat final sera généralement non minimal en nombre de blocs, et non unique.

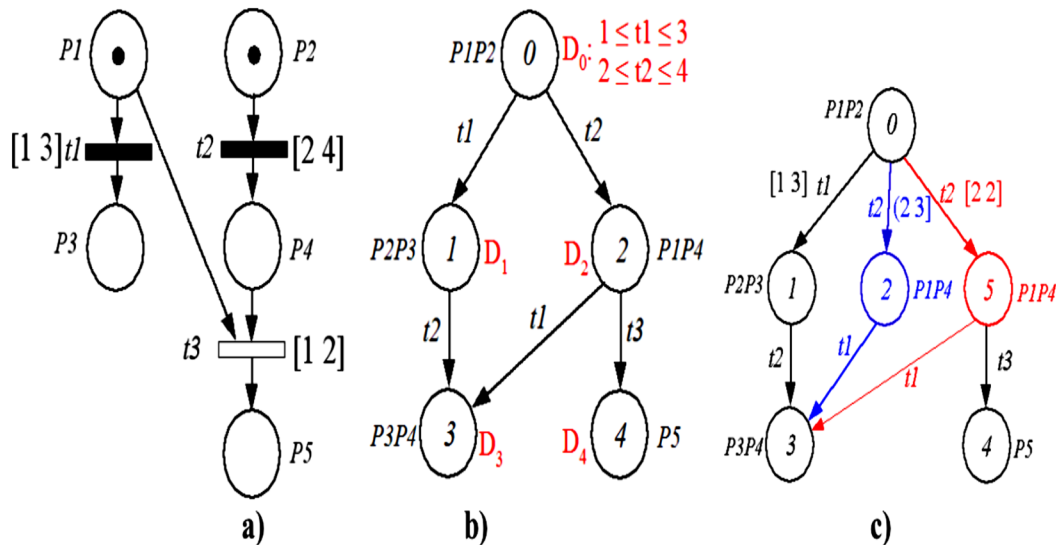


Figure 23: a)Réseau de Petri. b) graphe de classe LTL. c)graphe de classe CTL

6. La méthode des matrices de distances de Boucheneb

Nous montrons, dans ce qui suit, la méthode basée sur les matrices de distances proposée par Boucheneb et al [24]. En appliquant cette méthode, il est possible de réduire d'avantage la complexité de calcul des classes d'états précédentes [25], en évitant d'appliquer systématiquement l'algorithme de FLOYD-WARSHALL [26]. L'idée est d'exploiter les calculs déjà réalisés sur les classes précédentes pour simplifier ceux des classes successeurs. Les auteurs proposent un algorithme qui calcule, en parcourant un chemin du graphe des classes, les temps minimal et maximal d'exécution de la séquence de transitions du chemin. La complexité de l'algorithme est $O(m \times n)$, où m est la longueur du chemin et n est le plus grand nombre de transitions sensibilisées dans les classes du chemin.

6.1 Classe d'état

Soit \bullet l'origine de la classe (l'origine temporelle). $D[t, \bullet], D[\bullet, t]$ et $D[t, t']$ sont respectivement l'opposé du délai résiduel minimal de franchissement de t , le délai résiduel maximal de franchissement de t , et l'écart algébrique maximal, dans la classe E , entre les délais résiduels de franchissement des transitions t et t' . La classe d'états initiale $E_0 = (M_0, D_0)$ tel que M_0 est le marquage initial et la matrice D_0 définie sur $(E(M_0) \cup \bullet)^2$, comme suit :

$$D_0[t, \bullet] = -EFT(t) ; D_0[t, t] = 0 ; D_0[\bullet, \bullet] = 0 ; D_0[\bullet, t] = LFT(t),$$

$$D_0[t, t'] = D_0[t, \bullet] + D_0[\bullet, t'].$$

6.2 Transition d'état

Partant d'une classe $E = (M, D)$ (sous forme canonique), le calcul des transitions franchissables peut être réalisé sans utiliser l'algorithme de FLOYDWARSHALL ou tout autre algorithme de résolution de systèmes d'inéquations.

En effet, une transition t_f est franchissable à partir de la classe E uniquement si elle est sensibilisée pour le marquage M et la distance maximale qui la sépare de chaque transition sensibilisée t est positive ou nulle

$$(\forall t \in E(M), \text{Max}(t - t_f) \geq 0 \text{ ou encore } D[t_f, t] \geq 0).$$

Le tir de t_f conduit à la classe $E' = (M', D')$ suivante, caractérisée par :

▀ Le nouveau marquage $M' : \forall p \in P, M'(p) = M(p) - \text{Pre}(p, t_f) + \text{Post}(p, t_f) ;$

▀ La matrice D' est obtenue selon la procédure suivante :

a. $D[\bullet, \bullet] = 0, \forall t \in E(M) D_0[t, t] = 0$

b. $\forall t \in E'(M) ;$

▀ Si t Persistante :

$$D'[\bullet, t] = D[t_f, t] \quad D'[t, \bullet] = \text{Min} \{D[t, t']\}; (\forall t' \in E(M))$$

▀ Si t nouvellement sensibilisée :

$$D'[\bullet, t] = LFT(t) \quad D'[t, \bullet] = -EFL(t)$$

c. $\forall (t, t') \in (E'(M))^2 \wedge t \neq t' ;$

▀ Si (t, t') persistantes :

$$D'[t, t'] = \text{Min} (D[t, t'], D'[t, \bullet] + D'[\bullet, t'])$$

▀ Si (t, t') nouvellement sensibilisées :

$$D'[t, t'] = D'[t, \bullet] + D'[\bullet, t'].$$

6.3 Construction du graphe des classes d'états

La construction du graphe des classes d'états est réalisée en appliquant la règle de franchissement établie précédemment. Le graphe obtenu est semblable à celui du SCG plus haut à la différence cette approche réduit la complexité en temps qui passe de $O(n^3)$ à $O(n^2)$. Les auteurs ont ensuite développé un algorithme qui calcule, en explorant un chemin du graphe des classes, l'intervalle de franchissement de la séquence du chemin. La complexité de cet algorithme est de $O(m \times n)$, où, est la longueur du chemin et n est le nombre de transitions sensibilisées dans la plus grande classe du chemin (en nombre de transitions sensibilisées).

☞ **Exemple :**

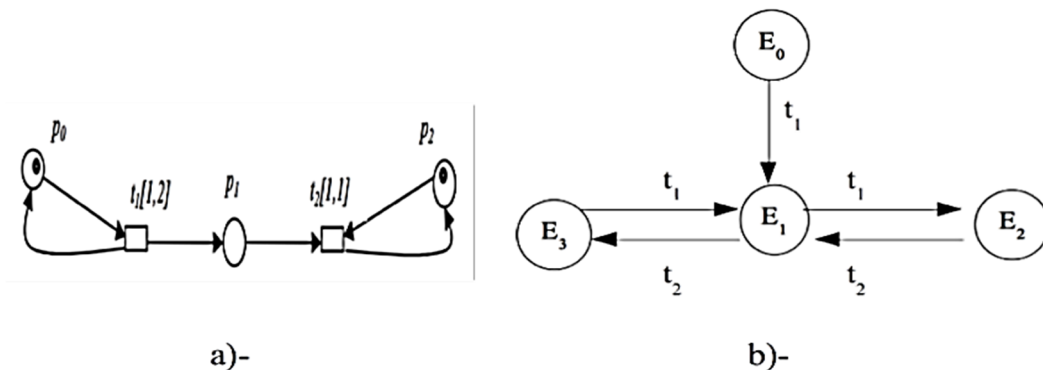


Figure24 : a) RDP t-temporel b) Son graphe de classes

Le graphe a quatre classes d'états et cinq transitions d'états.

La classe initiale $E_0 = (M_0, D_0)$ avec : $M_0: \{p_0, p_2\} \rightarrow 1$.

$$D_0 : \begin{array}{c|c|c} & \bullet & t_1 \\ \hline \bullet & 0 & 2 \\ \hline t_1 & -1 & 0 \end{array}$$

- Le franchissement de t_1 depuis E_0 conduit à la classe $E_1 = (M_1, D_1)$, avec :

$M_1 : \{p_0, p_1, p_2\} \rightarrow 1$

$$D_1 : \begin{array}{c|c|c|c} & \bullet & t_1 & t_2 \\ \hline \bullet & 0 & 2 & 1 \\ \hline t_1 & -1 & 0 & 0 \\ \hline t_2 & -1 & 1 & 0 \end{array}$$

➡ Vérifier si la transition t_1 est franchissable ? t_1 est franchissable car $D_1[t_1, t_2] \geq 0$. La classe successeur E_2 de E_1 par la transition t_1 a aussi deux transitions sensibilisées, i.e. : $E(M_2)$

$=t_1, t_2$. La transition t_1 est nouvellement sensibilisée alors que la transition t_2 est persistante. La matrice des distances D_2 de la classe E_2 est calculée comme suit :

$$D_2[\bullet, t_2] = D_1[t_1, t_2] = 0 ; \quad D_2[t_2, \bullet] = \text{MIN}(D_1[t_2, t_2], D_1[t_2, t_1]) = 0$$

$$D_2[\bullet, t_1] = \text{LFT}(t_1) = 2 ; \quad D_2[t_1, \bullet] = -\text{EFT}(t_1) = -1$$

$$D_2[t_1, t_2] = D_1[\bullet, t_2] + D_1[t_1, \bullet] = -1 ; \quad D_2[t_2, t_1] = D_1[\bullet, t_1] + D_1[t_2, \bullet] = 2.$$

$E_2 = (M_2, D_2)$, tel que :

$$M_2: \{p_0, 2 p_1, p_2\} \rightarrow 1$$

$$D_2: \begin{array}{c|c|c|c} & \bullet & t_1 & t_2 \\ \hline \bullet & 0 & 2 & 0 \\ \hline t_1 & -1 & 0 & -1 \\ \hline t_2 & 0 & 2 & 0 \end{array}$$

➤ On continue de la même façon pour obtenir la classe E_3 :

$E_3 = (M_3, D_3)$, tel que :

$$M_3: \{p_0, p_1, p_2\} \rightarrow 1$$

$$D_3: \begin{array}{c|c|c} & \bullet & t_1 \\ \hline \bullet & 0 & 1 \\ \hline t_1 & 0 & 0 \end{array}$$

Tableau comparatif

Construction/graphes	Propriétés	Les variable / système	taille et complexité	Classe initiale
Construction standard : SCG	marquage et <i>LTL</i>	Délai (dates de début et de fin de tir)	$O(n^3)$ pour le graphe et $O(m \times n^3)$ pour les chemins	$M_0 \wedge EFT(t) \preceq \underline{t} \preceq LFT(t)(\forall t \in M_0)$
Construction compacte : SCG	marquage	Délai (dates de début et de fin de tir)	temps de calcul et tailles de graphe réduites par rapport à SCG	$M_0 \wedge EFT(t) \preceq \underline{t} \preceq LFT(t)(\forall t \in M_0)$
Construction compacte : CSCG	marquage et (<i>LTL</i> ? selon des conditions)	Délai (dates de début et de fin de tir)	temps de calcul et tailles de graphe réduites par rapport à SCG	$M_0 \wedge EFT(t) \preceq \underline{t} \preceq LFT(t)(\forall t \in M_0)$
Construction relaxée : RSCG	marquage	Délai relaxés	temps de calcul et tailles de graphe réduites par rapport à SCG et SCG compacte	$M_0 \wedge EFT(t) \preceq \underline{t} \preceq LFT(t)(\forall t \in M_0)$
Construction : SSCG	états et <i>LTL</i>	Valuations d'horloge (le temps écoulé depuis la dernière sensibilisation de la transition)	temps de calcul et tailles de graphe plus important par rapport à SCG	$M_0 \wedge 0 \preceq \bar{t} \preceq 0(\forall t \in M_0)$
Construction par inclusion SSCG	états	Valuation d'horloge	temps de calcul et tailles de graphe réduite par rapport à SSCG	$M_0 \wedge 0 \preceq \bar{t} \preceq 0(\forall t \in M_0)$
Construction relaxée : RSSCG	états	Valuation d'horloge relaxés	temps de calcul et tailles de graphe réduite par rapport à SSCG	$M_0 \wedge 0 \preceq \bar{t} \preceq 0(\forall t \in M_0)$
Construction : ASCG	états et branchement <i>CTL</i>	Valuation d'horloge	temps de calcul et tailles de graphe plus important par rapport à SSCG	$M_0 \wedge 0 \preceq \bar{t} \preceq 0(\forall t \in M_0)$
Construction basée distance	marquage et <i>LTL</i>	distances de temps	$o(n^2)$ pour le graphe et $O(m \times n)$ pour les chemins	$M_0 \wedge EFT(t) \preceq \underline{t} \preceq LFT(t)(\forall t \in M_0)$

Tableau 2 : Comparaison entre les différents graphes des classes

7. Conclusion

Les méthodes exposées dans ce chapitre permettant de calculer l'espace d'états de réseau de Petri temporel RdPT, abstraire son comportement, disposer d'une représentation finie de cet espace infini d'états permet de mettre en œuvre les techniques de vérification de modèles.

L'ensemble des techniques présenté est supporté par l'environnement **Tina** qui offre des fonctionnalités d'édition, d'exploration et d'analyse de réseaux de Petri temporels.



Chapitre – V -

Cas d'étude Protocole du bit alterné (ABP)

1. Introduction

Afin d'échanger des données de manière structurée au sein d'un réseau, il faut avoir recours à des règles qui commandent le déroulement des communications : les protocoles.

On distingue généralement deux grands types de protocoles : les protocoles routables et les protocoles non routables. Dans les sections suivantes, nous allons étudier un protocole qui s'appelle bit alterné.

Pour commencer, nous allons citer quelques caractéristiques de protocoles *TCP/IP*. Après nous présenterons le protocole du bit alterné.

2. Le protocole *TCP/IP*

La famille de protocoles *Transmission Control Protocol* et *Internet Protocol* communément appelée pile de protocoles *TCP/IP* autorise l'échange de données en milieu hétérogène.

IP est un protocole routable autorisant une communication en mode connecté au travers de *TCP*. Un fonctionnement en mode déconnecté est également possible. Est alors utilisé le protocole *UDP* qui fait partie de la pile des protocoles *TCP/IP*. [28].

TCP/IP est actuellement un standard de l'industrie, évidemment en raison de son exploitation mondiale par Internet, mais aussi par son utilisation dans des réseaux d'entreprise de type Windows ou Novell.

TCP/IP est un jeu de protocoles dérivé du projet *ARPANet* du ministère de la Défense américain, pendant les années 60 et 70.

2.1 Les caractéristiques de protocoles *TCP/IP*

Les caractéristiques intéressantes du jeu de protocoles *TCP/IP* sont :

- L'indépendance des fabricants, ce qui n'est pas le cas de tous les jeux de protocoles.
- Presque tout système peut s'intégrer au réseau par *TCP/IP*.
- L'utilisable tant dans un LAN que dans un WAN.
- Le fantastique essor d'Internet l'a élevé au rang de pile des protocoles la plus utilisée. [29]

3. Le Modèle OSI

Au début des années 70, chaque constructeur a développé sa propre solution réseau autour d'architecture et de protocoles privés ; et on s'est rendu compte qu'il serait impossible d'interconnecter ces différents réseaux si une norme internationale n'était pas adoptée. Cette norme établie par l'International Standard Organisation (ISO) est la norme Open System Interconnexion (OSI).

Un système ouvert est un ordinateur, un terminal, un réseau, n'importe quel équipement respectant cette norme et donc apte à échanger des informations avec d'autres équipements hétérogènes et issus de constructeurs différents.

Le premier objectif de la norme OSI a été de définir un modèle de toute architecture de réseau basé sur un découpage en sept couches. Chacune de ces couches correspondant à une fonctionnalité particulière d'un réseau. Les couches 1, 2, 3 et 4 sont dites basses et les couches 5, 6 et 7 sont dites hautes. Chaque couche est constituée d'éléments matériels et logiciels et offre un service à la couche située immédiatement au-dessus d'elle en lui épargnant les détails d'implémentation nécessaires. Chaque couche du rang n d'une machine gère la communication avec la couche respective d'une autre machine en suivant un protocole propre du rang qui est un ensemble de règles de communication pour le service de niveau n comme [30], [31].

Modèle OSI		Modèle TCP/IP	
Application	Couches application	Application	Protocoles d'application
Présentation			
Session			
Transport	Couches de flux de données	Transport	Protocoles de transport
Réseau		Réseau	Protocoles réseau
Liaison de données		Accès réseau	
Physique			

Figure25: Le protocole TCP/IP et les couches OSI

4. Le protocole du bit alterné

Les protocoles de communication font un large usage de contraintes temporelles:

Les mécanismes de reconfiguration pour la perte de messages, par exemple, sont typiquement implantés à l'aide de temporisations. Le Protocole du Bit Alterné est certainement le plus simple de ces protocoles. Il s'agit d'un protocole de transfert de données du type envoi-attente : avant d'émettre à nouveau un message, le processus émetteur attend l'arrivée de l'accusé de réception du message qu'il a précédemment envoyé.

Les hypothèses sur le fonctionnement du médium de communication sont que les messages ou accusés de réception peuvent être perdus ou endommagés (dans ce dernier cas, ils sont simplement rejetés). Pour corriger les pertes de messages et/ou d'accusés, une

temporisation est lancée lorsqu'un message est émis. Si l'accusé de réception du message n'arrive pas avant l'expiration de la temporisation, le message est retransmis.[33] [34]

Le protocole du bit alterné peut être représenté par le réseau temporel de la *figure 39*. Par souci de simplification, les messages endommagés sont assimilés à des messages perdus. Notons que, dans ce réseau, les pertes de messages ou d'accusés sont simplement représentées par des transitions qui n'ont pas de place de sortie. Afin de reproduire la spécification, des estimations pour la durée de toutes les opérations élémentaires du protocole doivent être fournies. La temporisation de retransmission doit être choisie suffisamment longue pour que le médium de transmission ne contienne au plus qu'un message ou accusé ; elle se produira au bout d'un temps compris entre 5 et 6 unités de temps après émission de sa dernière copie. Des estimations identiques sont données pour la perte et la réception de messages ou d'accusés de réception (entre 0 et 1 unité).

Par contre, aucune contrainte de date n'est donnée pour l'émission de la première copie de chaque message ; les transitions correspondantes portent implicitement l'intervalle $[0, \infty [$.

Ce réseau est borné. Son graphe de classes, construit par Tina en utilisant l'algorithme 1 admet seize classes d'états.

4.1 Avantages

- Il permet d'éliminer les mauvais fonctionnements liés aux pertes et aux duplications de messages.
- Numérotation des messages.
- émetteur sait si le message est arrivé et réémission possible
- Est un protocole qui accepte une séquence de paquets de données et les remet à l'autre côté dans le même ordre.

4.2 Inconvénients

- Dans certaines situations, le protocole de bit alterné ne permet pas de récupérer les trames perdues, c'est pour cela que ce protocole est aujourd'hui remplacé par des protocoles plus efficaces, mais aussi plus complexes.
- N'autorise que la transmission d'une seule trame à la fois.
- Un faible taux d'utilisation du canal.
- La taille de la mémoire des nœuds est là encore infinie ou le protocole est probabiliste.
- Le canal peut perdre des données.

4.3 Principe de fonctionnement du BA

1. Communication unidirectionnelle
2. "Envoyer et attendre" (... un accusé de réception)
3. Si le message de ACK (M) est perdu, on va retransmettre M alors qu'il a été correctement reçu => duplication => numéro de séquence seq
4. Combien de bits pour coder seq?
5. Ce problème se pose seulement entre un message et le suivant, pas entre le prédécesseur et le suivant [35]

Le protocole de bit alterné est implémenté au niveau de la couche de "liaison de données" du modèle OSI (couche n°2), il ne concerne donc pas les paquets, mais les trames (on parle de paquets uniquement à partir de la couche "Réseau" (couche 3) du modèle OSI).

La règle est relativement simple : la première trame envoyée par A aura pour drapeau 0, dès cette trame reçue par B, ce dernier va envoyer un accusé de réception avec le drapeau 1 (ce 1 signifie "la prochaine trame que A va m'envoyer devra avoir son drapeau à 1"). Dès que A reçoit l'accusé de réception avec le drapeau à 1, il envoie la 2e trame avec un drapeau à 1, et ainsi de suite... [36]

Les deux côtés doivent initialiser une connexion, ce qui permet d'initialiser le comptage, étant donné qu'il ne faut pas gaspiller les bits pour le comptage, essayons avec le plus petit compteur possible: 1 bit

☞ Remarque

Ce protocole est aussi connu sous les termes « protocole stop-and-wait »

☞ Exemple

☛ Bakhta veut envoyer à Djeloul un message M, qu'elle a prédécoupé en sous-messages M0, M1, M2,...

☛ Bakhta envoie ses sous-messages à une cadence Δt fixée (en pratique, les sous-messages partent quand leur acquittement a été reçu ou qu'on a attendu celui-ci trop longtemps : on parle alors de timeout).

Situation idéale

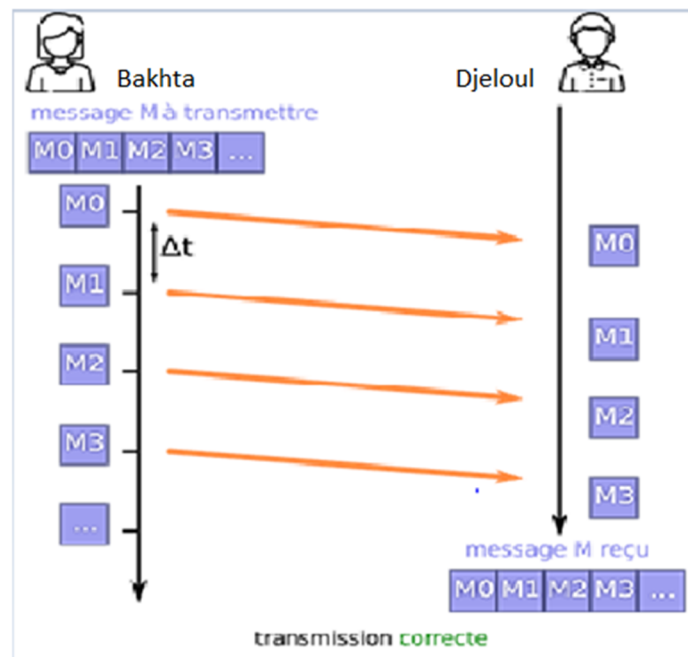


Figure 26 : Exemple bit alterné Situation idéale

Dans cette situation, les sous-messages arrivent tous à destination dans le bon ordre. La transmission est correcte.

- Chaque message contient son numéro, afin que le récepteur sache quel message il vient de recevoir
- Besoin de connexion: les deux côtés doivent initialiser une connexion, ce qui permet d'initialiser le comptage
- Étant donné qu'il ne faut pas gaspiller les bits pour le comptage, essayons avec le plus petit compteur possible: 1 bit.

Situation réelle

Mais parfois, les choses ne se passent pas toujours aussi bien. Car si on maîtrise parfaitement le timing de l'envoi des sous-messages de Bakhta, on ne sait pas combien de temps vont mettre ces sous-messages pour arriver, ni même (attention je vais passer dans un tunnel) s'ils ne vont pas être détruits en route.

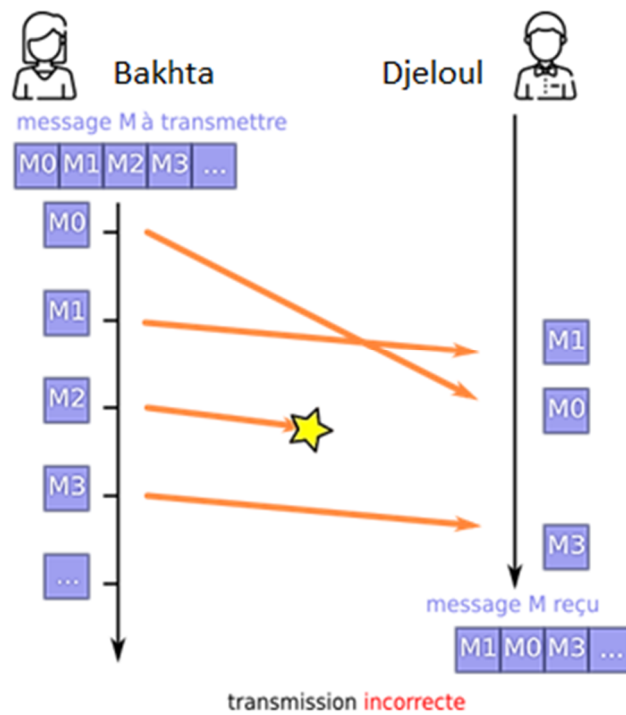


Figure 27: Exemple bit alterné Situation réelle

Le sous-message M0 est arrivé après le M1, le message M2 n'est jamais arrivé...

Solution naïve

Pourquoi ne pas demander à Djeloul d'envoyer un signal pour dire à Bakhta qu'il vient bien de recevoir son sous-message ? Nous appellerons ce signal **ACK** (comme *acknowledgement*, traduisible par «accusé de réception»). Ce signal ACK permettra à Bakhta de renvoyer un message qu'elle considérera comme perdu :

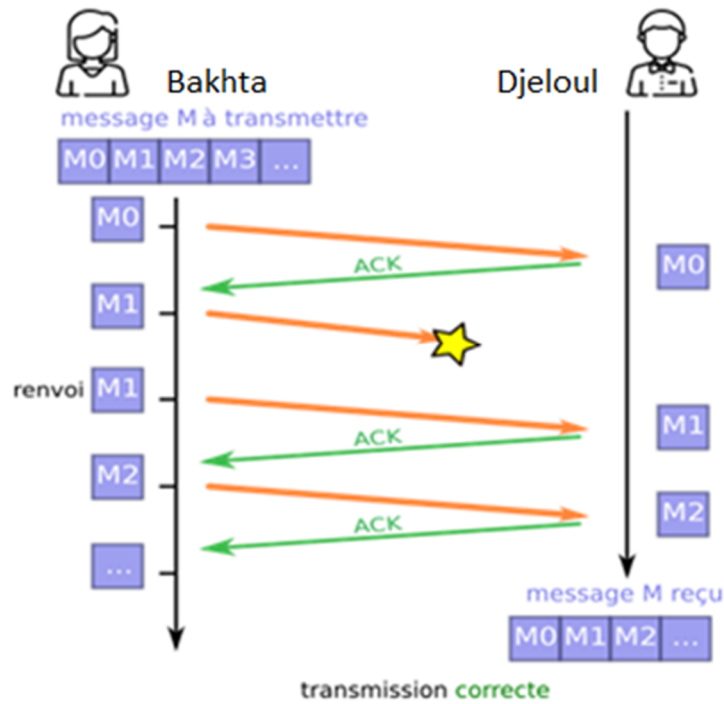


Figure 28: Exemple bit alterné Situation naïve

N'ayant pas reçu le ACK consécutif à son message M1, Bakhta suppose (avec raison) que ce message n'est pas parvenu jusqu'à Djeloul, et donc renvoie le message M1. **Mais peu efficace...**

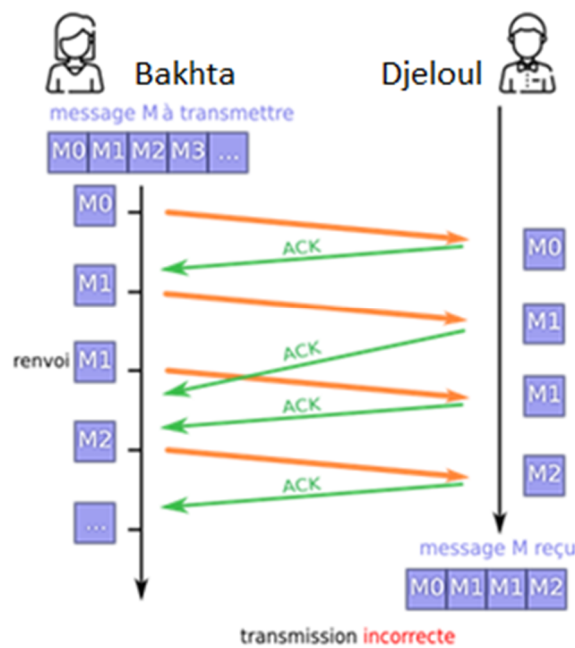


Figure 29: Exemple bit alterné Situation pas efficace

Le deuxième ACK de Djeloul a mis trop de temps pour arriver (ou s'est perdu en route) et donc Bakhta a supposé que son sous-message M1 n'était pas arrivé.

Elle l'a donc renvoyé, et Djeloul se retrouve avec deux fois le sous-message M1. La transmission est incorrecte. En faisant transiter un message entre Djeloul et Bakhta, nous multiplions par 2 la probabilité que des problèmes techniques de transmission interviennent. Et pour l'instant rien ne nous permet de les détecter.

☒ Remarque

- ▀ Le protocole BA fut le 1er protocole qui fut spécifié en utilisant la notion de modèle de transitions d'état [27]
- ▀ BA est un des plus simples protocoles de liaison données possible, mais il
 - a) Démontre les principes fondamentaux de tous les protocoles de liaison de données
 - b) Réussit à récupérer des erreurs de transmission dans certaines limites, nous verrons...
- ▀ Il a été utilisé dans un grand nombre d'études sur la validation des protocoles
- ▀ Exigence: un protocole qui accepte une séquence de paquets de données et les remet à l'autre côté dans le même ordre.

5. Conclusion

En guise de conclusion, nous devons retenir les points suivants :

TCP/IP est une suite de protocoles qu'on peut qualifier de « Super Star »!

L'avantage de TCP permet des communications fiables.

L'inconvénient est qu'il nécessite une négociation, ce qui prend du temps.

Le protocole du bit alterné a longtemps été utilisé au sein de la couche 2 du modèle OSI (distribution des trames Ethernet). Simple et léger, il peut toutefois être facilement mis en défaut, ce qui explique qu'il ait été remplacé par des protocoles plus performants.

On utilise l'outil Tina dans le chapitre suivant pour obtenir les graphes SCG, SSCG et ASCG de protocole BA.



Chapitre –V -

Cas d'étude : Analyse et vérification du protocole Bit Alterné

1. Introduction

Dans ce chapitre nous soulignerons quelques propriétés du protocole Bit Alterné par utilisation des réseaux de Petri temporel et un logiciel très puissant et populaire qui peut traiter pas mal d'extensions des Rdp.

L'outil Tina propose un certain nombre de constructions de graphes de comportements pour les réseaux de Petri et les réseaux temporels. Différentes techniques sont utilisées pour extraire des vues du comportement du réseau préservant certaines classes de propriétés de l'ensemble des états.

2. L'outil Tina

Dans cette partie nous introduisons l'outil Tina un environnement logiciel permettant l'édition et l'analyse de réseaux de Petri et réseaux de Petri temporels ses avantages :

- Entrée graphique ou textuelle.
- Génère graphes de comportements.
- Préservant familles de propriétés.
- Détection propriété borne, vivacité.
- Sortie en clair ou formats dédiés.

2.1. Fonctionnalités générales

La boîte à outils Tina inclut les outils suivants :

-Nd (NetDraw) : Un éditeur textuel et graphique pour réseaux de Petri, réseaux de Petri temporels et automates.

-Tina : Construction d'espaces d'états. Les réseaux d'entrée peuvent être sous forme textuelle ou graphique. Les graphes produits peuvent l'être sous forme graphique ou textuelle. Différents formats de sortie sont disponibles pour permettre la connexion à d'autres model-checkers ou equivalence-checkers. Suivant les options, l'outil construit:

- Le graphe de couverture d'un réseau de Petri (technique de Karp et de Miller).
- Le graphe des marquages d'un réseau borné (le caractère borné étant vérifié à la volée).
- Des graphes partiels utilisant différentes techniques d'exploration "ordre partiel".
- Diverses abstractions de l'espace d'états pour des réseaux de Petri temporels (graphes de classes d'états). Selon l'option choisie, les abstractions obtenues préservent les marquages, les états, les propriétés LTL, ou les propriétés de CTL.

2.2. Interface avec Tina

La boîte à outil Tina [21] est un logiciel *open source* qui génère différents graphes de classes d'état d'un réseau de petri t-temporel sous la forme d'un automate avec deux types de sortie : textuel (.aut) et graphique (.adr). Le format d'entrée de Tina peut être textuel (.net) et graphique (.ndr).

L'interface graphique Tina permet de dessiner un RdPT comme celui montré sur la *figure 39* sous la forme d'un fichier .ndr. Tina permet de générer à partir de ce fichier, un fichier texte .net présenté dans la *figure 40*.

Lorsque le graphe de classes est construit dans l'outil **GraphC**, il génère directement un fichier texte .aut dans le format textuel d'un automate dans Tina. Tina permet de générer, à partir d'un fichier (.aut), un fichier .adr qui permet de dessiner un graphe. Un fragment du fichier .aut

2.3. Graphe de classes sous Tina

2.3.1. Préservation des propriétés LTL, classes d'états linéaires

2.3.1.1. Classes d'états linéaires, mode W

La première construction fournie par Tina concernant les réseaux temporels est celle, classique, introduite dans [32]. Elle peut être expliquée comme suit.

Pour toute séquence de tire δ , soit C_δ l'ensemble des états accessibles par des échéanciers de support δ . Considérons la relation d'équivalence \cong satisfaite par deux de ces ensembles d'état. Si ceux-ci ont même marquage et même domaine de tir, le marquage d'un ensemble état défini comme celui de ses états, et son domaine de tir comme la réunion des domaines de tirs des états le constituant.

Le graphe mode W préserve les propriétés du graphe d'états exprimables en logiques temporelles à temps linéaire (comme LTL), d'où son nom. Le graphe de classes en mode W du RdPT.

2.3.1.2. Classes d'états linéaires fortes, mode S

Une deuxième construction préservant aussi les propriétés LTL, est proposée comme précédemment, soit C_δ l'ensemble des états accessibles par des échéanciers de support δ . Les classes d'états linéaires fortes (mode S) sont exactement ces ensembles C_δ , pour toute δ tirable, c'est-à-dire considérés sans autre relation d'équivalence que l'égalité d'ensembles d'états.

Une construction du mode S est proposée dans [32]. Les classes linéaire fortes sont représentables par un marquage associé à un système d'inéquation en horloges.

2.4. Les interfaces de Tina

2.4.1. L'interface officielle

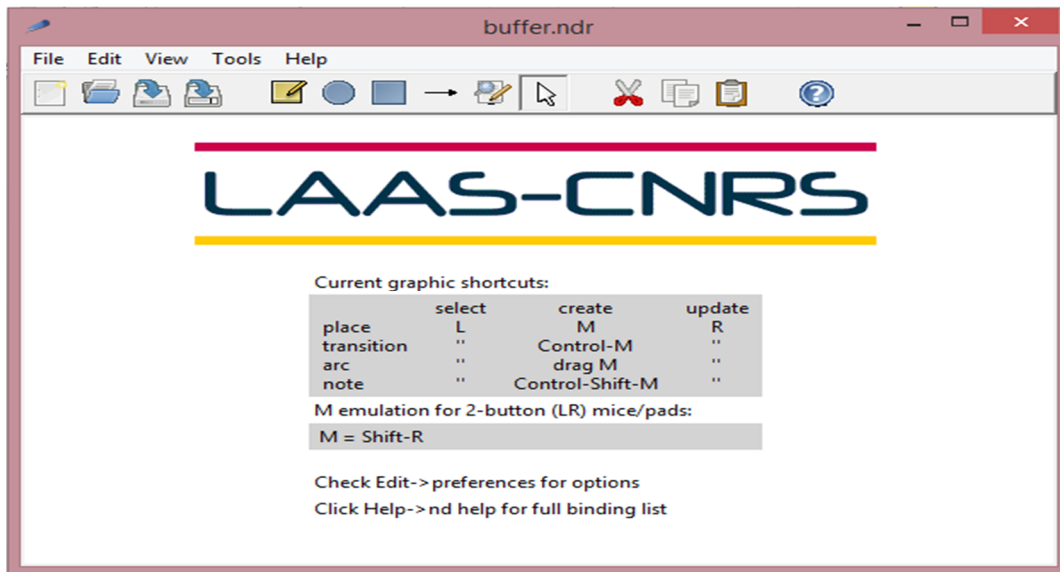


Figure 30: La page d'accueil de TINA

2.4.2. Les extensions de TINA

Dans notre outil il existe deux extensions d'entrée et deux de sortie :

2.4.2.1. Les entrées

En entrée on a (.ndr) pour le graphe et (.net) pour le texte

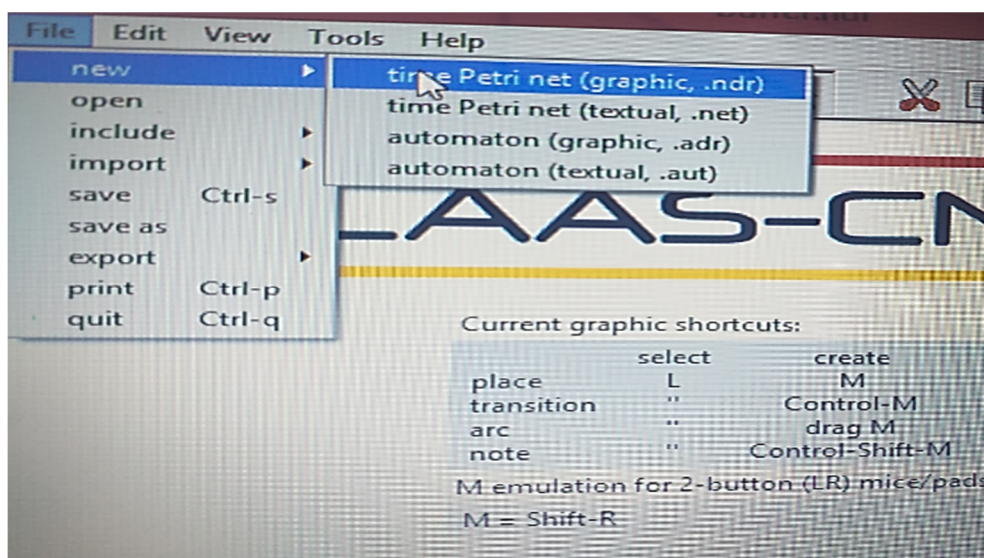


Figure 31 : Création graphique d'un réseau (.ndr)

Exemple :

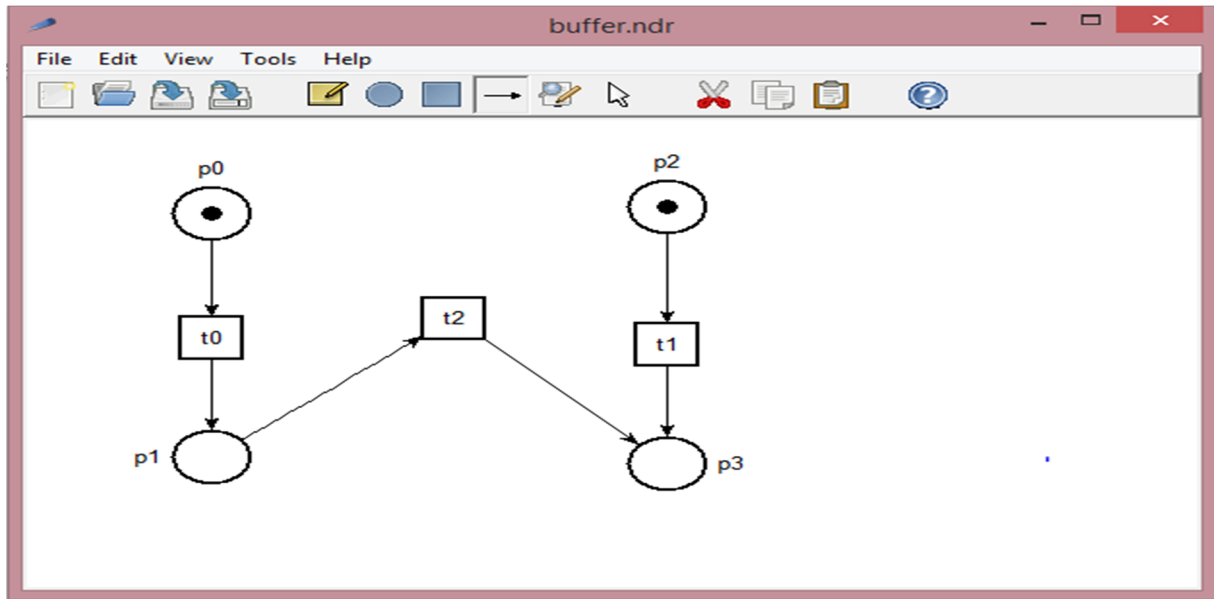


Figure 32: Exp. fichier .ndr (PN mode graphique)

Création textuel d'un réseau

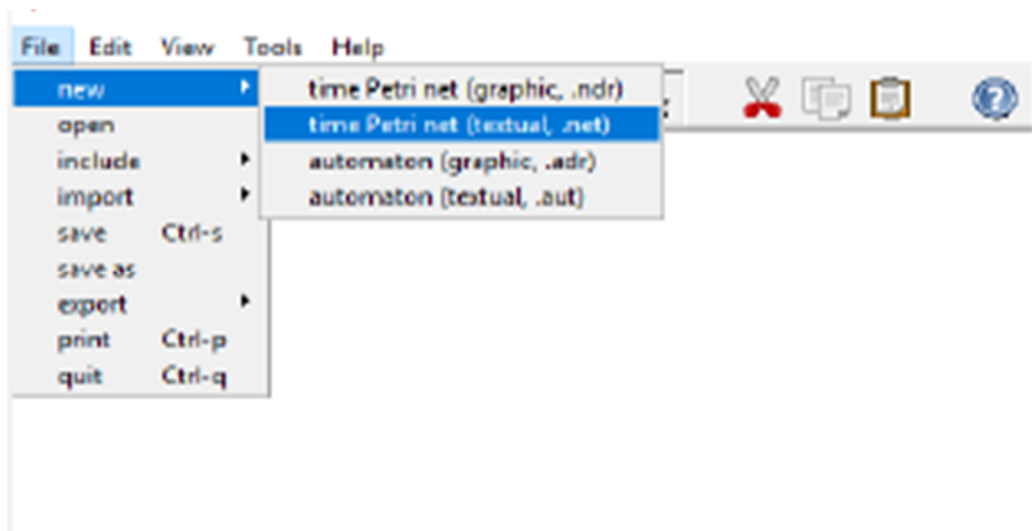
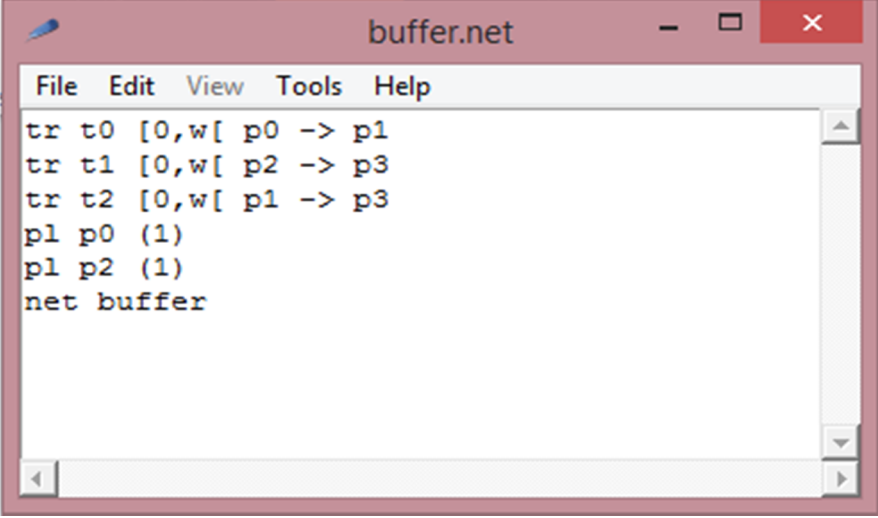


Figure 33 : Création textuelle d'un réseau (.net)

☒ Exemple :



```

File Edit View Tools Help
tr t0 [0,w[ p0 -> p1
tr t1 [0,w[ p2 -> p3
tr t2 [0,w[ p1 -> p3
p1 p0 (1)
p1 p2 (1)
net buffer
  
```

Figure 34: Exp fichier .net (PN mode texte)

On peut traduire le graphe en texte, on obtient le même réseau (.net)

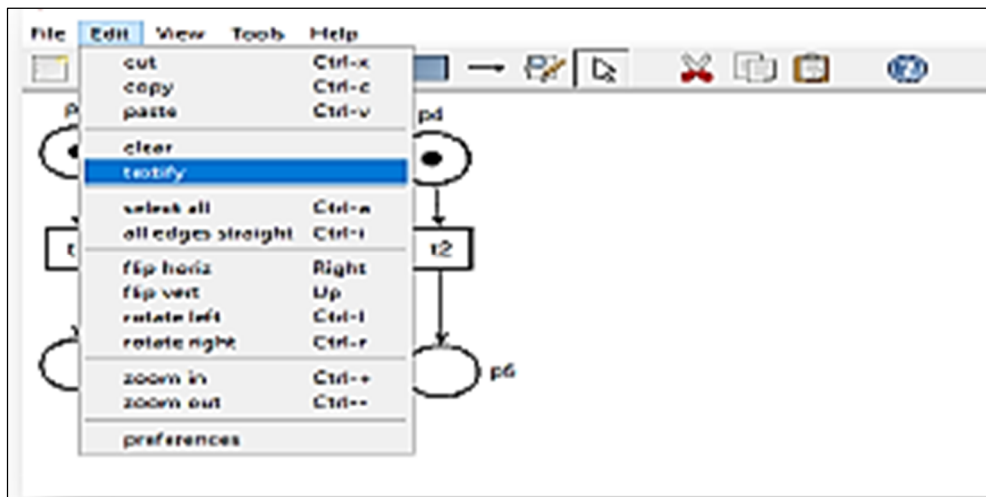


Figure 35: Traduction du graphe en texte

Comme on peut aussi traduire le texte en graphe, le résultat est le même réseau graphique (.ndr).

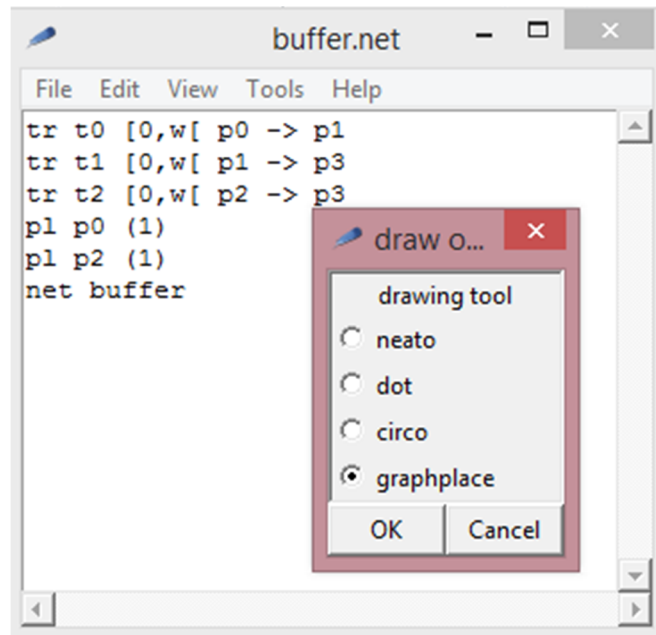


Figure 36: création de graphe depuis un texte

2.4.2.2. Les sorties

En sortie, on a (.adr) pour le graphe et (.aut) pour le texte

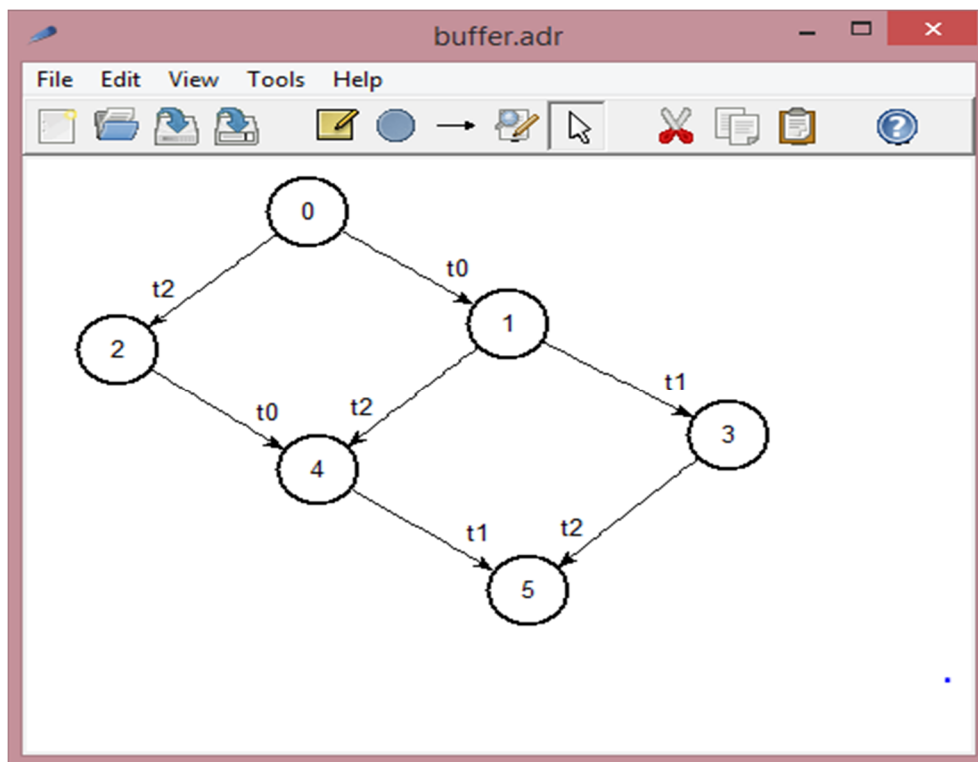


Figure 37 : La sortie graphique (.adr)

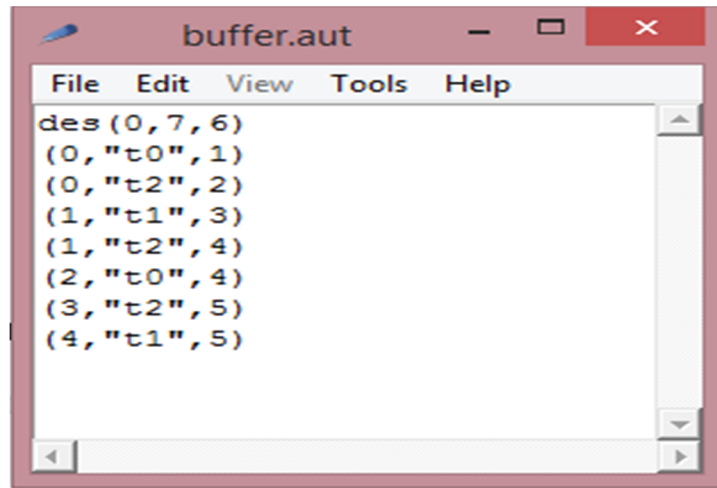


Figure 38 : La sortie textuelle (.aut)

3. Expérimentation

Nous passons à notre cas d'étude le protocole bit alterné bien étudié au chapitre précédent, La figure suivante représente le réseau de petri temporel de ce dernier.

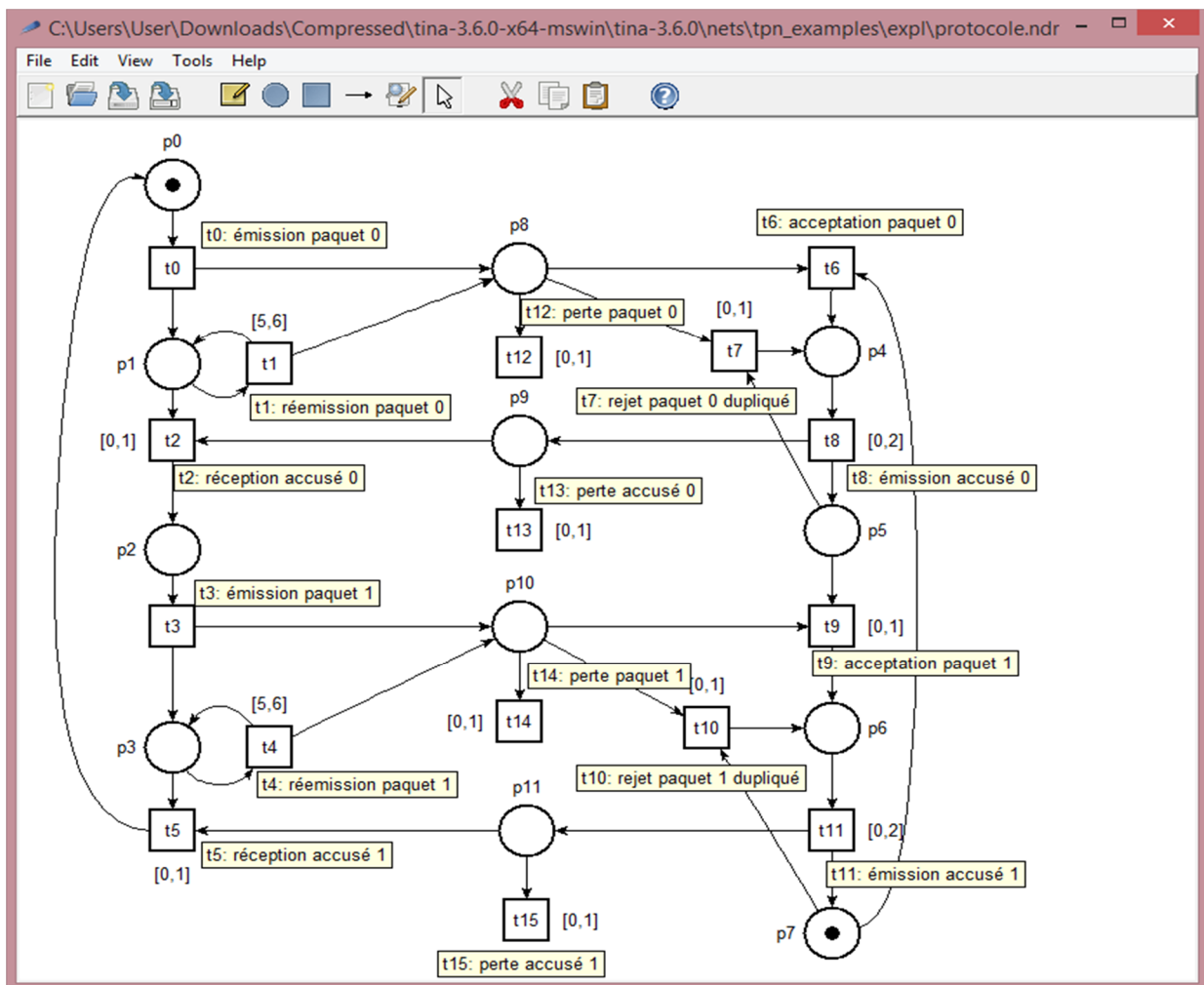


Figure 39: Un réseau temporel pour le Protocole du Bit Alterné(.ndr).

Place:

P1 : Emetteur prêt émettre

P2 : Emetteur en attente Ack

P9 : Message en transit

P5 : Récepteur prêt à recevoir message bit 0

P6 : Récepteur prêt à envoyer Ack

P7 : Récepteur prêt à recevoir message bit 0 et bit 1

La deuxième sortie du protocole sous Rdp est le texte de la figure suivante

```

File Edit View Tools Help
tr t0 [0,w[ p0 -> p1 p8
tr t15 [0,1] p11 ->
tr t14 [0,1] p10 ->
tr t2 [0,1] p1 p9 -> p2
tr t3 [0,w[ p2 -> p3 p10
tr t4 [5,6] p3 -> p3 p10
tr t1 [5,6] p1 -> p8 p1
tr t12 [0,1] p8 ->
tr t7 [0,1] p8 p5 -> p4
tr t8 [0,2] p4 -> p9 p5
tr t9 [0,1] p10 p5 -> p6
tr t10 [0,1] p7 p10 -> p6
tr t11 [0,2] p6 -> p11 p7
tr t5 [0,1] p3 p11 -> p0
tr t6 [0,w[ p8 p7 -> p4
tr t13 [0,1] p9 ->
p1 p0 (1)
p1 p7 (1)
an n0 1 {t0: émission paquet 0}
an n1 1 {t1: réémission paquet 0}
an n2 1 {t2: réception accusé 0}
an n3 1 {t3: émission paquet 1}
an n4 1 {t4: réémission paquet 1}
an n5 1 {t5: réception accusé 1}
an n6 1 {t6: acceptation paquet 0}
an n8 1 {t7: rejet paquet 0 dupliqué}
an n7 1 {t8: émission accusé 0}
an n9 1 {t9: acceptation paquet 1}
an n10 1 {t11: émission accusé 1}
an n11 1 {t10: rejet paquet 1 dupliqué}
an n12 1 {t12: perte paquet 0}
an n13 1 {t13: perte accusé 0}
an n14 1 {t14: perte paquet 1}
an n15 1 {t15: perte accusé 1}
net protocole

```

Figure 40:Le texte correspond au graphe de BA (.net)

Avec l'outil Tina, On va suivre les étapes indiquées dans la figure suivante pour obtenir les graphes SCG (mode W), SSCG (mode -S), ASCG (mode -A), Graphe de marquage (mode, Graphe préservant seulement les états et un graphe d'atomisé

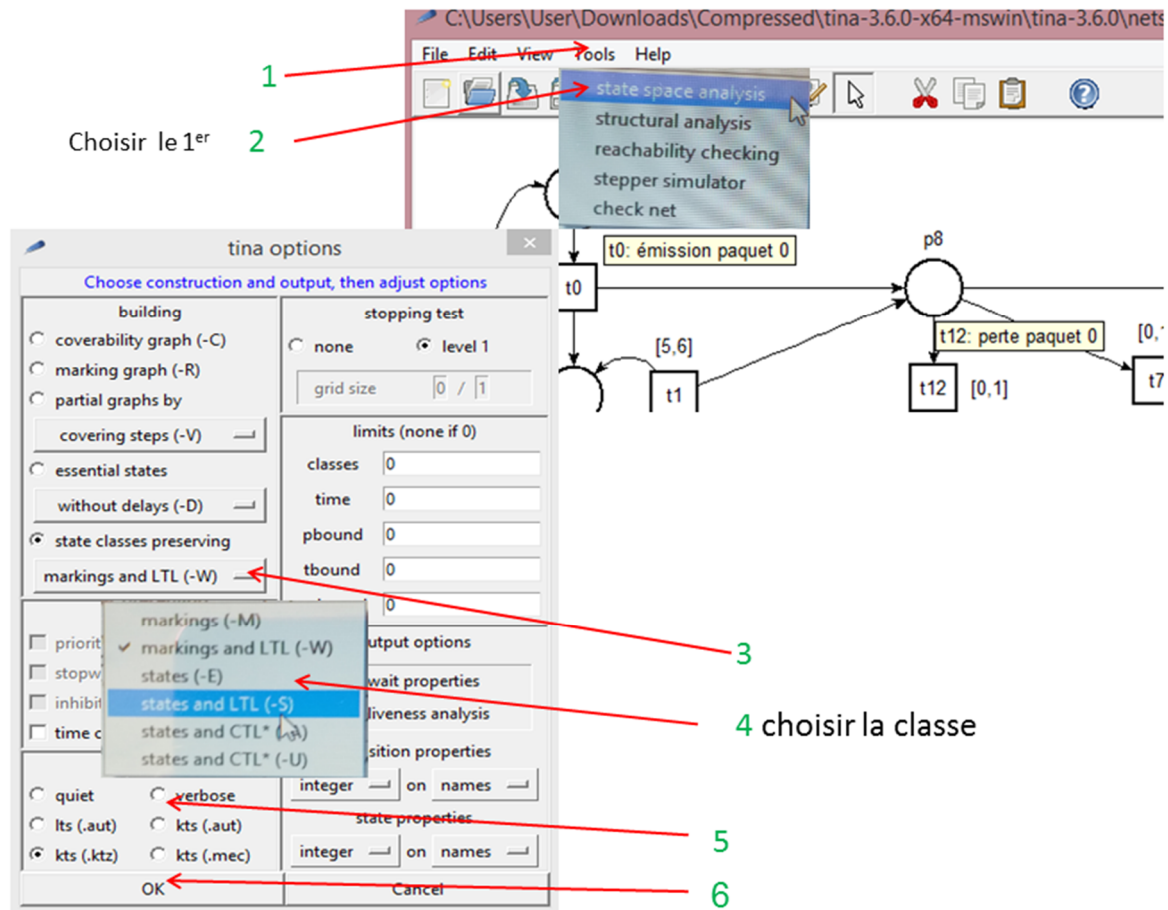


Figure 41 : Les étapes d'analyse fondamentale de l'outil Tina

Les deux figures suivantes représentent le SCG de notre protocole :

digest	places 12	transitions 16	net	bounded Y	live ?	reversible ?
	abstraction	count	props	psets	dead	live
	states	16	12	?	0	16
	transitions	22	16	?	0	16


```

Tina version 3.6.0 -- 07/07/20 -- LAAS/CNRS
mode -W
INPUT NET -----
parsed net protocole
12 places, 16 transitions, 40 arcs

net protocole
tr t0 p0 -> p1 p8
tr t1 [5,6] p1 -> p1 p8
tr t10 [0,1] p10 p7 -> p6
tr t11 [0,2] p6 -> p11 p7
tr t12 [0,1] p8 ->
tr t13 [0,1] p9 ->
tr t14 [0,1] p10 ->
tr t15 [0,1] p11 ->
tr t2 [0,1] p1 p9 -> p2
tr t3 p2 -> p10 p3
tr t4 [5,6] p3 -> p10 p3
tr t5 [0,1] p11 p3 -> p0
tr t6 p7 p8 -> p4
tr t7 [0,1] p5 p8 -> p4
tr t8 [0,2] p4 -> p5 p9
tr t9 [0,1] p10 p5 -> p6
p1 p0 (1)
p1 p7 (1)

0.000s
REACHABILITY ANALYSIS -----
bounded
16 classe(s), 22 transition(s)

CLASSES:
class 0
  marking
    p0 p7
  domain
    0 <= t0

class 1
  marking
    p1 p7 p8
  domain
    5 <= t1 <= 6
    0 <= t12 <= 1
    0 <= t6

class 2
  marking
    p1 p7
  domain
    4 <= t1 <= 6

class 3
  marking
    p1 p4
  domain
    4 <= t1 <= 6
    0 <= t8 <= 2

class 4
  marking
    p1 p5 p9
  domain
    2 <= t1 <= 6

class 5
  marking
    p1 p5
  domain
    1 <= t1 <= 6

class 6
  marking
    p2 p5
  domain
    0 <= t3

class 7
  marking
    p1 p5 p8
  domain
    5 <= t1 <= 6
    0 <= t12 <= 1
    0 <= t7 <= 1

class 8
  marking
    p10 p3 p5
  domain
    0 <= t14 <= 1
    5 <= t4 <= 6
    0 <= t9 <= 1

class 9
  marking
    p1 p5
  domain
    4 <= t1 <= 6
  
```

Figure 42: Le SCG du protocole BA

```

class 10
  marking
    p3 p5
  domain
    4 <= t4 <= 6

class 11
  marking
    p3 p6
  domain
    0 <= t11 <= 2
    4 <= t4 <= 6

class 12
  marking
    p11 p3 p7
  domain
    0 <= t15 <= 1
    2 <= t4 <= 6
    0 <= t5 <= 1

class 13
  marking
    p3 p7
  domain
    1 <= t4 <= 6

class 14
  marking
    p10 p3 p7
  domain
    0 <= t10 <= 1
    0 <= t14 <= 1
    5 <= t4 <= 6

class 15
  marking
    p3 p7
  domain
    4 <= t4 <= 6

REACHABILITY GRAPH:
0 -> t0 in [0,w]/1
1 -> t12 in [0,1]/2, t6 in [0,1]/3
2 -> t1 in [4,6]/1
3 -> t8 in [0,2]/4
4 -> t13 in [0,1]/5, t2 in [0,1]/6
5 -> t1 in [1,6]/7
6 -> t3 in [0,w]/8
7 -> t12 in [0,1]/9, t7 in [0,1]/3
8 -> t14 in [0,1]/10, t9 in [0,1]/11
9 -> t1 in [4,6]/7
10 -> t4 in [4,6]/8
11 -> t11 in [0,2]/12
12 -> t15 in [0,1]/13, t5 in [0,1]/0
13 -> t4 in [1,6]/14
14 -> t10 in [0,1]/11, t14 in [0,1]/15
15 -> t4 in [4,6]/14

0.000s

LIVENESS ANALYSIS -----
possibly live
possibly reversible
0 dead classe(s), 16 live classe(s)
0 dead transition(s), 16 live transition(s)

STRONG CONNECTED COMPONENTS:

STRONG CONNECTED COMPONENTS:
0 : 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

SCC GRAPH:
0 -> t4/0, t10/0, t14/0, t15/0, t5/0, t11/0, t1/0, t9/0, t12/0, t7/0, t3/0,
0.000s

ANALYSIS COMPLETED -----

```

Figure43 : La suite de la figure 42

```

[C:\Users\User\Downloads\Compressed\tina-3.6.0-x64-mswin\tina-3.6.0\nets\tpn_examples\expl\protocole.aut]
File Edit View Tools Help
des (0,24,16)
(0,"*",0)
(0,"τ0",1)
(1,"τ12",2)
(1,"τ6",3)
(2,"τ1",1)
(3,"τ8",4)
(4,"τ13",5)
(4,"τ2",6)
(5,"τ1",7)
(6,"*",6)
(6,"τ3",8)
(7,"τ12",9)
(7,"τ7",3)
(8,"τ14",10)
(8,"τ9",11)
(9,"τ1",7)
(10,"τ4",8)
(11,"τ11",12)
(12,"τ15",13)
(12,"τ5",0)
(13,"τ4",14)
(14,"τ10",11)
(14,"τ14",15)
(15,"τ4",14)
    
```

Figure44 : La forme textuelle du SCG (.aut)

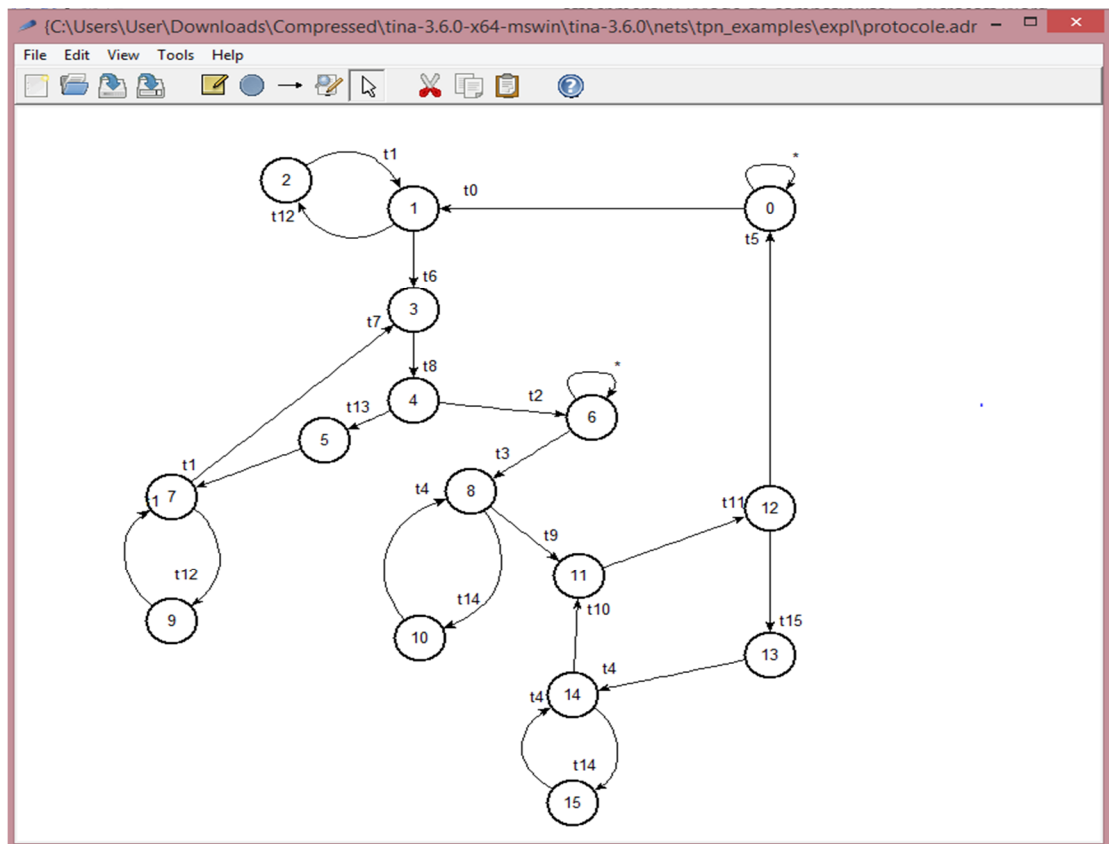


Figure 45 : Le Graphede classes SCG de BA (.adr)

Les deux figures suivantes représentent le SSCG du protocole BA :

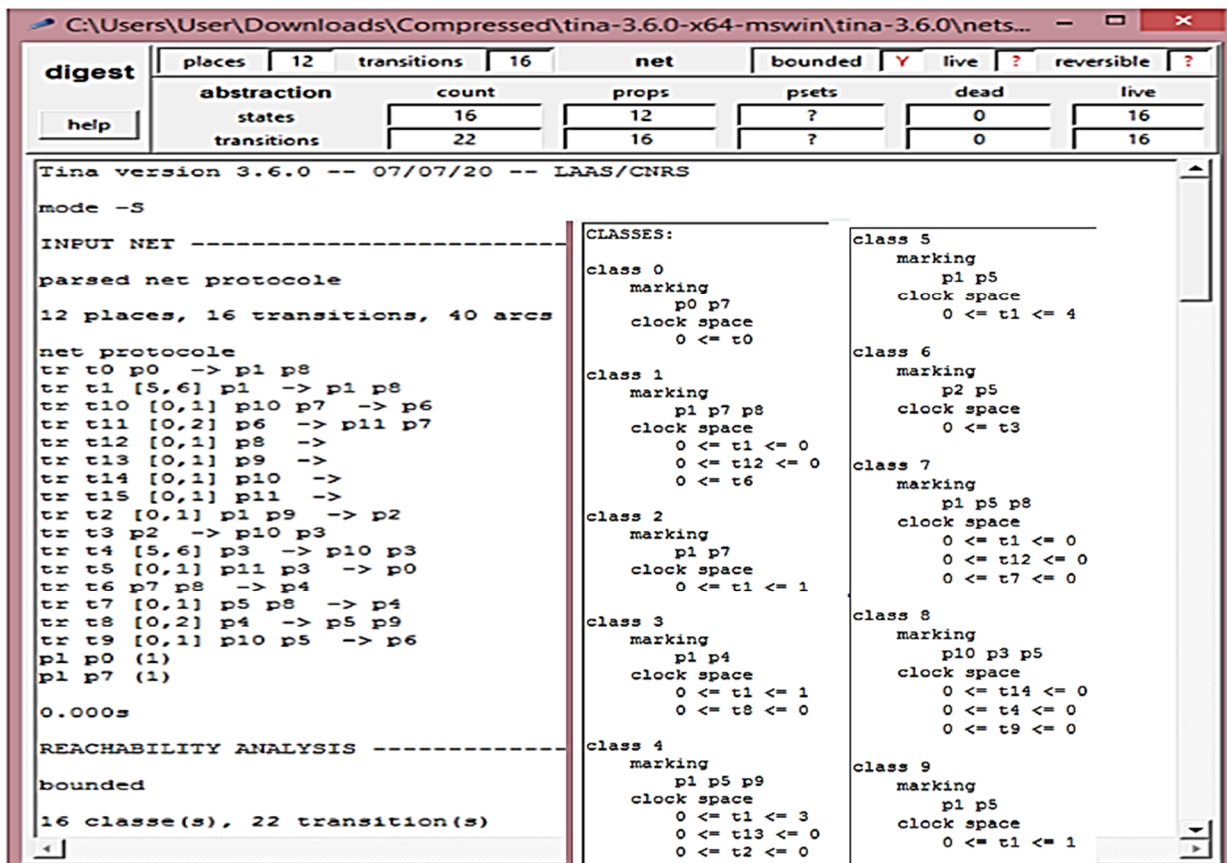


Figure 46:Le SSCG de protocole BA

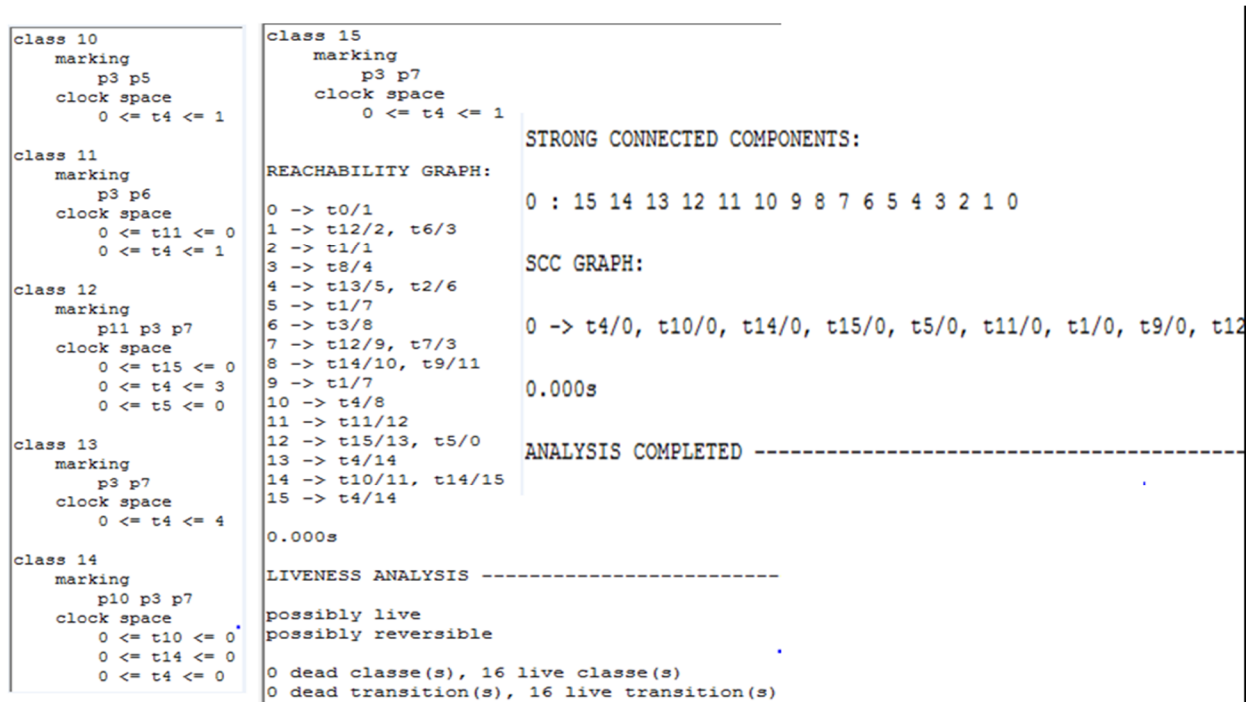


Figure 47:La suite de la figure 46

```

{C:\Users\User\Downloads\Compressed\tina-3.6.0-x64-mswin\tina-3.6.0\nets\tpn_examples\expl\protocole.aut}
File Edit View Tools Help
des(0,24,16)
(0,"*",0)
(0,"τ0",1)
(1,"τ12",2)
(1,"τ6",3)
(2,"τ1",1)
(3,"τ8",4)
(4,"τ13",5)
(4,"τ2",6)
(5,"τ1",7)
(6,"*",6)
(6,"τ3",8)
(7,"τ12",9)
(7,"τ7",3)
(8,"τ14",10)
(8,"τ9",11)
(9,"τ1",7)
(10,"τ4",8)
(11,"τ11",12)
(12,"τ15",13)
(12,"τ5",0)
(13,"τ4",14)
(14,"τ10",11)
(14,"τ14",15)
(15,"τ4",14)
    
```

Figure 48 : Le texte du SSCG (.aut)

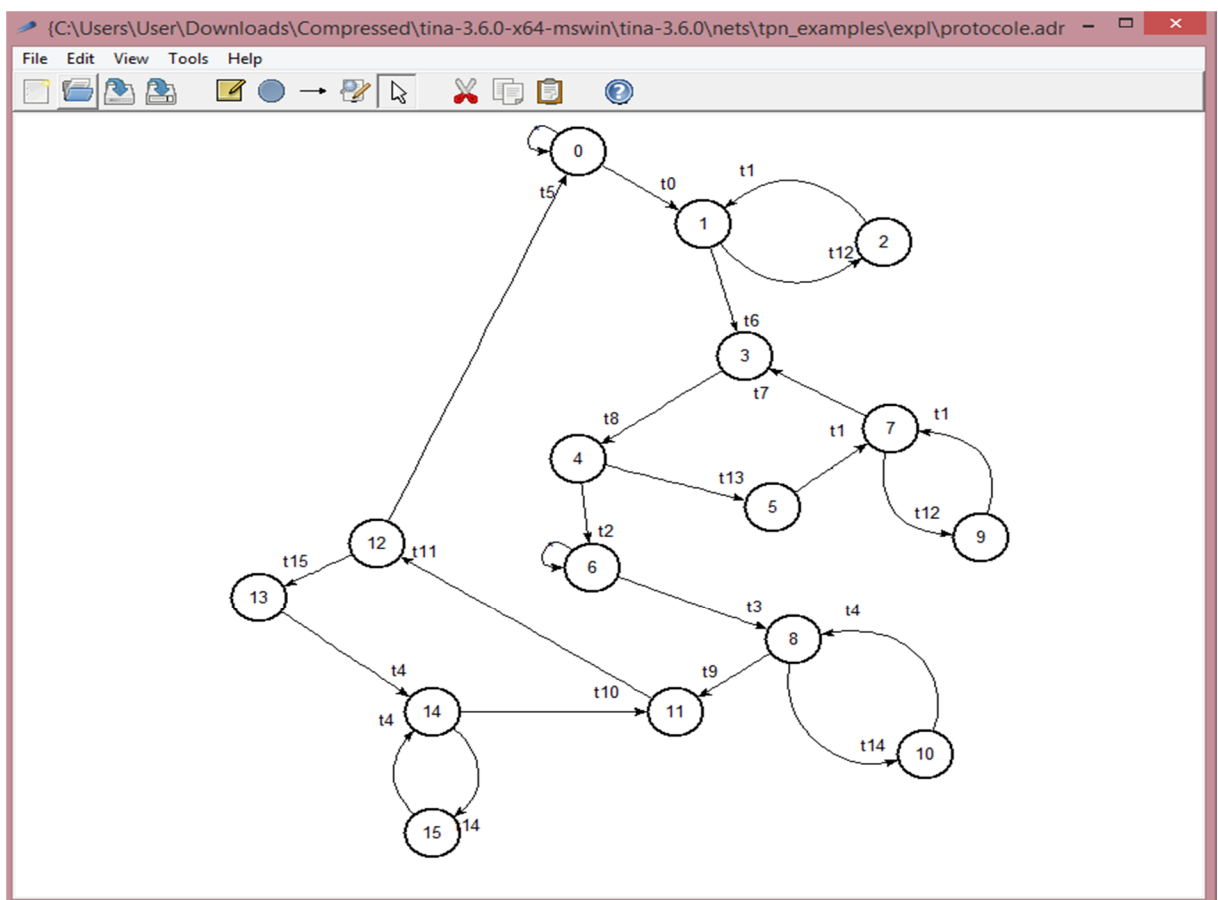


Figure 49 : Le graphe SSCG (.adr)

Les deux figures suivantes représentent ASCG du protocole BA :

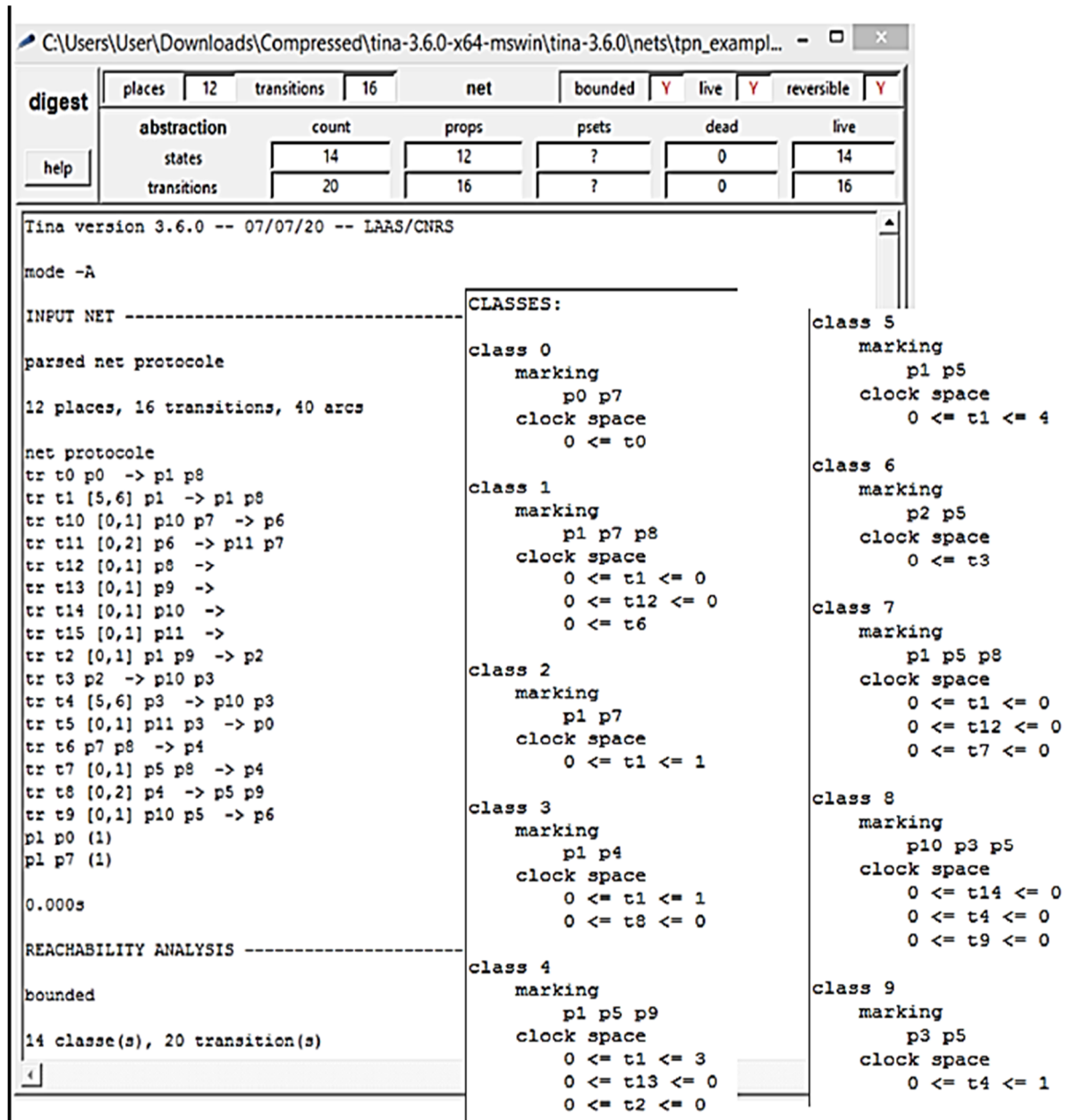


Figure 50: Le ASCG du protocole BA

```

class 10
  marking
    p3 p6
  clock space
    0 <= t11 <= 0
    0 <= t4 <= 1
class 11
  marking
    p11 p3 p7
  clock space
    0 <= t15 <= 0
    0 <= t4 <= 3
    0 <= t5 <= 0
class 12
  marking
    p3 p7
  clock space
    0 <= t4 <= 4
class 13
  marking
    p10 p3 p7
  clock space
    0 <= t10 <= 0
    0 <= t14 <= 0
    0 <= t4 <= 0

REACHABILITY GRAPH:
0 -> t0/1
1 -> t12/2, t6/3
2 -> t1/1
3 -> t8/4
4 -> t13/5, t2/6

1 -> t12/2, t6/3
2 -> t1/1
3 -> t8/4
4 -> t13/5, t2/6
5 -> t1/7
6 -> t3/8
7 -> t12/5, t7/3
8 -> t14/9, t9/10
9 -> t4/8
10 -> t11/11
11 -> t15/12, t5/0
12 -> t4/13
13 -> t10/10, t14/12

0.000s

LIVENESS ANALYSIS -----
live
reversible

0 dead classe(s), 14 live classe(s)
0 dead transition(s), 16 live transition(s)

STRONG CONNECTED COMPONENTS:
0 : 13 12 11 10 9 8 7 6 5 4 3 2 1 0

SCC GRAPH:
0 -> t10/0, t14/0, t4/0, t15/0, t5/0, t11/0, t9/0, t12/0, t7/0, t3/0, t1/0, t13/0, t2.

0.000s

ANALYSIS COMPLETED -----

```

Figure 51: La suite de la figure 50

```

[C:\Users\User\Downloads\Compressed\tina-3.6.0-x64-mswin\tina-3.6.0\nets\tpn_examples\expl\protocole.aut]
File Edit View Tools Help
des(0,22,14)
(0,"*",0)
(0,"t0",1)
(1,"t12",2)
(1,"t6",3)
(2,"t1",1)
(3,"t8",4)
(4,"t13",5)
(4,"t2",6)
(5,"t1",7)
(6,"*",6)
(6,"t3",8)
(7,"t12",5)
(7,"t7",3)
(8,"t14",9)
(8,"t9",10)
(9,"t4",8)
(10,"t11",11)
(11,"t15",12)
(11,"t5",0)
(12,"t4",13)
(13,"t10",10)
(13,"t14",12)
    
```

Figure52 : Le texte d'algorithme ASCG (.aut)

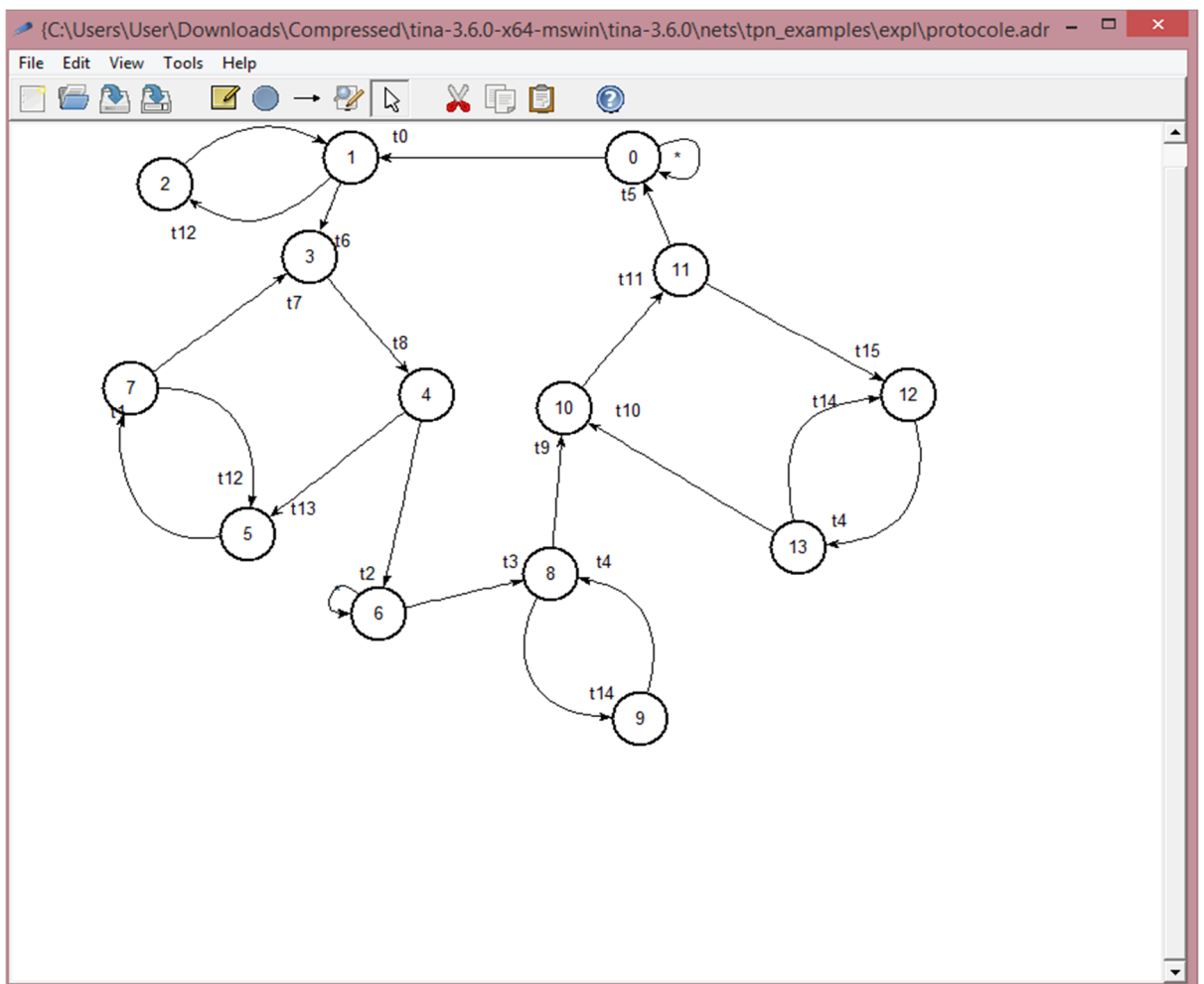


Figure53 : Le graphe d'algorithme ASCG


```

digest
  places 12 transitions 16 net bounded Y live Y reversible Y
  help
  abstraction count props psets dead live
  states      16     12   ?     0    16
  transitions 22     16   ?     0    16

Tina version 3.6.0 -- 07/07/20 -- LAAS/CNRS

mode -U

INPUT NET -----
parsed net protocole

12 places, 16 transitions, 40

net protocole
tr t0 p0 -> p1 p8
tr t1 [5,6] p1 -> p1 p8
tr t10 [0,1] p10 p7 -> p6
tr t11 [0,2] p6 -> p11 p7
tr t12 [0,1] p8 ->
tr t13 [0,1] p9 ->
tr t14 [0,1] p10 ->
tr t15 [0,1] p11 ->
tr t2 [0,1] p1 p9 -> p2
tr t3 p2 -> p10 p3
tr t4 [5,6] p3 -> p10 p3
tr t5 [0,1] p11 p3 -> p0
tr t6 p7 p8 -> p4
tr t7 [0,1] p5 p8 -> p4
tr t8 [0,2] p4 -> p5 p9
tr t9 [0,1] p10 p5 -> p6
p1 p0 (1)
p1 p7 (1)

0.016s

REACHABILITY ANALYSIS -----
bounded

16 classe(s), 22 transition(s)

CLASSES:

class 0
  marking
  p0 p7
  clock space
  0 <= t0

class 1
  marking
  p1 p7 p8
  clock space
  0 <= t1 <= 0
  0 <= t12 <= 0
  0 <= t6

class 2
  marking
  p1 p7
  clock space
  0 <= t1 <= 1

class 3
  marking
  p1 p4
  clock space
  0 <= t1 <= 1
  0 <= t8 <= 0

class 4
  marking
  p1 p5 p9
  clock space
  0 <= t1 <= 3
  0 <= t13 <= 0
  0 <= t2 <= 0

class 5
  marking
  p1 p5
  clock space
  0 <= t1 <= 4

class 6
  marking
  p2 p5
  clock space
  0 <= t3

class 7
  marking
  p1 p5 p8
  clock space
  0 <= t1 <= 0
  0 <= t12 <= 0
  0 <= t7 <= 0

class 8
  marking
  p10 p3 p5
  clock space
  0 <= t14 <= 0
  0 <= t4 <= 0
  0 <= t9 <= 0

class 9
  marking
  p1 p5
  clock space
  0 <= t1 <= 1

class 10
  marking
  p3 p5
  clock space
  0 <= t4 <= 1

class 11
  marking
  p3 p6
  clock space
  0 <= t11 <= 0
  0 <= t4 <= 1

class 12
  marking
  p11 p3 p7
  clock space
  0 <= t15 <= 0
  0 <= t4 <= 3
  0 <= t5 <= 0

class 13
  marking
  p3 p7
  clock space
  0 <= t4 <= 4

class 14
  marking
  p10 p3 p7
  clock space
  0 <= t10 <= 0
  0 <= t14 <= 0
  0 <= t4 <= 0

class 15
  marking
  p3 p7
  clock space
  0 <= t4 <= 1

REACHABILITY GRAPH:

0 -> t0/1
1 -> t12/2, t6/3
2 -> t1/1
3 -> t8/4
4 -> t13/5, t2/6
5 -> t1/7
6 -> t3/8
7 -> t12/9, t7/3
8 -> t14/10, t9/11
9 -> t1/7
10 -> t4/8
11 -> t11/12
12 -> t15/13, t5/0
13 -> t4/14
14 -> t10/11, t14/15
15 -> t4/14

0.000s
  
```

Figure 54: Les classes de graphe atomisé mode (-U).

```

des(0,24,16)
(0,"*",0)
(0,"t0",1)
(1,"t12",2)
(1,"t6",3)
(2,"t1",1)
(3,"t8",4)
(4,"t13",5)
(4,"t2",6)
(5,"t1",7)
(6,"*",6)
(6,"t3",8)
(7,"t12",9)
(7,"t7",3)
(8,"t14",10)
(8,"t9",11)
(9,"t1",7)
(10,"t4",8)
(11,"t11",12)
(12,"t15",13)
(12,"t5",0)
(13,"t4",14)
(14,"t10",11)
(14,"t14",15)
(15,"t4",14)
  
```

Figure 55: Le texte de classes atomisé mode (-U).

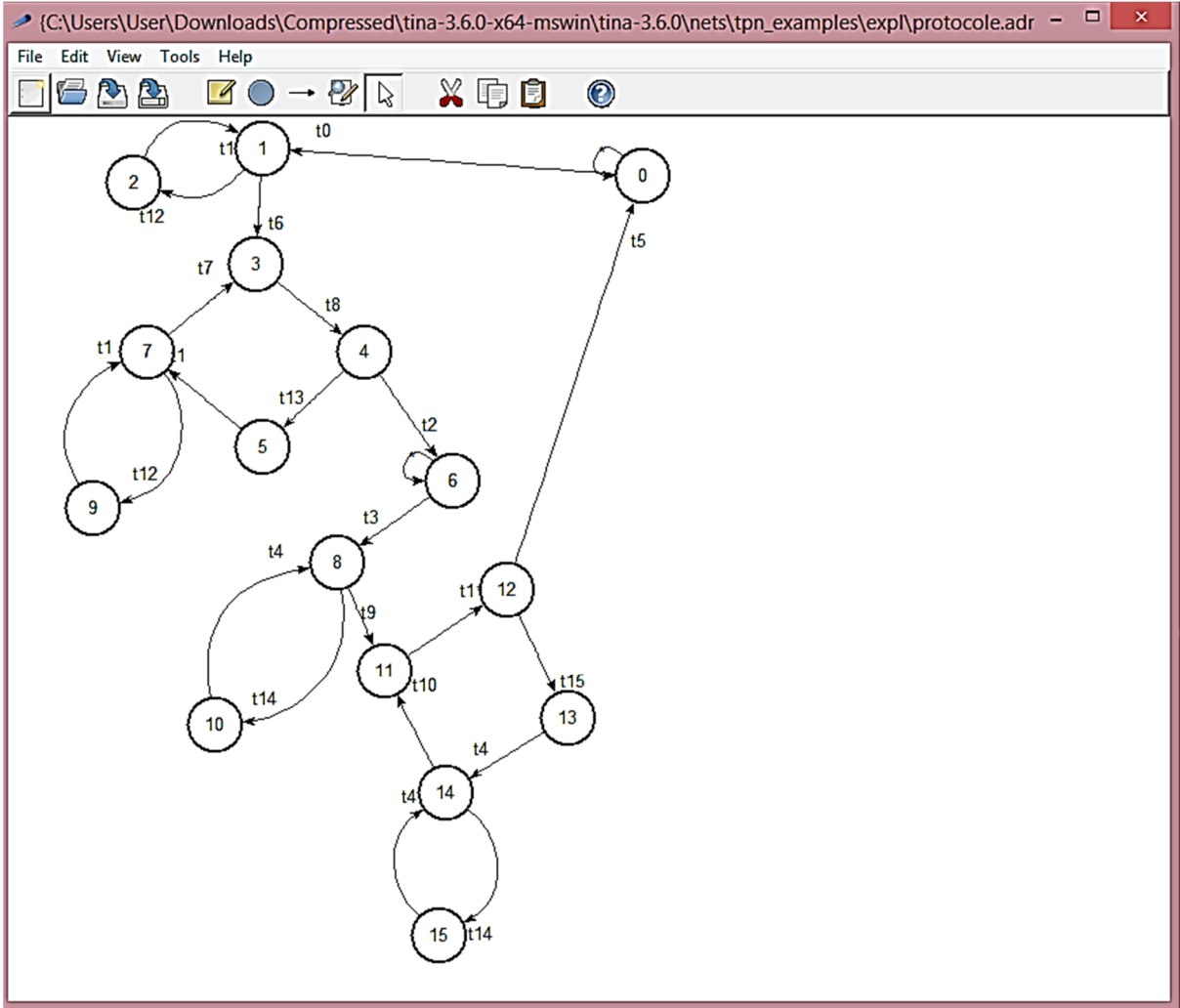


Figure 56: Le graphe de classes atomisé mode (-U).

abstraction	count	props	psets	dead	live
states	14	12	?	0	14
transitions	20	16	?	0	16

```

Tina version 3.6.0 -- 07/07/20 -- LAAS/CNRS
mode -E
INPUT NET -----
parsed net protocole
12 places, 16 transitions, 40 arcs

net protocole
tr t0 p0 -> p1 p8
tr t1 [5,6] p1 -> p1 p8
tr t10 [0,1] p10 p7 -> p6
tr t11 [0,2] p6 -> p11 p7
tr t12 [0,1] p8 ->
tr t13 [0,1] p9 ->
tr t14 [0,1] p10 ->
tr t15 [0,1] p11 ->
tr t2 [0,1] p1 p9 -> p2
tr t3 p2 -> p10 p3
tr t4 [5,6] p3 -> p10 p3
tr t5 [0,1] p11 p3 -> p0
tr t6 p7 p8 -> p4
tr t7 [0,1] p5 p8 -> p4
tr t8 [0,2] p4 -> p5 p9
tr t9 [0,1] p10 p5 -> p6
p1 p0 (1)
p1 p7 (1)
0.000s
REACHABILITY ANALYSIS -----
bounded
14 classe(s), 20 transition(s)

CLASSES:
class 0
  marking
  p0 p7
  clock space
  0 <= t0
class 1
  marking
  p1 p7 p8
  clock space
  0 <= t1 <= 0
  0 <= t12 <= 0
  0 <= t6
class 2
  marking
  p1 p7
  clock space
  0 <= t1 <= 1
class 3
  marking
  p1 p4
  clock space
  0 <= t1 <= 1
  0 <= t8 <= 0
class 4
  marking
  p1 p5 p9
  clock space
  0 <= t1 <= 3
  0 <= t13 <= 0
  0 <= t2 <= 0
class 5
  marking
  p1 p5
  clock space
  0 <= t1 <= 4
class 6
  marking
  p2 p5
  clock space
  0 <= t3
class 7
  marking
  p1 p5 p8
  clock space
  0 <= t1 <= 0
  0 <= t12 <= 0
  0 <= t7 <= 0
class 8
  marking
  p10 p3 p5
  clock space
  0 <= t14 <= 0
  0 <= t4 <= 0
  0 <= t9 <= 0
class 9
  marking
  p3 p5
  clock space
  0 <= t4 <= 1
class 10
  marking
  p3 p6
  clock space
  0 <= t11 <= 0
  0 <= t4 <= 1
class 11
  marking
  p11 p3 p7
  clock space
  0 <= t15 <= 0
  0 <= t4 <= 3
  0 <= t5 <= 0
class 12
  marking
  p3 p7
  clock space
  0 <= t4 <= 4
class 13
  marking
  p10 p3 p7
  clock space
  0 <= t10 <= 0
  0 <= t14 <= 0
  0 <= t4 <= 0
REACHABILITY GRAPH:
0 -> t0/1
1 -> t12/2, t6/3
2 -> t1/1
3 -> t8/4
4 -> t13/5, t2/6
  
```

Figure 57 : Les classes d'états mode (E)

```

des (0,22,14)
(0,"*",0)
(0,"t0",1)
(1,"t12",2)
(1,"t6",3)
(2,"t1",1)
(3,"t8",4)
(4,"t13",5)
(4,"t2",6)
(5,"t1",7)
(6,"*",6)
(6,"t3",8)
(7,"t12",5)
(7,"t7",3)
(8,"t14",9)
(8,"t9",10)
(9,"t4",8)
(10,"t11",11)
(11,"t15",12)
(11,"t5",0)
(12,"t4",13)
(13,"t10",10)
(13,"t14",12)
  
```

Figure 58: Le texte de classes d'états mode (E)

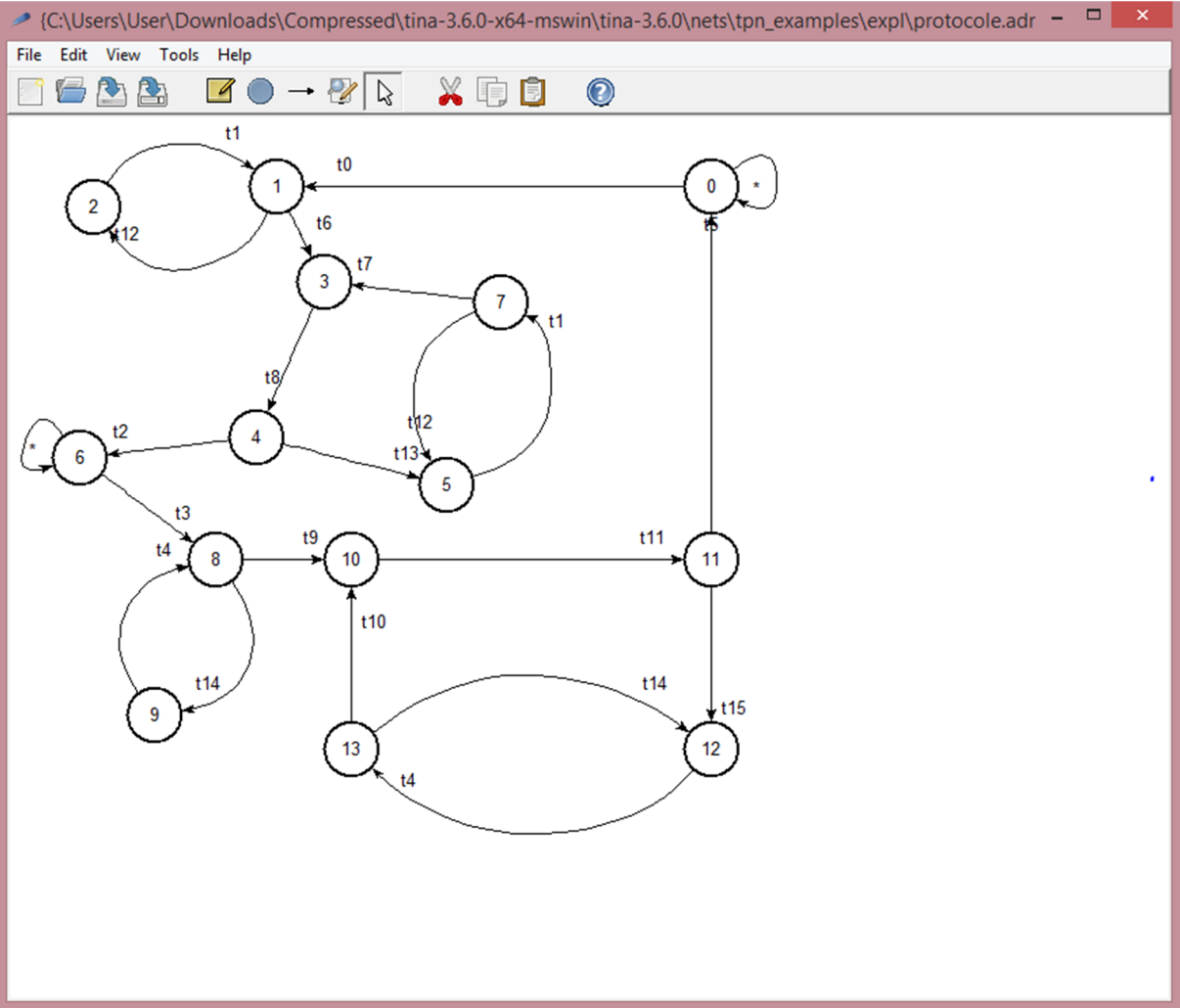


Figure 59: Le graphe de classes d'états mode (E)

```

digest | places | 12 transitions | 16 net | bounded |  live |  reversible |
help |
abstraction | count | props | psets | dead | live
states | 14 | 12 | ? | 0 | 14
transitions | 20 | 16 | ? | 0 | 16

Tina version 3.6.0 -- 07/07/20 -- LAAS/CNRS
mode -M
INPUT NET -----
parsed net protocole
12 places, 16 transitions, 40 arc

net protocole
tr t0 p0 -> p1 p8
tr t1 [5,6] p1 -> p1 p8
tr t10 [0,1] p10 p7 -> p6
tr t11 [0,2] p6 -> p11 p7
tr t12 [0,1] p8 ->
tr t13 [0,1] p9 ->
tr t14 [0,1] p10 ->
tr t15 [0,1] p11 ->
tr t2 [0,1] p1 p9 -> p2
tr t3 p2 -> p10 p3
tr t4 [5,6] p3 -> p10 p3
tr t5 [0,1] p11 p3 -> p0
tr t6 p7 p8 -> p4
tr t7 [0,1] p5 p8 -> p4
tr t8 [0,2] p4 -> p5 p9
tr t9 [0,1] p10 p5 -> p6
p1 p0 (1)
p1 p7 (1)
0.000s

REACHABILITY ANALYSIS -----
bounded
14 classe(s), 20 transition(s)

CLASSES:
class 0
  marking
  p0 p7
  domain
  0 <= t0

class 1
  marking
  p1 p7 p8
  domain
  5 <= t1 <= 6
  0 <= t12 <= 1
  0 <= t6

class 2
  marking
  p1 p7
  domain
  4 <= t1 <= 6

class 3
  marking
  p1 p4
  domain
  4 <= t1 <= 6
  0 <= t8 <= 2

class 4
  marking
  p1 p5 p9
  domain
  2 <= t1 <= 6
  0 <= t13 <= 1
  0 <= t2 <= 1

class 5
  marking
  p1 p5
  domain
  1 <= t1 <= 6

class 6
  marking
  p2 p5
  domain
  0 <= t3

class 7
  marking
  p1 p5 p8
  domain
  5 <= t1 <= 6
  0 <= t12 <= 1
  0 <= t7 <= 1

class 8
  marking
  p10 p3 p5
  domain
  0 <= t14 <= 1
  5 <= t4 <= 6
  0 <= t9 <= 1

class 9
  marking
  p3 p5
  domain
  4 <= t4 <= 6

class 10
  marking
  p3 p6
  domain
  0 <= t11 <= 2
  4 <= t4 <= 6

class 11
  marking
  p11 p3 p7
  domain
  0 <= t15 <= 1
  2 <= t4 <= 6
  0 <= t5 <= 1

class 12
  marking
  p3 p7
  domain
  1 <= t4 <= 6

class 13
  marking
  p10 p3 p7
  domain
  0 <= t10 <= 1
  0 <= t14 <= 1
  5 <= t4 <= 6

REACHABILITY GRAPH:
0 -> t0 in [0,w[/1
1 -> t12 in [0,1]/2, t6 in [0,1]/3
2 -> t1 in [4,6]/1
3 -> t8 in [0,2]/4
4 -> t13 in [0,1]/5, t2 in [0,1]/4
  
```

Figure 60: Les classes d'algorithme de marquage

```

(C:\Users\User\Downloads\Compressed\tina-3.6.0-x64-mswin\tina-3.6.0\nets\tpn_examples\expl\protocole.aut)
File Edit View Tools Help
des (0,22,14)
(0,"*",0)
(0,"t0",1)
(1,"t12",2)
(1,"t6",3)
(2,"t1",1)
(3,"t8",4)
(4,"t13",5)
(4,"t2",6)
(5,"t1",7)
(6,"*",6)
(6,"t3",8)
(7,"t12",5)
(7,"t7",3)
(8,"t14",9)
(8,"t9",10)
(9,"t4",8)
(10,"t11",11)
(11,"t15",12)
(11,"t5",0)
(12,"t4",13)
(13,"t10",10)
(13,"t14",12)
  
```

Figure 61: Le texte du graphe de marquage

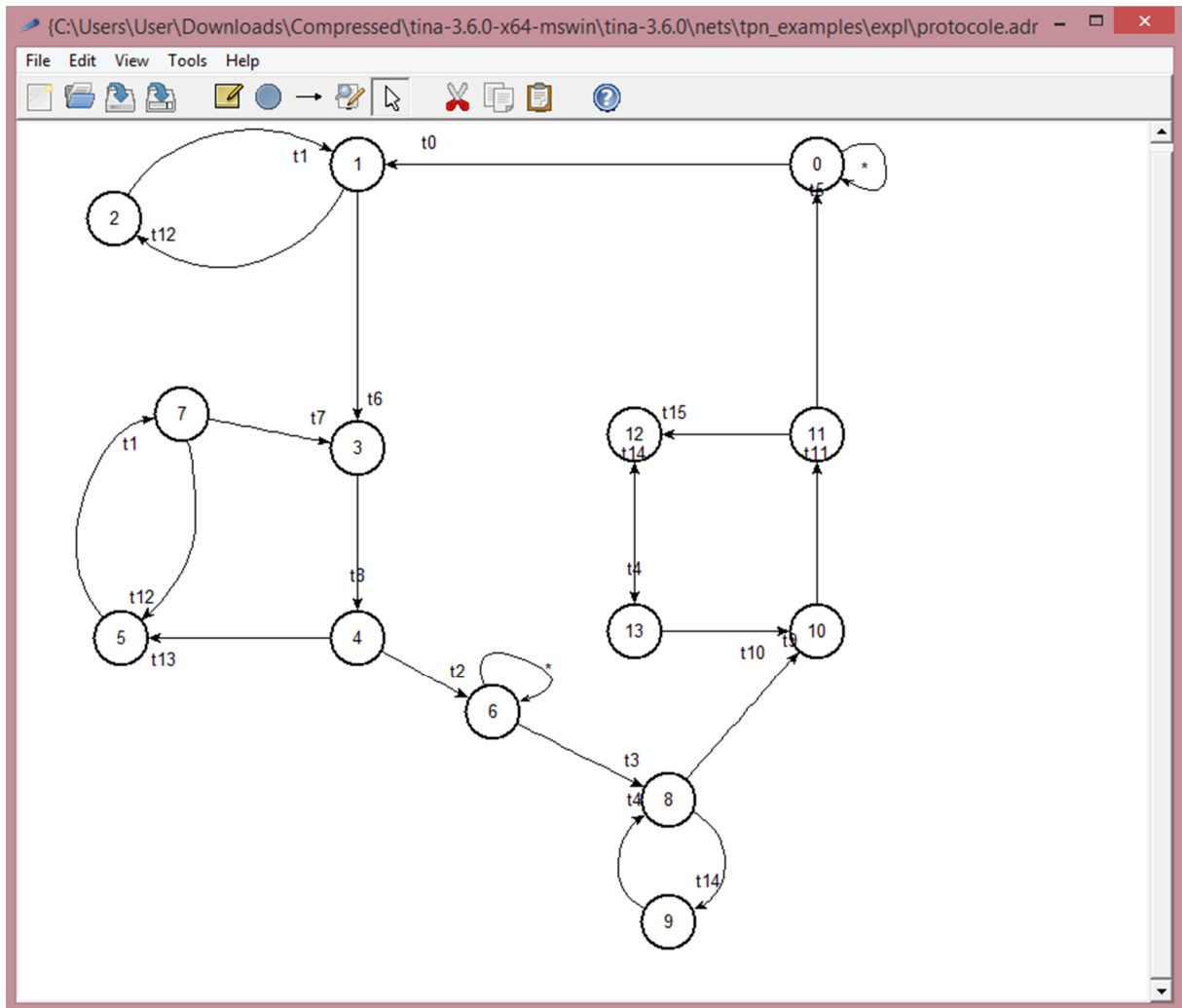


Figure62 : Le graphe de marquage mode (M)

3.1. Discussion

D'après la construction des différents graphes d'analyse appliquée sur ABP on avoir des résultats représentent ses propriétés tandis que tous les graphes prennent un temps nul (0.000s) sauf le graphe des classes atomisé qui prend (0.016s), notons que pour chacun les nombres de classes et de transitions sont respectivement soit 16 et 22 ou 14 et 20, et tous les états sont vivantes et seulement 16 transitions dans chaque graphe sont vivantes et aucune transition est morte, concernant les propriétés : les mode atomiques (mode U et le ASCG) dont des graphes bornés, vivants et réinitialisable ; et autres modes dont des graphes bornés et possible vivants et possible réinitialisable.

Tous les composants sont fortement connectés.

3.2. Comparaison

Tableau résume la différence entre les graphes :

Graphe		SCG	SSCG	ASCG	Marquage (M)	Classes d'états (E)	Classes Atomisé (U)
Nombre de classes		16	16	14	14	14	16
Nombre de transition		22	22	20	20	20	22
Propriétés	Borné	Oui	Oui	Oui	Oui	Oui	Oui
	Vivant	?	?	Oui	?	?	Oui
	Réinitialisable	?	?	Oui	?	?	Oui
Nombre d'états vivants		16	16	14	14	14	16
Nombre de transitions vivantes		16	16	16	16	16	16

Tableau3 : Comparaison entre les différents types des graphes

3.3. Discussion :

Le tableau plus-haut donne présente le nombre de classes et de transition de chaque graphe de classe calculé par l'outil TINA en utilisant les algorithmes détaillés dans les chapitres précédents. Il est à préciser que la méthode de *Berthomieu* et de *Boucheneb* ne diffère que dans la complexité de l'algorithme de calcule les résultats sont les mémés (on a le même graphe de sortie en nombre de classes et de transitions pour les deux).

Enfin, l'analyse et la vérification du protocole du bit alternée se fait naturellement à partir des graphes calculés tel que la bornitude, la vivacité la réinitialisationect.

4. Conclusion

Dans ce chapitre, nous avons discuté protocole Bit Alterné avec les RdpT dans un environnement logiciel Tina permet d'obtenue des abstractions préservent les propriétés linéaires (LTL) et les propriétés arborescentes (CTL), les marquages et les états.

•
 *Conclusion générale*

Conclusion générale


Le travail effectué dans le cadre de notre PFE est le protocole de communication « Le Bit Alterné » suivi d'une étude détaillée de ses propriétés qualitatives. L'outil principal de travail est les réseaux de Petri, et plus particulièrement on parle des réseaux de Petri temporels où le temps étant un facteur non négligeable. Ce formalisme est bien connu pour la modélisation, l'analyser et la vérification des systèmes complexes.

Notre étude a bien cernée un bon nombre de propriété (exp. vivacité, bornitude et réinitialisation), du protocole ABP et cela permet aux futur utilisateurs de cerner l'aspect *fiabilité et sureté* de ce dernier ?

Notre perspective pourrait être résumée comme suit :

- ▀ L'étude de propriétés quantitative du protocole
- ▀ L'étude de propriétés qualitative d'un autre protocole
- ▀ L'étude de propriétés quantitative de ce dernier.

.

 *Références bibliographiques*

Références bibliographiques

- [1] Carl Adam PETRI, Kommunikation mit Automaten, Thèse de doctorat, faculté de Darmstadt, Bonn, 1962.
- [2] VADE-MECUM DE L'ÉTUDIANT EN RÉSEAUX DE PETRI par Stéphane MARIEL
- [3] Réseaux de Petri par Vincent Augusto (Ecole Nationale Supérieure des Mines de Saint-Etienne '2012-2013)
- [4] Fouzia OUNNAR, Prise en compte des aspects décision dans la modélisation par réseaux de Petri des systèmes flexibles de production, Thèse de doctorat en automatique-productique présentée à l'INPG, 1999.
- [5] Carl Adam Petri et Wolfgang Reisig, « Petri net » [*Scholarpedia* 3(4):6477]
- [6] Introduction à la vérification structurelle des réseaux de Petri et des réseaux de haut-niveau Jean-François Pradat-Peyre Année 2010-2011
- [7] Petri Nets : Propriétés, analyse et application TADAŌ Murata IEEE
- [8] G. VIDAL-NAQUET et A. CHOQUET-GENIET : *Réseaux de Petri et systèmes parallèles*. 1992 Armand COLIN
- [9] G. Vidal-Naquet and A. Choquet-Geniet. Réseaux de Petri et systèmes parallèles Armand Colin
- [10] H. P. Hillion et J. M. Proth. « Analyse de Fabrication non linéaire et répétitive à l'aide des Graphes d'Événements Temporisés ». R.A.I.R.O., vol. 22, n°2, septembre 1988.
- [11] M. Hervé Hillion « Modélisation et analyse des systèmes de production discrets par les réseaux de Petri. ». Thèse de Doctorat, L'université Pierre et Marie Curie Paris VI, 23 Janvier 1989
- [12] W. Khansa, J.P. Denat, S. Collart-Dutilleul, « P-Time Petri Nets for Manufacturing Systems ». Wodes'96, Edinburgh UK, August 19-21, pp. 94-102, 1996, International Workshop on Discret Event Systems.
- [13] W. Khansa, « Réseaux de Petri P-temporels: contribution à l'étude des Systèmes à Événements Discrets ». Thèse de Doctorat de l'Université de Savoie, France, mars 1997.
- [14] P. Merlin, « A Study of the Recoverability of Communication Protocols ». Ph.D. Thesis, Computer Science Dep. University of California, Irvine, 1974.
- [15] B. Berthomieu & M. Diaz. Modeling and verification of time dependent systems using time Petri nets. IEEE Transactions on Software Engineering, vol. 17, n° 3, 1991.
- [16] David de Frutos Escrig, V. Valero Ruiz & O. Marroquin Alonso. Decidability of Properties of Timed-Arc Petri Nets. In Proceedings of ICATPN. Springer Verlag, 2000.
- [17] Parosh Aziz Abdulla & Aletta Nylen. Timed Petri Nets and BQOs. In Proceedings of ICATPN, pages 53–70. Springer-Verlag, 2001.

Références bibliographiques

- [18] Hans-Michael Hanisch. Analysis of Place/Transition Nets with Timed Arcs and its Application to Batch Process Control. In Proceedings of ICATPN, pages 282–299. Springer-Verlag, 1993.
- [19] M. Pezze & M. Toung. Time Petri nets : A primer introduction. In Tutorial presented at the Multi-Workshop on Formal Methods in Performance Evaluation and Applications, Zaragoza, Spain, September 1999.
- [20] B. Walter, "Timed Petri nets for modeling and analyzing protocol switch time", in IEEE Int. conf on protocol Specification Testing and verification.
- [21] B. Berthomieu, P.-O. Ribet & F. Vernadat. The tool TINA – Construction of Abstract State Spaces for Petri Nets and Time Petri Nets. International Journal of Production Research, vol. 42, no. 4, July 2004.
- [22] Bernard Berthomieu and François Vernadat. State class constructions for branching analysis of time petri nets. In International Conference on Tools and Algorithms for the Construction and Analysis of Systems, pages 442–457. Springer, 2003
- [23] Tomohiro Yoneda and Hikaru Ryuba. Ctl model checking of time petri Nets using geometric regions. IEICE Transactions on Information and Systems,
- [24] Hanifa Boucheneb and John Mullins. Analyse des réseaux temporels : Calcul des classes en $O(n^2)$ et des temps de chemin en $O(m \times n)$. 2003.
- [25] Bernard Berthomieu and Michel Diaz. Modeling and verification of time dependent systems using time petri nets. IEEE transactions on software engineering, 17(3):259, 1991.
- [26] Robert W Floyd. Algorithm 97: shortest path. Communications of the ACM, 5(6):345, 1962
- [27] Bartlett et Scantlebury dans Comm. ACM May 1969, disponible à partir de l'UQO .
- [28] D. Kofman et M. Gagnaire «Réseaux haut débits », Tome 1. 2ème édition Dunod 1999
- [29] Alexis Ferréro «Les réseaux locaux commutés et ATM », Inter Editions 1998.
- [30] Craig Hunt «TCP/IP : Administration de réseaux», O'Reilly, 2002.
- [31] Charles M. Kozirook «The TCP/IP guide», Aquarelle, 2003.
- [32] Bernard Berthomieu & Miguel Menasche. An Enumerative Approach.
- [33] AFEK Y., BROWN G. M., « Self-stabilization over unreliable communication media », *Distributed Computing*, vol. 7, no 1, 1993, p. 27-34.

Références bibliographiques

- [34] HOWELL R. R., NESTERENKO M., MIZUNO M., « Finite-state self-stabilizing protocols in message-passing systems », *Journal of Parallel and Distributed Computing*, vol. 62, no 5, 2002, p. 792-817.
- [35] DOLEV S., ISRAELI A., MORAN S., « Resource bounds for self-stabilizing message-driven protocols », *SIAM Journal on Computing*, vol. 26, no 1, 1997.
- [36] GOUDA M. G., MULTARI N., « Stabilizing communication protocols », *IEEE Transactions on Computers*, vol. 40, no 4, 1991.



Annexes

Graphe de classes d'états

Graphe		SCG
Nombre de classes		16
Nombre de transition		22
Propriétés	borné	Oui
	Vivant	?
	Réinitialisable	?
<i>Classe</i>	<i>Marquage</i>	<i>Domaine de tir</i>
0	P0, p7	$0 \leq t0$
1	P1, p7, p8	$5 \leq t1 \leq 6$ $0 \leq t12 \leq 1$ $0 \leq t6$
2	P1, p7	$4 \leq t1 \leq 6$
3	P1, p4	$4 \leq t1 \leq 6$ $0 \leq t8 \leq 2$
4	P1, p5, p9	$2 \leq t1 \leq 6$ $0 \leq t13 \leq 1$ $0 \leq t2 \leq 1$
5	P1, p5	$1 \leq t1 \leq 6$
6	P2, p5	$0 \leq t3$
7	P1, p5, p8	$5 \leq t1 \leq 6$ $0 \leq t12 \leq 1$ $0 \leq t7 \leq 1$
8	P10, p3, p5	$0 \leq t14 \leq 1$ $5 \leq t4 \leq 6$ $0 \leq t9 \leq 1$
9	P1, p5	$4 \leq t1 \leq 6$
10	P3, p5	$4 \leq t4 \leq 6$
11	P3, p6	$0 \leq t11 \leq 2$ $4 \leq t4 \leq 6$
12	P11, p3, p7	$0 \leq t15 \leq 1$ $2 \leq t4 \leq 6$ $0 \leq t5 \leq 1$
13	P3, p7	$1 \leq t4 \leq 6$
14	P10, p3, p7	$0 \leq t10 \leq 1$ $0 \leq t14 \leq 1$ $5 \leq t4 \leq 6$
15	P3, p7	$4 \leq t4 \leq 6$

Tableau 4 : Résultat détaillé de SCGduBA

Graphe de classes d'états fort

Graphe		SSCG
Nombre de classes		16
Nombre de transition		22
Propriétés	Borné	Oui
	Vivant	?
	Réinitialisable	?
<i>Classe</i>	<i>Marquage</i>	<i>Domaine de tir</i>
0	P0, p7	$0 \leq t_0$
1	P1, p7, p8	$0 \leq t_1 \leq 0$ $0 \leq t_{12} \leq 0$ $0 \leq t_6$
2	P1, p7	$0 \leq t_1 \leq 1$
3	P1, p4	$0 \leq t_1 \leq 1$ $0 \leq t_8 \leq 0$
4	P1, p5, p9	$0 \leq t_1 \leq 3$ $0 \leq t_{13} \leq 0$ $0 \leq t_2 \leq 0$
5	P1, p5	$0 \leq t_1 \leq 4$
6	P2, p5	$0 \leq t_3$
7	P1, p5, p8	$0 \leq t_1 \leq 0$ $0 \leq t_{12} \leq 0$ $0 \leq t_7 \leq 0$
8	P10, p3, p5	$0 \leq t_{14} \leq 0$ $0 \leq t_4 \leq 0$ $0 \leq t_9 \leq 0$
9	P1, p5	$0 \leq t_1 \leq 1$
10	P3, p5	$0 \leq t_4 \leq 1$
11	P3, p6	$0 \leq t_{11} \leq 0$ $0 \leq t_4 \leq 1$
12	P11, p3, p7	$0 \leq t_{15} \leq 0$ $0 \leq t_4 \leq 3$ $0 \leq t_5 \leq 0$
13	P3, p7	$0 \leq t_4 \leq 4$
14	P10, p3, p7	$0 \leq t_{10} \leq 0$ $0 \leq t_{14} \leq 0$ $0 \leq t_4 \leq 0$
15	P3, p7	$0 \leq t_4 \leq 1$

Tableau 5: Résultat détaillé de SSCGduBA

Graphe de classes d'états Atomique

Graphe		ASCG
Nombre de classes		14
Nombre de transition		20
Propriétés	Borné	Oui
	Vivant	Oui
	Réinitialisable	Oui
<hr/>		
<i>Classe</i>	<i>Marquage</i>	<i>Domaine de tir</i>
<i>0</i>	P0, p7	$0 \leq t_0$
<i>1</i>	P1, p7, p8	$0 \leq t_1 \leq 0$ $0 \leq t_{12} \leq 0$ $0 \leq t_6$
<i>2</i>	P1, p7	$0 \leq t_1 \leq 1$
<i>3</i>	P1, p4	$0 \leq t_1 \leq 1$ $0 \leq t_8 \leq 0$
<i>4</i>	P1, p5, p9	$0 \leq t_1 \leq 3$ $0 \leq t_{13} \leq 0$ $0 \leq t_2 \leq 0$
<i>5</i>	P1, p5	$0 \leq t_1 \leq 4$
<i>6</i>	P2, p5	$0 \leq t_3$
<i>7</i>	P1, p5, p8	$0 \leq t_1 \leq 0$ $0 \leq t_{12} \leq 0$ $0 \leq t_7 \leq 0$
<i>8</i>	P10, p3, p5	$0 \leq t_{14} \leq 0$ $0 \leq t_4 \leq 0$ $0 \leq t_9 \leq 0$
<i>9</i>	P3, p5	$0 \leq t_4 \leq 1$
<i>10</i>	P3, p6	$0 \leq t_{11} \leq 0$ $0 \leq t_4 \leq 1$
<i>11</i>	P11, p3, p7	$0 \leq t_{15} \leq 0$ $0 \leq t_4 \leq 3$ $0 \leq t_5 \leq 0$
<i>12</i>	P3, p7	$0 \leq t_4 \leq 4$
<i>13</i>	P10, p3, p7	$0 \leq t_{10} \leq 0$ $0 \leq t_{14} \leq 0$ $0 \leq t_4 \leq 0$

Tableau 6 : Résultat détaillé de ASCGduBA

Graphe d'états

Graphe		Graphe d'états
Nombre de classes		14
Nombre de transition		20
Propriétés	Borné	Oui
	Vivant	?
	Réinitialisable	?
<i>Classe</i>	<i>Marquage</i>	<i>Domaine de tir</i>
0	P0, p7	$0 \leq t0$
1	P1, p7, p8	$0 \leq t1 \leq 0$ $0 \leq t12 \leq 0$ $0 \leq t6$
2	P1, p7	$0 \leq t1 \leq 1$
3	P1, p4	$0 \leq t1 \leq 1$ $0 \leq t8 \leq 0$
4	P1, p5, p9	$0 \leq t1 \leq 3$ $0 \leq t13 \leq 0$ $0 \leq t2 \leq 0$
5	P1, p5	$0 \leq t1 \leq 4$
6	P2, p5	$0 \leq t3$
7	P1, p5, p8	$0 \leq t1 \leq 0$ $0 \leq t12 \leq 0$ $0 \leq t7 \leq 0$
8	P10, p3, p5	$0 \leq t14 \leq 0$ $0 \leq t4 \leq 0$ $0 \leq t9 \leq 0$
9	P3, p5	$0 \leq t4 \leq 1$
10	P3, p6	$0 \leq t11 \leq 0$ $0 \leq t4 \leq 1$
11	P11, P3, p7	$0 \leq t15 \leq 0$ $0 \leq t4 \leq 3$ $0 \leq t5 \leq 0$
12	p3, p7	$0 \leq t4 \leq 4$
13	P10, P3, p7	$0 \leq t10 \leq 0$ $0 \leq t14 \leq 0$ $0 \leq t4 \leq 0$

Tableau 7 : Résultat détaillé de graphe d'états

Graphe de marquage

Graphe		Marquage
Nombre de classes		14
Nombre de transition		22
Propriétés	Borné	Oui
	Vivant	?
	Réinitialisable	?
<i>Classe</i>	<i>Marquage</i>	<i>Domaine de tir</i>
0	P0, p7	$0 \leq t0$
1	P1, p7, p8	$5 \leq t1 \leq 6$ $0 \leq t12 \leq 1$ $0 \leq t6$
2	P1, p7	$4 \leq t1 \leq 6$
3	P1, p4	$4 \leq t1 \leq 6$ $0 \leq t8 \leq 2$
4	P1, p5, p9	$2 \leq t1 \leq 6$ $0 \leq t13 \leq 1$ $0 \leq t2 \leq 1$
5	P1, p5	$1 \leq t1 \leq 6$
6	P2, p5	$0 \leq t3$
7	P1, p5, p8	$5 \leq t1 \leq 6$ $0 \leq t12 \leq 1$ $0 \leq t7 \leq 1$
8	P10, p3, p5	$0 \leq t14 \leq 1$ $5 \leq t4 \leq 6$ $0 \leq t9 \leq 1$
9	P3, p5	$4 \leq t4 \leq 6$
10	P3, p6	$0 \leq t11 \leq 2$ $4 \leq t4 \leq 6$
11	P11, P3, p7	$0 \leq t15 \leq 1$ $2 \leq t4 \leq 6$ $0 \leq t5 \leq 1$
12	p3, p7	$1 \leq t4 \leq 6$
13	P10, P3, p7	$0 \leq t4 \leq 1$ $0 \leq t4 \leq 1$ $5 \leq t4 \leq 6$

Tableau 8 : Résultat détaillé du graphe de Marquage

Graphe atomisé mode (U)

Graphe		Graphe atomisé de boucheneb
Nombre de classes		16
Nombre de transition		22
Propriétés	Borné	Oui
	Vivant	Oui
	Réinitialisable	Oui
<i>Classe</i>	<i>Marquage</i>	<i>Domaine de tir</i>
0	P0, p7	$0 \leq t0$
1	P1, p7, p8	$0 \leq t1 \leq 0$ $0 \leq t12 \leq 0$ $0 \leq t6$
2	P1, p7	$0 \leq t1 \leq 1$
3	P1, p4	$0 \leq t1 \leq 1$ $0 \leq t8 \leq 0$
4	P1, p5, p9	$0 \leq t1 \leq 3$ $0 \leq t13 \leq 0$ $0 \leq t2 \leq 0$
5	P1, p5	$0 \leq t1 \leq 4$
6	P2, p5	$0 \leq t3$
7	p1, p5, p8	$0 \leq t1 \leq 0$ $0 \leq t12 \leq 0$ $0 \leq t7 \leq 0$
8	P10, p3, p5	$0 \leq t14 \leq 0$ $0 \leq t4 \leq 0$ $0 \leq t9 \leq 0$
9	P1, p5	$0 \leq t1 \leq 1$
10	P3, p5	$0 \leq t4 \leq 1$
11	P3, p6	$0 \leq t11 \leq 0$ $0 \leq t4 \leq 1$
12	P11, p3, p7	$0 \leq t15 \leq 0$ $0 \leq t4 \leq 3$ $0 \leq t5 \leq 0$
13	P3, p7	$0 \leq t4 \leq 4$
14	P10, p3, p7	$0 \leq t10 \leq 0$ $0 \leq t14 \leq 0$ $0 \leq t4 \leq 0$
15	P3, p7	$0 \leq t4 \leq 1$

Tableau 9 : Résultat détaillé Graphe atomisé mode (U)