République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 08 Mai 1945 Guelma
Faculté des Mathématiques, de l'Informatique et des Sciences de la Matière
Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Science et Technologie de l'information et de la communication

Thème :

Implémentation d'un outil pour générerle graphe d'état d'un réseau de Petri réversible : Compilation

Le jury

Dr. Khebizi Ali	Président
Dr. Hannousse Abdelhakim	Examinateur
Dr. Benamira Adel	Encadreur

Réalisé Par :
Mlle. KHERAIFIA KARIMA

Juin 2023

REMERCîMENT

Cher comité d'évaluation,

Je tiens à exprimer ma sincère gratitude pour l'opportunité qui m'a été donnée de présenter ma mémoire devant vous. C'est avec une profonde reconnaissance que ge souhaite adresser mes remerciements à chacun d'entre vous pour votre temps, votre expertise et votre évaluation minutieuse.

Je suis consciente que l'évaluation d'une mémoire demande un investissement considérable en termes d'efforts et d'attention. Votre engagement envers la recherche et le développement académique est admirable, et ge suis honorée d'avoir pu bénéficier de votre expertise dans ce processus.

Je tiens également à remercier tout particulièrement Monsieur BENAMIRA ADEL *pour sa guidance, ses conseils précieux et son soutien tout au long de la réalisation de cette mémoire. Votre expertise et votre mentorat m'ont permis de repousser mes limites et de développer mes compétences de recherche de manière significative.*

Je suis également reconnaissante envers mes proches, ma famille et mes amis, qui m'ont soutenue tout au long de ce parcours académique. Votre amour, votre soutien inconditionnel et vos encouragements ont été les moteurs qui m'ont poussée à donner le meilleur de moi-même.

Encore une fois, ge tiens à vous remercier du fond du cœur pour votre évaluation attentive et constructive de ma mémoire. Vos commentaires et suggestions me seront d'une grande valeur pour ma croissance académique future. Je suis reconnaissante de l'occasion qui m'a été offerte de présenter mes recherches et de contribuer au domaine qui m'inspire.

Avec mes sentiments de gratitude les plus sincères.

KARIMA

DÉDICACE

Je dédie cette mémoire à ceux qui ont joué un rôle précieux dans ma vie et m'ont soutenu à chaque étape de mon parcours. Je suis profondément reconnaissant envers ALLAH*, qui a été ma source de force et de guidance.*

Je tiens à exprimer ma gratitude à ma mère, dont la bienveillance et l'amour ont été une source d'inspiration pour moi. Tu as été là pour moi, guidant mes pas avec sagesse et compassion.

À mon père, tu as été mon pilier tout au long de mon cheminement. Ta présence et ton soutien inconditionnel m'ont permis d'atteindre mes objectifs. Je suis honoré d'avoir eu un père aussi exceptionnel et engagé.

À mon fiancé, OUSSAMA, *ta main droite a été mon soutien constant. Je te remercie pour ton encouragement infini et pour les nuits passées à mes côtés lorsque les temps étaient difficiles. Ta présence a été une lumière dans les moments sombres.*

À ma sœur, KHAWLA, *je t'exprime ma profonde gratitude pour ton soutien indéfectible et ta foi en mes capacités. Ton soutien inébranlable m'a donné la confiance nécessaire pour surmonter les obstacles.*

Je souhaite également exprimer ma profonde gratitude envers mes amies proches, CHOUROUK *et* HIND. CHOUROUK, *tu es une amie précieuse qui m'a toujours soutenue avec ton soutien inconditionnel et tes encouragements. Ta présence dans ma vie a apporté de la joie et de la force, et je te suis reconnaissante pour ta fidélité et ton amitié sincère.* HIND, *tu es plus qu'une amie, tu es à la fois une collègue et une amie fidèle. Ton soutien et ta camaraderie m'ont accompagnée à chaque étape de ce parcours, et je suis reconnaissante d'avoir quelqu'un comme toi à mes côtés.*

Je tiens également à exprimer ma gratitude envers ma proche amie MALAK*, dont la présence et le soutien ont été inestimables. Ta gentillesse et ton amitié ont illuminé ma vie, et je te suis reconnaissante pour tous les moments partagés ensemble.*

Enfin, je souhaite adresser mes remerciements à ma sœur SAMIRA*, ainsi qu'à mes frères. Votre soutien constant et votre amour inconditionnel ont été une source d'inspiration et de réconfort tout au long de ce parcours. Votre présence dans ma vie a été une bénédiction, et je suis reconnaissante d'avoir une famille aussi aimante et solidaire.*

À vous tous, je suis profondément reconnaissant pour votre présence dans ma vie. Vos encouragements, votre soutien et votre amour m'ont permis de réaliser cette étape importante. Que notre lien d'amitié et de soutien perdure toujours.

Avec amour et gratitude

Résumé

Le calcul réversible est un paradigme de calcul où chaque opération est réversible, ce qui signifie qu'il est possible de retrouver l'état précédent à partir de l'état actuel. Contrairement au calcul classique, où des informations peuvent être perdues, le calcul réversible conserve toutes les informations nécessaires pour inverser les opérations et revenir à un état antérieur.

Les réseaux de Petri réversibles sont une extension des réseaux de Petri, qui sont des outils graphiques utilisés pour modéliser les systèmes concurrents et distribués. Dans les réseaux de Petri réversibles, les transitions sont réversibles, ce qui signifie qu'il est possible d'inverser le marquage du réseau pour revenir à un état précédent. Cela permet de représenter des systèmes où les actions peuvent être annulées, les ressources récupérées et les états précédents restaurés.

Ces concepts ont des applications dans des domaines tels que la conception de circuits logiques réversibles, la simulation de systèmes physiques réversibles et la conception de systèmes informatiques à faible consommation d'énergie. Ils sont particulièrement utiles dans les domaines de la recherche et de l'application où la réversibilité et la récupération d'information sont cruciales, comme dans les systèmes de calcul quantique.

Dans ce présent projet nous avons conçu un module (compilateur) qui prend en entrée une description textuelle d'un Réseau de Petri Réversible (RPR) et la transformes en structure de données spécifique pour être utilisée ensuite par d'autres modules. Précisément, nous avons (i) proposé une grammaire adéquate aux réseaux de Petri réversibles (ii) développé l'analyseur lexical.

Mots-clés :
Réseau de Petri réversible (RPR), Reversible computation (RC), LEX, YACC.

Abstract

Reversible computing is a computing paradigm where each operation is reversible, meaning it is possible to recover the previous state from the current state. Unlike classical computing, where information can be lost, reversible computing preserves all the necessary information to reverse operations and return to a previous state.

Reversible Petri nets are an extension of Petri nets, which are graphical tools used to model concurrent and distributed systems. In reversible Petri nets, transitions are reversible, meaning it is possible to reverse the marking of the net to return to a previous state. This allows representing systems where actions can be undone, resources recovered, and previous states restored.

These concepts have applications in areas such as reversible logic circuit design, simulation of reversible physical systems, and low-power computing system design. They are particularly useful in research and application domains where reversibility and information recovery are crucial, such as quantum computing systems.

In this project, we have designed a module (compiler) that takes a textual description of a Reversible Petri Net (RPN) as input and transforms it into a specific data structure to be used by other modules. Specifically, we have (i) proposed a suitable grammar for reversible Petri nets and (ii) developed the lexical analyzer.

1 Table des matières

Table des matières

INTRODUCTION GENERALE **1**

LE CALCUL REVERSIBLE **6**

I.1 LE CALCUL REVERSIBLE 7
I.1.1. Modèles Séquentielle *8*
I.1.1.1 Automates 8
I.1.1.2 Machines Turing 9
I.1.2. Modèles concurrentes *9*
I.1.2.1 Automate Cellulaire 9
I.1.2.2 Processus de Calcul 10
I.1.2.3 Calcul Membrane 11
I.1.2.4 Réseau de Petri 11
I.2 LE CALCUL REVERSIBLE ET RESEAU DE PETRI 11
I.3 CONCLUSION 14

RESEAUX DE PETRI REVERSIBLES **16**

I.1 LE RESEAU DE PETRI 17
II.1.2. Définitions fondamentales *17*
II.1.2.1. Définition informelle 17
I.1.2.2 Définition formelle 17
II.2 GRAPHE DE MARQUAGE 19
II.3 LA REVERSIBILITE DANS LE RESEAU DE PETRI 20
II.4 SEMANTIQUE DE CAUSALITE 23
II.4.1 Les interprétations individuelles et collectives des jetons *23*
II.4.2 Une règle de tir pour l'interprétation individuelle des jetons *25*
II.5 RESEAU DE PETRI REVERSIBLE 27
II.5.1. La réversibilité de réseau de Petri avec plusieurs jetons : *30*
II.6 CONCLUSION 34

CONCEPTION ET IMPLEMENTATION **35**

III.1 LA CONCEPTION 36
III.1.1 Positionnement *36*
III.2 PILE D'ANALYSE 37
Transformateur textuel : 39
III.3 IMPLEMENTATION 40
III.3.1 Lex et yacc *40*
III.3.1.1 Lex 41
III.3.1.2 Yacc 42
III.3.2 L'implémentation de la pile d'analyseur : *44*

CONCLUSION GENERALE **47**

BIBLIOGRAPHIE **48**

Liste des figures :

Figure 0.1: Un environnement pour RPR....10
Figure 1.1: Schéma du calcul réversible 13
Figure 1.2: Un automate réversible....13
Figure 1.3: Machine Turing réversible....14
Figure 1.4: Un automate cellulaire....15
Figure 1.5: Réversibilité causale cohérente (gauche) et hors causalité(droite)....15
Figure 2.1: graphe représente un réseau de Petri 21
Figure 2.2: le graphe de marquage de réseau de Petri de l'exemple (1)....23
Figure 2.3: un réseau avec un inverse pour toutes les transitions et son graphe d'accessibilité....23
Figure 2.4: un réseau avec une seule transition inverse et son graphe d'accessibilité 24
Figure 2.5: un réseau avec une seule transition inverse et son graphe d'accessibilité.....25
Figure 2.6: Un réseau réversible avec une seule transition inverse et son graphe d'accessibilité 25
Figure 2.7: Réseau de causalité disjonctive....28
Figure 2.8: Réversibilité dans l'interprétation des jetons individuels réseau 31
Figure 2.9: un système avec un stylo préassemblé 36
Figure 3.1: Environnement....39
Figure 3.2: Architecture de l'environnement 40
Figure 3.3: Architecture de la pile d'analyseur....41
Figure 3.4:grammaire de réseau de Petri reversible....41
Figure 3.5::Grammaire utilisé par le transformateur textuel 42
Figure 3.6: Exemple d'un graphe de marquage vers une sortie textuelle avec la grammaire fournie....42

Introduction générale

La thermodynamique est une branche de la physique qui étudie les propriétés thermiques et énergétiques de la matière. Elle examine les lois fondamentales qui régissent les processus énergétiques et les transformations de la matière. Les processus thermodynamiques peuvent être réversibles ou irréversibles, selon qu'ils peuvent être inversés ou non. Les processus réversibles sont théoriques et peuvent être inversés sans perdre d'énergie, ce qui signifie qu'un système qui fonctionne de manière réversible peut convertir l'énergie de manière très efficace, en minimisant les pertes d'énergie inutiles [1]. En revanche, les processus irréversibles sont associés à une perte d'énergie sous forme de chaleur et une augmentation de l'entropie. La modélisation de processus réversibles est devenue une question clé dans la conception de systèmes énergétiques plus efficaces.

Le calcul réversible comme paradigme est une approche de calcul qui permet d'effectuer des opérations réversibles, c'est-à-dire des opérations qui peuvent être annulées et dont l'état initial peut être récupéré à partir de l'état actuel d'un système. C'est un calcul dit non-conventionnel. Contrairement au calcul classique, où des informations sont perdues lors des opérations annulées, le calcul réversible vise à minimiser la perte d'information. Cette approche trouve des applications dans plusieurs domaines, notamment :

1. Informatique quantique : Dans le domaine de l'informatique quantique, le calcul réversible est essentiel pour minimiser la perte d'information lors des opérations sur les qubits, les unités de calcul quantique [2] [3]. Les algorithmes quantiques réversibles sont utilisés pour réaliser des calculs avec une consommation d'énergie minimale.
2. Circuits intégrés : Le calcul réversible est utilisé dans la conception de circuits intégrés basse consommation d'énergie, où la réduction de la dissipation d'énergie est un critère clé. Les circuits réversibles permettent de réduire la production de chaleur et d'optimiser l'utilisation de l'énergie [4].
3. Cryptographie : Dans le domaine de la cryptographie, le calcul réversible est utilisé pour des opérations de chiffrement et de déchiffrement, où la réversibilité des opérations est importante pour assurer la sécurité des données.
4. Biologie et nanotechnologie : Le calcul réversible trouve également des applications dans la modélisation de processus biologiques et dans la conception de systèmes nanotechnologiques, où la manipulation de l'information à l'échelle moléculaire nécessite des opérations réversibles [5] [6].
5. Systèmes transactionnels : L'utilisation du calcul réversible dans les systèmes transactionnels est un domaine de recherche relativement nouveau et en évolution. Le calcul réversible peut offrir certains avantages potentiels dans le contexte des systèmes transactionnels[7][8].
6. Systèmes d'exploration d'états : L'utilisation du calcul réversible dans les systèmes d'exploration d'états offre la possibilité de revenir en arrière et d'explorer différentes alternatives sans avoir à réexécuter l'ensemble des calculs. Cela peut améliorer

l'efficacité des algorithmes d'exploration d'états en réduisant la duplication des calculs et en permettant une exploration plus efficace des solutions.

En ce qui concerne les travaux récents sur l'utilisation du calcul réversible dans les systèmes d'exploration d'états, voici quelques avancées et recherches :

1. Recherche opérationnelle : Dans le domaine de la recherche opérationnelle, des travaux récents explorent l'utilisation du calcul réversible dans les algorithmes de recherche et d'optimisation [9] [10]. Par exemple, certains chercheurs ont proposé des extensions réversibles de l'algorithme A* pour la recherche de chemins dans les graphes, ce qui permet de revenir en arrière plus efficacement lors de l'exploration des alternatives.
2. Intelligence artificielle : Dans le domaine de l'intelligence artificielle, des chercheurs se sont intéressés à l'utilisation du calcul réversible dans les solveurs de problèmes et les algorithmes de planification [11] [12]. Par exemple, des approches basées sur le calcul réversible ont été proposées pour résoudre des problèmes de planification dans des environnements dynamiques où il est nécessaire de revenir en arrière et de réviser les plans en fonction des changements.
3. Méthodes de recherche heuristique : Des travaux récents explorent également l'utilisation du calcul réversible dans les méthodes de recherche heuristique [13] [14], telles que les algorithmes génétiques réversibles. Ces approches permettent de revenir en arrière dans l'espace de recherche pour explorer d'autres solutions et peuvent améliorer l'efficacité de la recherche de solutions optimales.
4. Mémoire transactionnelle réversible : Une autre direction de recherche concerne l'utilisation de la mémoire transactionnelle réversible [15] dans les systèmes d'exploration d'états. Cette approche permet de capturer et de restaurer des états précédents en utilisant des transactions réversibles, ce qui facilite l'exploration des alternatives et améliore l'efficacité des algorithmes d'exploration d'états.

Ces exemples représentent quelques-uns des travaux récents sur l'utilisation du calcul réversible dans les systèmes d'exploration d'états. Cependant, il convient de noter que ce domaine est encore en développement et que de nouvelles recherches sont nécessaires pour explorer davantage les possibilités et les limitations de cette approche.

En résumé, le calcul réversible est utilisé dans des domaines tels que l'informatique quantique, la conception de circuits intégrés, la cryptographie, la biologie et la nanotechnologie, où la réversibilité des opérations est importante pour minimiser la perte d'information et optimiser l'utilisation de l'énergie.

Les méthodes formelles sont un ensemble d'approches mathématiques et logiques rigoureuses utilisées pour spécifier, concevoir et vérifier les systèmes informatiques. Lorsqu'il s'agit de calcul réversible, les méthodes formelles peuvent être appliquées pour garantir la qualité d'un système réversibilité. Les réseaux de Petri, une méthode de modélisation basée sur des graphes, peuvent être utilisés pour modéliser des processus de manière réversible. En effet, la réversibilité de réseau de Petri permet d'identifier les processus réversibles. [16] [17] sont les deux récents travaux sur l'utilisation du calcul réversible via réseaux de Petri. Ils ont fait appel à la sémantique de causalité à fin d'assurer le calcul réversible

Développer un environnement formelle adoptant le calcul réversible via réseaux de Petri est quasiment indispensable pour aller plus loin dans cet axe de recherche. Dans ce présent projet en va concevoir sa première pierre qui consiste à développer un module (compilateur) qui prend en entrée une description textuelle d'un Réseau de Petri Réversible (RPR) et la

transformes en structure de données spécifique pour être utilisée ensuite par d'autre module de l'environnement (Figure 0.1). Précisément, en va (i) proposer une grammaire adéquate aux réseaux de Petri réversibles (ii) développer l'analyseur lexical (iii) développer l'analyseur syntaxique.

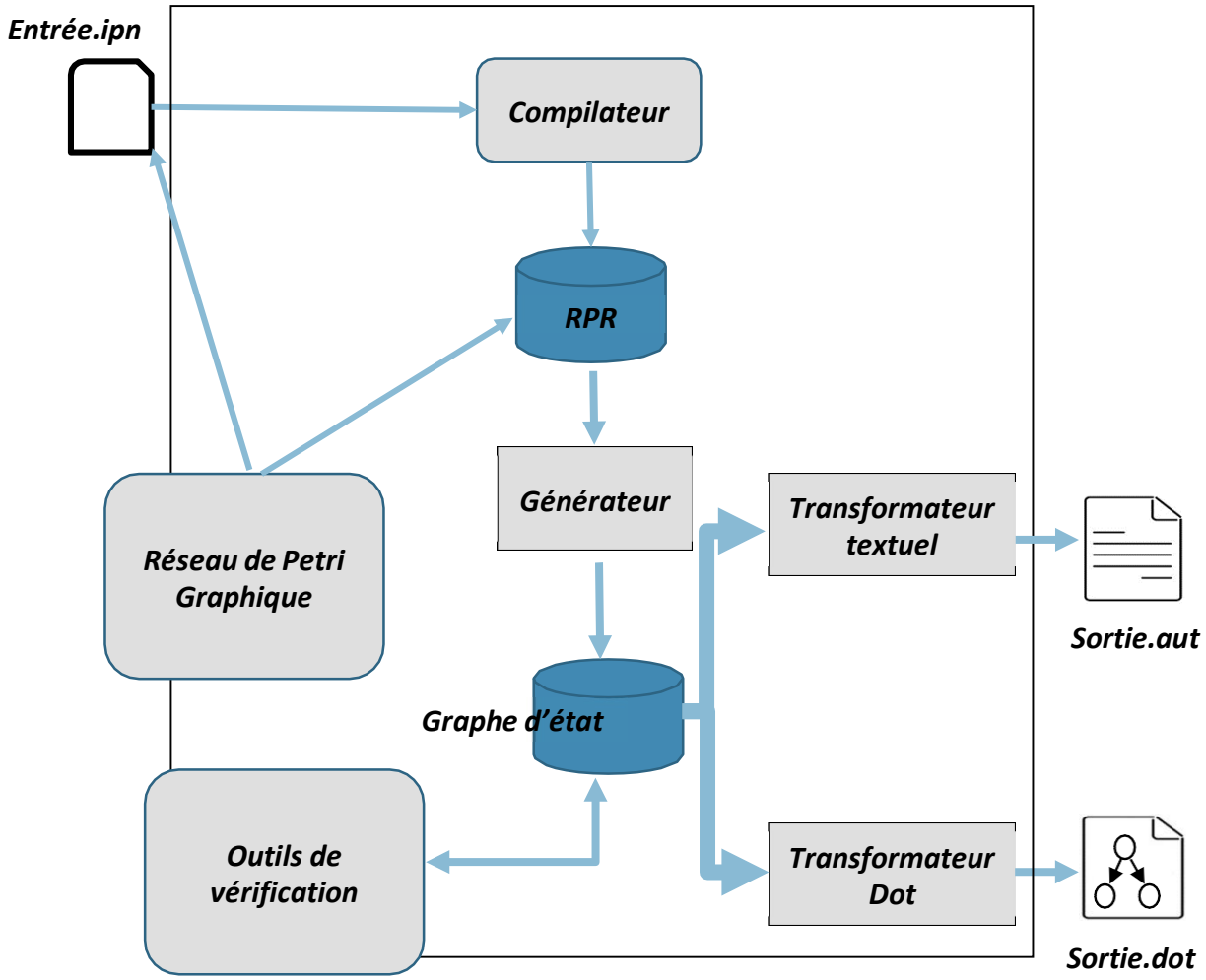


Figure 0.01: Un environnement pour RPR

Ce document est organisé comme suite. Le premier chapitre met en évidence les avancées scientifiques récentes les plus importantes dans le domaine du calcul réversible formel. En détails, le chapitre deux présente le réseau de Petri réversible ainsi sa sémantique adjacente. Dans le troisième chapitre, la conception et l'implémentation de notre module sont données. Le document est clôturé par une conclusion générale et des perspectives.

Chapitre 1

Le Calcul réversible

I.1 Le calcul réversible

Le calcul réversible permet d'utiliser à la fois les directions avant et arrière conventionnelles, permettant la récupération des états précédents ainsi que le calcul des entrées à partir des sorties. Des applications aussi diverses que le calcul basse consommation la simulation, la robotique et le débogage ont toutes montré un intérêt pour le calcul réversible.

De telles applications, cependant, nécessitent une solide compréhension des principes sous-jacents à l'informatique réversible. De nombreux aspects théoriques de l'informatique réversible ont été étudiés au fil des ans, y compris les fondements catégoriels de la réversibilité, les fondements des langages de programmation et la réécriture de termes, tout en prenant également en compte différents modèles d'informatique :

- Séquentielle
  - Machines de Turing
  - Automates
- Concurrente
  - Automates cellulaires
  - Processus de calculs
  - Calcul membranaire
  - Réseaux de Petri

Et résoudre les difficultés soulevées par le traitement quantique, qui est pour la plupart intrinsèquement réversible.

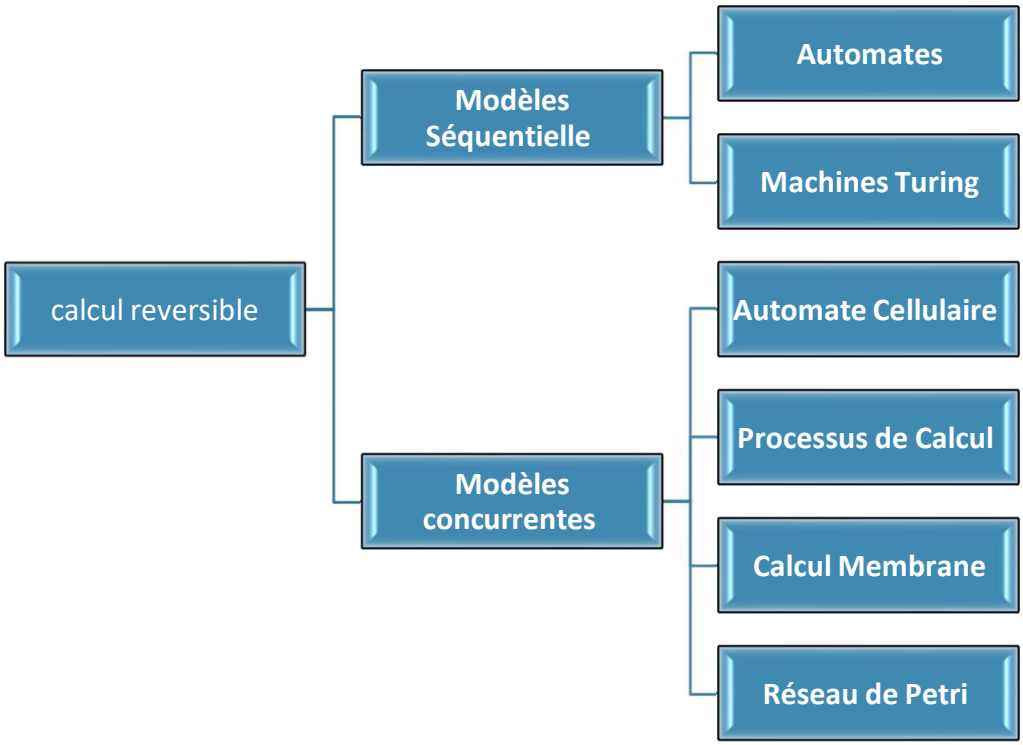


Figure 1.02: Schéma du calcul réversible

I.1.1. Modèles Séquentielle

I.1.1.1 Automates

L'étude des machines abstraites, ou automates, en tant que modèles mathématiques computationnels est connue sous le nom de théorie des automates. Ils aident à comprendre les limites du calcul et l'influence de diverses ressources, y compris le temps et l'emplacement, sur la puissance de calcul. Des exemples de classes d'automates largement étudiées comprennent les automates cellulaires, les automates à pile, la machine à compteur, les machines de Turing et automates finis. Si l'information est suffisamment préservée pour que le calcul d'un automate puisse être retracé dans le temps, l'automate est dit réversible. La réversibilité est prise en charge par tous les types d'automates susmentionnés.

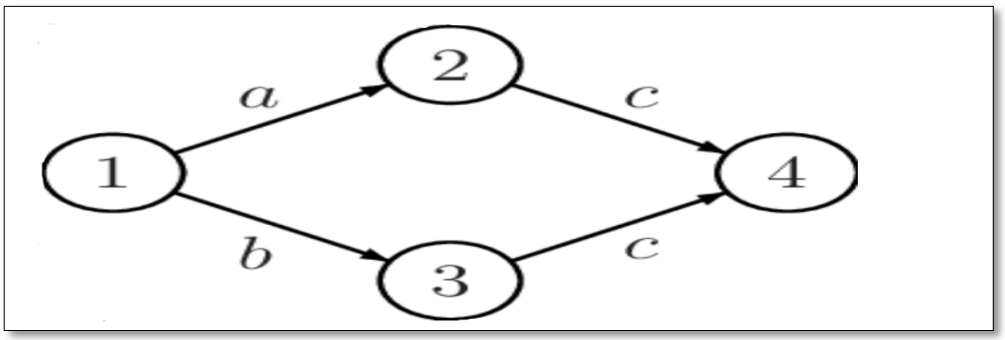


Figure 1.3 : un automate réversible

I.1.1.2 Machines Turing

La tête d'une machine de Turing (TM), un modèle informatique traditionnel, parcourt une bande bi-infinie de cellules contenant chacune un symbole de bande. Selon une règle de transition définie, le programme du TM, la tête lit et écrit des symboles sur la bande, modifie son état interne et avance vers les cellules voisines à des pas de temps discrets. La machine est réversible si le programme est bien choisi (RTM).

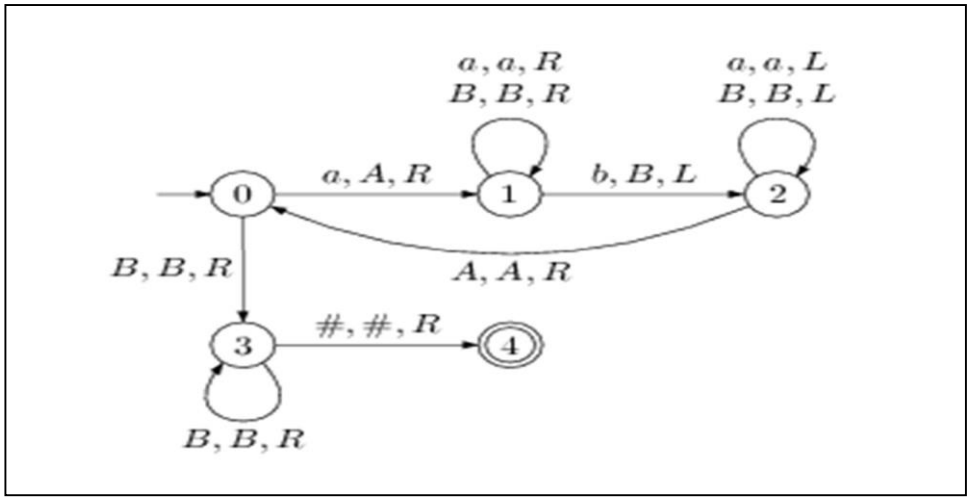


Figure 1.4: Machine Turing réversible

I.1.2. Modèles concurrentes

I.1.2.1 Automate Cellulaire

Un système dynamique sur une grille infinie de cellules connu sous le nom d'automate cellulaire (CA) est déterminé par une règle de mise à jour locale qui est mise en œuvre simultanément à chaque cellule. Plus précisément, dans l'arrangement rectiligne à d dimensions typique, chaque cellule stocke une composante de l'ensemble d'états finis A. Les cellules sont les éléments de Z. La dynamique globale c → c' peut être rendue conservatrice d'informations en sélectionnant la règle de mise à jour dans cette situation, un automate cellulaire inverse le calcul, et l'automate est dit réversible (RCA).

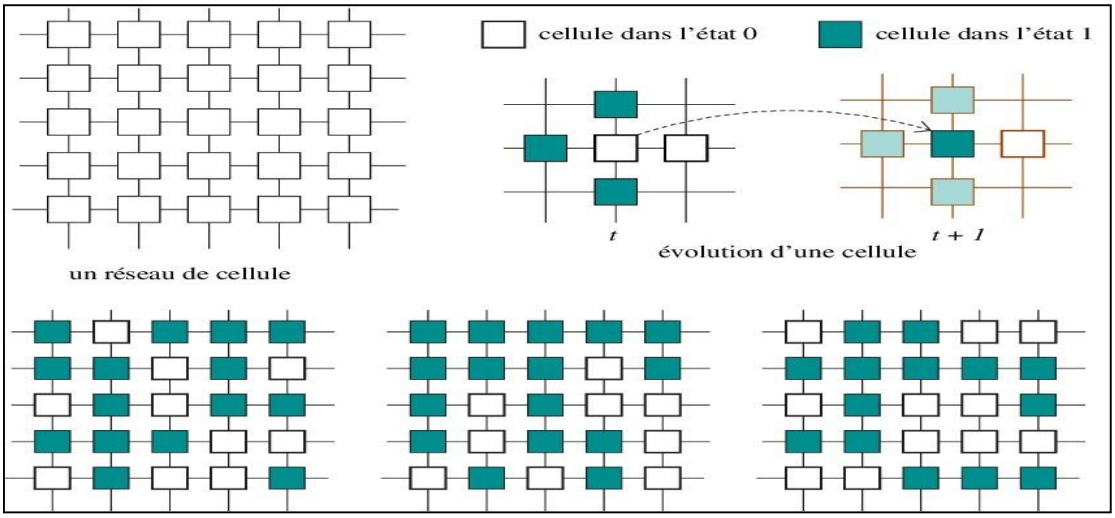


Figure 1.5: Un automate cellulaire

I.1.2.2 Processus de Calcul

Une classe de modèles algébriques pour les systèmes concurrents et distribués est connue sous le nom de calculs de processus. Les calculs de processus permettent d'expliquer succinctement le comportement d'un système concurrent, en faisant abstraction des spécificités de l'implémentation et en se concentrant sur les modèles d'interaction entre les parties du système. En conséquence, il est possible d'expliquer mathématiquement avec précision le comportement d'un système, et des outils de vérification peuvent être construits juste en plus de cela. La Chemical Abstract Machine, un calcul inspiré des processus chimiques dont la sémantique opérationnelle fournit à la fois des relations de réduction directes et inverses, peut représenter l'origine de la recherche sur les calculs de processus d'inversion.

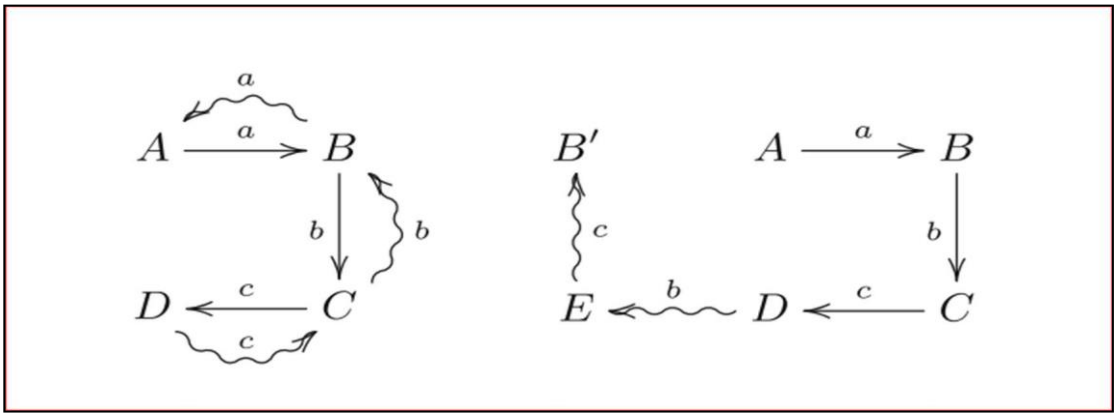


Figure 1.6: Réversibilité causale cohérente (gauche) et hors causalité(droite)

I.1.2.3 Calcul Membrane

Naturelle est un domaine d'étude complexe qui traite des modèles et des méthodes de calcul tirés du monde naturel et qui aident à notre connaissance de la façon dont le monde biologique traite l'information. Deux idées informatiques naturelles importantes qui s'inspirent des opérations des cellules vivantes sont l'informatique membranaire et les systèmes de réaction. L'intérêt pour le sujet de la réversibilité dans plusieurs paradigmes informatiques s'est accru récemment. L'une des premières publications sur la réversibilité dans les systèmes membranaires [5] proposait la réversibilité comme un type de dualité sous l'influence de la théorie des catégories.

I.1.2.4 Réseau de Petri

Les réseaux de Petri sont un formalisme mathématique permettant de modéliser et de raisonner sur des systèmes concurrents. Dans la plupart des cas, les réseaux de Petri sont des quadruplets contenant deux ensembles finis, d'éléments actifs (actions/transitions) et statiques (lieux), qui sont reliés par une fonction de flux (ou relation) dont l'état initial est donné par des jetons disséminés sur les lieux. Les stratégies basées sur l'action et sur l'état sont toutes deux prises en charge par les réseaux de Petri. La réversibilité a toujours été un concept clé dans les réseaux Petri Fondations of Réversible Computation.

I.2 Le calcul réversible et réseau de Petri

Le terme « réversibilité » a d'abord été utilisé pour décrire les réseaux dans les années 1970 où chaque transition avait un inverse. Une telle notion de réversibilité locale est assez similaire à celle qui est maintenant employée dans d'autres domaines, comme les calculs de processus ou les langages de programmation [19], et parfois encore utilisé pour décrire le réseau inverse. Par rapport à la situation générale des réseaux bornés, certains problèmes de décision dans les réseaux de Petri symétriques bornés ont une complexité temporelle réduite.

La réversibilité dans les réseaux de Petri fait également référence à la cyclicité, ce qui est une autre signification. [20] offre une perspective globale et exige que l'état de départ du réseau soit accessible à partir de tout autre état atteignable. [21] Outre l'idée locale de réversibilité susmentionnée, les réseaux de Petri sont appelés symétriques. [22] notion de réversibilité locale a été investiguée. On peut diviser les contributions proposées en trois fils principaux : deux d'entre eux examinent comment inverser une seule transition dans un réseau de Petri, permettant d'utiliser, respectivement, une seule transition inverse ou un ensemble d'inverses. Le dernier fil se concentre sur la modélisation de la sémantique réversible dans des modèles spécifiques basés

sur les réseaux de Petri.

Une approche pour inverser une seule transition à l'aide d'un seul inverse a été étudiée à la fois sous la sémantique séquentielle et sous la véritable sémantique concurrente. Un autre fait important montré est lié à la cyclicité : l'introduction d'un inverse strict dans un réseau cyclique peut modifier l'ensemble des marquages accessibles[23].

Le problème susmentionné de déterminer si l'ajout d'une inversion stricte pour une seule transition modifie l'ensemble des marqueurs accessibles dans un réseau peut être résolu pour des réseaux limités.

À la lumière de cela, une question plus générale pourrait être posée : est-il possible d'inverser la transition indiquée tout en exigeant simplement que le réseau résultant et celui quia été fourni aient un comportement isomorphe (c'est-à-dire un graphe d'accessibilité isomorphe)et permettant de modifier la structure du réseau !

En utilisant des outils de théorie des régions bien connus, la question a été résolue. [24] Il est possible de combiner des solutions indépendantes pour diverses transitions pour résoudre un problème dans certains systèmes de transition, qui sont des graphes d'accessibilité d'un réseau de Petri limité. Dans ces systèmes, les transitions ne peuvent pas être inversées par des inversions rigoureuses [23].

Les transitions ne peuvent pas toujours être inversées par des inversions serrées, même dans la situation exceptionnelle des systèmes de transition linéaires sur des ensembles binaires d'actions. La tâche de déterminer si l'ajout d'une inversion stricte pour une seule transition modifie l'ensemble des marqueurs accessibles dans de tels systèmes a une complexité temporelle linéaire [25].

Les réseaux d'occurrences, qui sont des réseaux 1-sûrs et acycliques sans conflits en arrière et où des inversions rigoureuses peuvent toujours être utilisées, sont un autre exemple spécifique de réseaux bornés. Expliqué par la suite, cette caractéristique des réseaux d'occurrences et de leurs extensions infinies a été utilisée comme étape intermédiaire. [26]

Les systèmes sous sémantique concurrente d'exécution d'action sont pris en compte dans une autre branche d'étude sur les inverses stricts.

Ces systèmes permettent l'exécution simultanée de nombreuses actions, y compris plusieurs instances d'une même action (auto-concurrence) [25].

Dans un contexte plus large, il est possible de créer une série d'inversions ayant l'effet

inverse, appelées inversions d'effet, afin d'inverser une seule transition. Trouver un réseau de Petri limité avec un comportement isomorphe et des transitions réversibles devient toujours résolu dans ce scénario. En conséquence, dans ce contexte, plusieurs systèmes où les inverses rigoureux ne pouvaient pas être utilisés pour inverser les transitions sont devenus réversibles. De plus, il ne coûte pas cher de préparer un réseau limité pour l'inversion par des ensembles d'inversions d'effets ; il suffit de convertir le réseau d'origine en son homologue complémentaire, qui fait deux fois la taille de l'ensemble des emplacements [23].

Les réseaux illimités font l'objet d'une tentative similaire dans [23]. Il existe des réseaux illimités qui ne peuvent pas être inversés même si leurs transitions utilisent des ensembles

illimités d'inversions d'effets. Cependant, les ensembles finis sont suffisants si cela est faisable. Trouver un réseau potentiellement entièrement nouveau avec un comportement isomorphe quipeut être inversé a été simplifié pour consister à étendre le réseau actuel par des emplacements supplémentaires qui n'empêchent aucune transition dans n'importe quel état accessible et à déterminer s'il existe des paires d'états problématiques.

Il y a une structure forte et un ordre partiel naturel dans ces appariements d'états problématiques. Bien que la collection de toutes les paires minimales d'états problématiques pour un système donné soit limitée, il n'est pas élémentaire de déterminer si deux états donnés constituent une paire problématique, et il est insoluble de déterminer s'il existe au moins une telle paire [18]. Les extensions des réseaux de Petri avec une réversibilité locale causale cohérente sont étudiées dans un domaine d'étude distinct [23].

Le modèle d'inspiration biologique des réseaux de Petri inversés est une autre façon d'examiner la réversibilité locale causale cohérente ainsi que la réversibilité locale désordonnée. Dans [27], Il existe des jetons, qui sont des bases durables reliées par des liens qui sont déplacés par les changements de réseau. Le principal inconvénient de l'approche est que le réseau ainsi décrit doit être limité et acyclique. D'autre part, les réseaux de Petri inversés peuvent être encodés comme des réseaux de Petri colorés avec une palette de couleurs limitée [28], les systèmes conventionnels de transition de lieu délimité également. De plus, le problème de sélection d'antenne distribuée a été résolu efficacement à l'aide de réseaux de Petri inversés .[29]

Un modèle populaire de concurrence est le réseau de Petri, qui offre l'environnement parfait pour étudier les principes fondamentaux des systèmes concurrents. Par conséquent, on peut dériver la sémantique réversible d'un réseau P/T à partir de la sémantique de son déploiement. On peut dériver la sémantique réversible d'un réseau P/T à partir de la sémantique de son déploiement. Etant donné N comme un IPN, la cohérence de N est garantie en ajoutant une règle de déclenchement intuitive pour annuler les transitions conformément au lien de

causalité ; par conséquent, l'ensemble de tous les états atteignables de N dans la version réversible et celui de l'original sont les mêmes. De plus, la réversibilité de N est polyvalente, permettant d'accéder à leur état de départ en sens inverse à partir de n'importe quel état.[30].

Après cette étude alors la réversibilité de réseau de Petri rencontre des problèmes Tel une modification statique peut avoir un impact sévère sur le comportement du système, par exemple, le problème d'établir si le réseau modifié a le mêmes états que l'original est indécidable, une transition parallèle qui pas nous nous concentrons donc sur des réseaux avec des espaces d'états finis et montrent, en particulier, que chaque transition dans de tels réseaux peut être inversée en utilisant un ensemble approprié de nouvelles transitions. La solution alors c'est l'utilisation de la sémantique de causalité.

I.3 Conclusion

Dans ce chapitre, nous avons introduire les différents modelés utilisés pour représenter le calcul réversible à savoir automate cellulaire, processus de calcul, calcul membrane et les réseaux de Petri. Nous avons montré que la réversibilité via les réseaux de Petri est actuellement en plein d'exploration.

Chapitre 2

Réseaux de Petri Réversibles

I.1 Le réseau de Petri

En 1962, à l'université technique de Darmstadt, en Allemagne de l'ouest, un certain jeune chercher, nommé Carl Adam Petri, a présenté dans sa thèse un nouveau formalisme qui a pris par la suite son nom « les réseaux de Petri ». Un peu plus tard et au début des années 70, ce formalisme a été utilisé par Anatol W. Holt, F. Commoner, M. Hack et leurs collègues dans le groupe de recherche de MIT (Massachussetts Institute Of Technologie) et grâce à eux ce modèle a marqué ses qualités. En effet, depuis ce temps-là, les européens se sont enfoncés au fond dans l'organisation d'atelier et la publication d'actes de congrès sur les réseaux de Petri.

En 1975 MIT organise la première conférence sur les réseaux de Petri et les méthodes relationnelles. Par la suite, J. Peterson a publié en 1981, le premier ouvrage sur les réseaux de Petri.

II.1.2. Définitions fondamentales

II.1.2.1. Définition informelle

Le réseau est un graphe biparti ayant deux types de nœuds :

- Les places (S) qui représentent les états du système modélisé, elles sont représentées par des cercles.
- Les transitions (T) qui représentent les évènements ou les actions qui causent le changement de l'état, elles sont représentées par des rectangles.

Les places et les transitions sont reliées par des arcs évalués. Il n'y a pas d'arcs entre deux nœuds de même type.

La valeur 1 étant omise et l'absence d'arcs indiquant la valeur 0. Un réseau de Petri est un graphe muni d'une sémantique opérationnelle, c'est-à-dire un comportement associé au graphe, ce qui permet de décrire la dynamique du système représenté. Pour cela les jetons sont ajoutés aux places. Ces jetons sont représentés soit par des points, soit par des nombres à l'intérieur des places. Une répartition des jetons dans les places à un instant donné est appelée marquage du réseau de Petri.

I.1.2.2 Définition formelle

Définition 2.1. Un réseau de Petri est un tuple (S, T, F, I, L) avec
- S et T deux ensembles disjoints de places et de transitions,
- $F : (S \times T \cup T \times S) \rightarrow \mathbb{N}$, la relation de flux,
- $I : S \rightarrow \mathbb{N}$, le marquage initial,
- et $l : T \rightarrow A$, pour A un ensemble d'actions, la fonction de labellisation.

Les réseaux de Petri sont représentés en dessinant les emplacements sous forme de cercles et les transitions sous forme de boîtes contenant leur étiquette. Il existe $F(s, t)$ arcs de x vers y pour le couple $x, y \in S \cup T$. Lorsqu'un réseau de Petri représente un système concurrent, la fonction $M : S \rightarrow \mathbb{N}$ est utilisée pour représenter l'état global du système. Un tel état est

représenté en plaçant $M(s)$ points (jetons) à chaque emplacement. La marque I indique l'état initial. Le comportement d'un filet est décrit en définissant la relation entre les étapes de transition des marquages.[31]

Exemple 1 : le graphe dans la Fig. 1 représente un réseau de Petri N = (S,T,I), ou :

S= *{*s1,s2, *s3, s4}*

T ={t1,t2, t3,t4}

I_0*={2,0,0,0,0}.*

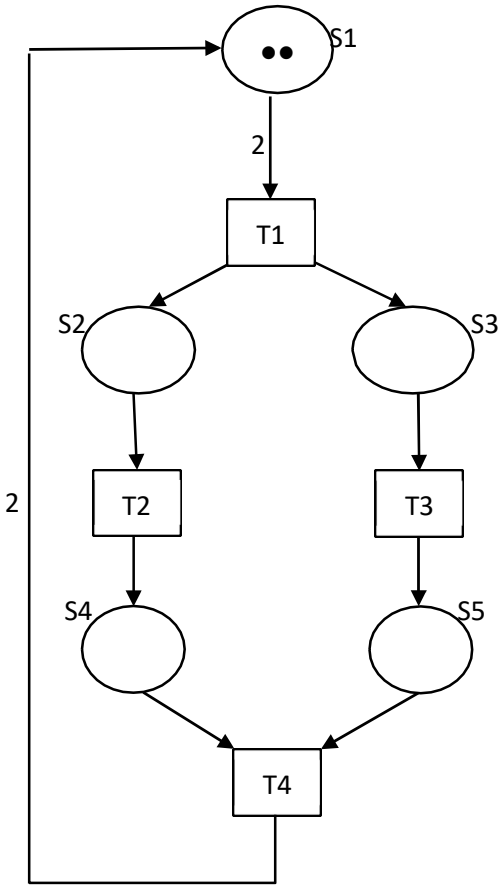


Figure 2.1: graphe représente un réseau de Petri

Definition 2.2. Un multi-ensemble sur un ensemble S est une fonction de la forme $M : S \rightarrow N$, ou $M \in \mathbb{N}^s$. Ecrire $M \leq N$ si $M(s) \leq N(s)$ pour tout $s \in S$ pour les multi-ensembles M et N sur S. Le multi-ensemble $M + N \in \mathbb{N}^s$ est celui où $(M + N)(s) = M(s) + N(s)$ et la fonction $M - N$ est celle où $(M - N)(s) = M(s) - N(s) -$ ce n'est pas toujours un multi-ensemble. Le multi ensemble vide est donné par la fonction $0 : S \rightarrow \mathbb{N}$ donnée par $0(s) = 0$. Le groupe $\{s \in S \mid M(s) = 1\}$ permet d'identifier un multi-ensemble $M \in \mathbb{N}^s$ avec $M(s) \leq 1$ pour tout $s \in S$. Un multi-ensemble M sur S est complet si $\{s \in S \mid M(s) > 0\}$. C'est $M(S)$ l'ensemble des multi-ensembles complétés sur S.

Definition 2.3. Pour un multi-ensemble fini $U : T \to \mathbb{N}$ de transitions dans un réseau de Petri, soit ${}^\bullet U, U^\bullet : S \to \mathbb{N}$ les multiensembles de places d'entrée et de sortie de U, donnés par :
${}^\bullet U(s) = \sum_{t \in T} F(s,t) \cdot U(t)$ et $U^\bullet(s) = \sum_{t \in T} U(t) \cdot F(t,s)$ pour tout $s \in S$.

U est activé sous un marquage M si ${}^\bullet U \leq_U M$. Dans ce cas U peut tirer sous M donnant le marquage $M' = M - {}^\bullet U + U^\bullet$, noté $M \to M'$.

Un jeton est déplacé le long de cet arc de s vers t pour chaque transition t dans U et chaque arc du multi-ensemble U de transitions déclenchées. Ces jets sont consommés par le tir, mais de nouveaux jets sont également produits, un pour chaque arc qui émerge de la cible. Ils peuvent être trouvés dans les zones à la fin de ces arcs. Si cela se produit plusieurs fois aux États-Unis, tout le reste se produit plusieurs fois. Il n'est possible que U soit abattu que s'il y a suffisamment de jetons dans ses espaces réservés.
Les termes S^N, T^N, F^N, I^N *et* l^N, qui s'appliquent également à d'autres structures fournies sous forme de tuples, sont utilisés pour identifier les parties d'un réseau N. Lorsqu'elles ressortent du contexte. Il n'y a pas d'indice N.

Deux réseaux P et Q sont isomorphes, notés $P \cong Q$, s'ils ne diffèrent que par les noms de leurs places et transitions, soit s'il y a des bijections: $\eta: T^P \to T^Q$ et $\beta: S^P \to S^Q$: TPT tel que, pour $s \in S^P et\ t \in T^P$: $I^Q(\beta(s)) = I^P(s) : F^Q(\beta(s), \eta(t)) = F^P\ (s,t), = F^Q(\eta(t), \beta(s)) = F^P(t,s)\ et\ l^Q(\eta(t)) = l^P(t)$.

II.2 Graphe de marquage

Le graphe de marquage est un graphe orienté qui représente les nombreux états atteignables dans un réseau de Petri à partir d'un marquage initial donné ainsi que les transitions permettant à un état de céder la place à un autre. Il peut être utilisé pour examiner les caractéristiques de comportement du réseau de Petri, telles que la vivacité, l'atteignabilité, la répétabilité, etc., et analyser le comportement dynamique du système.

Les sommets du graphe de marquage représentent les différentes répartitions de jetons qui peuvent être réalisées sur le réseau, ou les différents marquages du réseau de Petri. Les transitions du réseau de Petri activées par les marquages et permettant le passage d'un marquage à l'autre sont représentées par les arcs du graphe de marquage. Lorsqu'une transition peut être franchie pour passer d'une marque à une autre en respectant le Petri les règles de franchissement des transitions du réseau, un arc est dirigé d'un pic à l'autre.

A partir de l'unique marquage de départ $M0$, on construit progressivement les différents arcs et nœuds pour créer le graphe de marquage d'un RdP. Pour chaque nouveau marquage Mi, le nombre de transitions pouvant être traversées est déterminé, et un arc est ajouté au nouveau marquage pour chaque transition du groupe. Si un balisage est déjà présent dans le graphe, il suffit de tracer l'arc vers lui.[32]

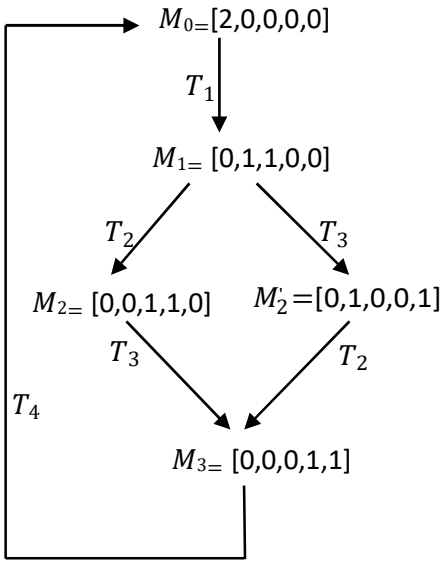


Figure 2.2: le graphe de marquage de réseau de Petri de l'exemple (1)

II.3 La réversibilité dans le réseau de Petri

Cette section expliquera comment l'ajout de transitions inverses affecte le comportement net avec l'exemple mentionné dans l'article de réversible computation vs. Reversibility in Petri nets.

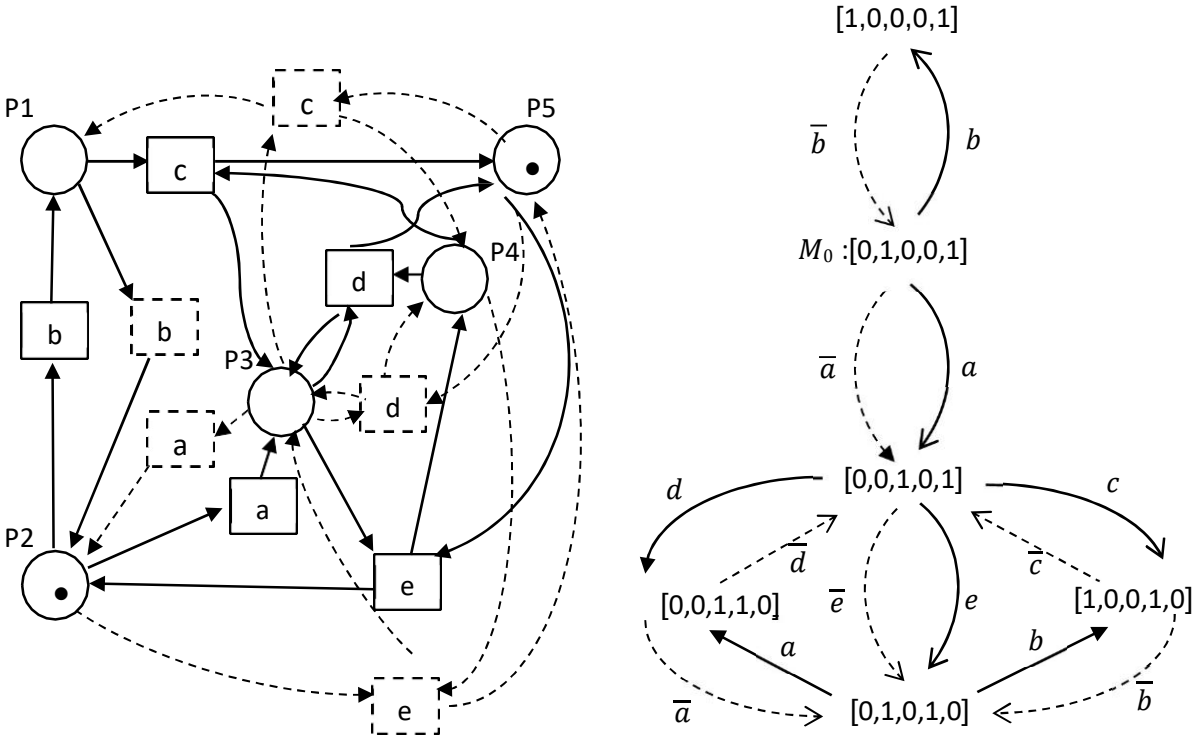


Figure 2.3: un réseau avec un inverse pour toutes les transitions et son graphe d'accessibilité.

La figure 2.3 montre un réseau et son accessibilité en traits pleins. Le diagramme, selon Morcover, illustre l'ajout de la transition inverse au premier réseau et la croissance résultante du graphe d'accessibilité initial. Si le réseau modifié est réversible et possède le même ensemble de marquages accessibles que le net d'origine, le réseau d'origine n'était pas réversible. [28]

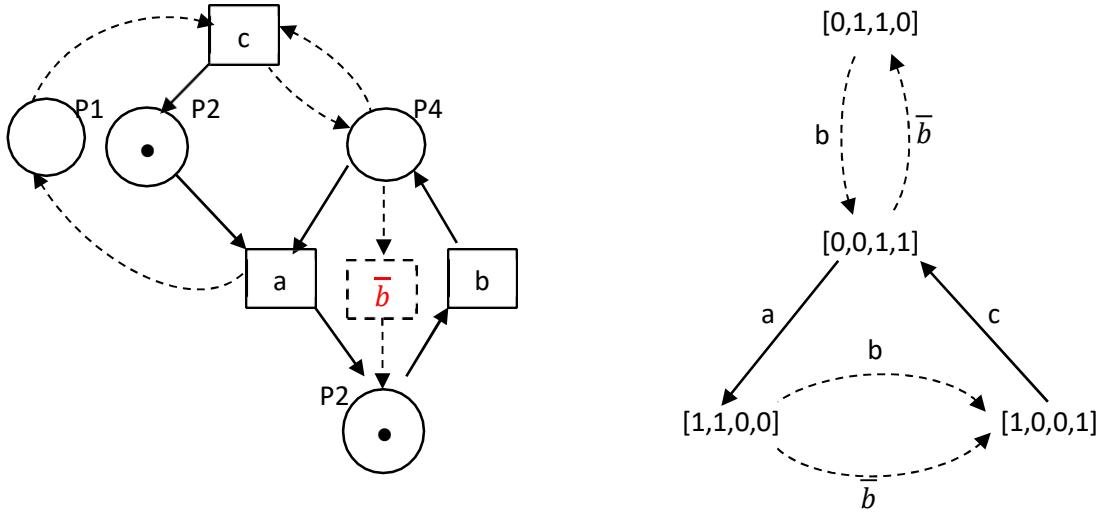


Figure 2.4: un réseau avec une seule transition inverse et son graphe d'accessibilité.

Un réseau avec l'état d'origine [0,0, 1,1] est représenté sur la figure 2.4. Dans ce cas, tout ce qui est nécessaire pour créer un réseau réversible est d'ajouter un inverse $\overline{\boldsymbol{b}}$ à la transition b. De plus, le groupe de marquages accessibles ne change pas.

L'ajout de transitions inverses peut occasionnellement améliorer le comportement du réseau d'origine, comme indiqué dans les deux premiers cas. Cependant, en Général, L'inversion de la transition modifie l'ensemble des accessibilités et permet des calculs basés sur les transitions initiales qui n'étaient pas possibles dans le premier réseau.

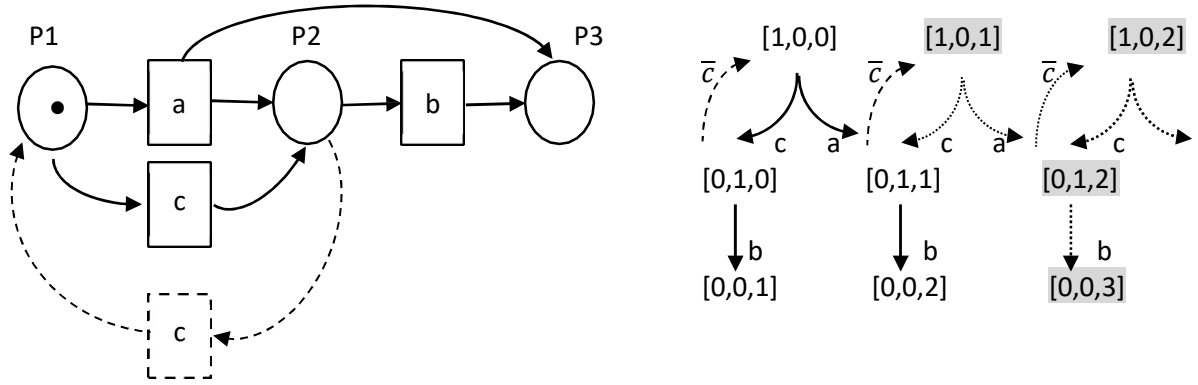


Figure 2.5: un réseau avec une seule transition inverse et son graphe d'accessibilité

Dans l'exemple de la Figure 2.5, les arcs pleins désignent les arcs présents dans le graphe d'accessibilité du réseau d'origine ; le pointillé les arcs indiquent l'inverse introduit de c activé aux marques accessibles dans l'original filet ; et les arcs en pointillés représentent les transitions (ou inversions) activées uniquement aux marquages (sur fond gris) qui n'étaient pas accessibles dans le réseau d'origine.

Un réseau avec un nombre limité de repères accessibles est illustré à la figure 2.5; en ajoutant un seul $\overline{c}$, l'ensemble d'accessibilité est augmenté à un ensemble infini. Gardez à l'esprit qu'à [0,1,1], l'exécution de la transition inverse $\overline{c}$ est activée avant la première exécution de c.

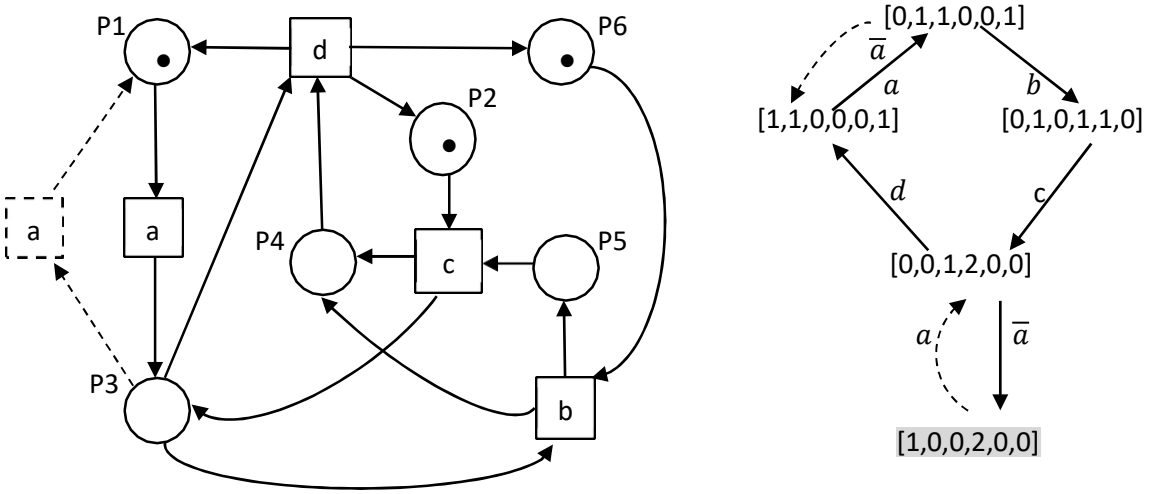


Figure 2.6: Un réseau réversible avec une seule transition inverse et son graphe d'accessibilité.

En conséquence, ce réseau P/T simulerait un système dans lequel certaines actions peuvent être inversées avant qu'elles aient été effectuées, ce qui va à l'encontre pour inverser les calculs. Comme le montre la Figure 2.6. L'état d'origine du réseau est [0,1,0,1, 1,0], mais une fois de plus, l'ajout d a donné un réseau avec un plus grand ensemble de marqueurs accessibles.

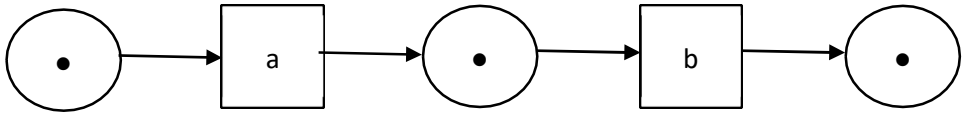
Selon les exemples susmentionnés, il n'est pas toujours évident de savoir quand on peut ajouter des revers à un réseau sans modifier de manière significative son comportement. Nous pourrions également observer que même une telle transition supplémentaire peut modifier de manière significative le comportement global. Par conséquent, il est essentiel de pouvoir déterminer si un inverse spécifique peut être ajouté à un réseau sans modifier de manière significative son ensemble d'accessibilité.

II.4 Sémantique de causalité

II.4.1 Les interprétations individuelles et collectives des jetons

Dans l'interprétation des jetons individuels des réseaux de Petri, divers jetons hébergés au même endroit sont distingués tandis que leurs origines sont enregistrées. Si une transition s'exécute en utilisant un jeton créé par une autre transition. Entre les deux, il existe une relation causale. Par conséquent, un ordre partiel peut toujours être utilisé pour représenter les relations causales entre les transitions dans un réseau. D'autre part, dans l'interprétation des jetons collectifs, les jetons ne sont pas distinguables ; par exemple, s'il y a deux jetons à un emplacement, tout ce qui s'y trouve est le numéro 2. En conséquence, des liens de causalité plus nuancés qui ne peuvent pas être représentés par des ordres partiels émergent entre les transitions d'un parcours d'un réseau. La différence entre les deux interprétations est démontrée par l'exemple qui suit.[31]

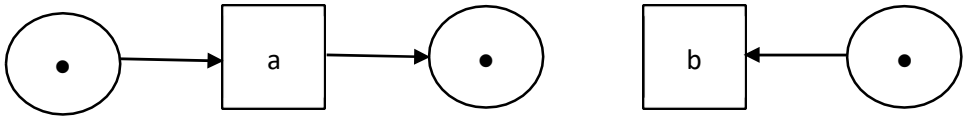
A:



Les transitions désignées par a et b dans ce réseau peuvent toutes deux se déclencher une fois. Il y a deux jetons au milieu après que a a tiré. Lequel de ces jetons est utilisé pour tirer b importe en termes de philosophie de jeton individuel. Les transitions a et b sont causalement indépendantes si le jeton qui était précédemment présent est utilisé. B est causalement dépendant de a si le jeton qui a été créé par an est utilisé. Les commandes partielles $\begin{smallmatrix} a \\ b \end{smallmatrix}$ et $a \rightarrow b$ peuvent être utilisées pour décrire les deux exécutions maximales du réseau A ci-dessus. D'autre part, le nombre 2 est le seul élément présent dans l'espace du milieu après l'apparition de a, selon la théorie des jetons collectifs.

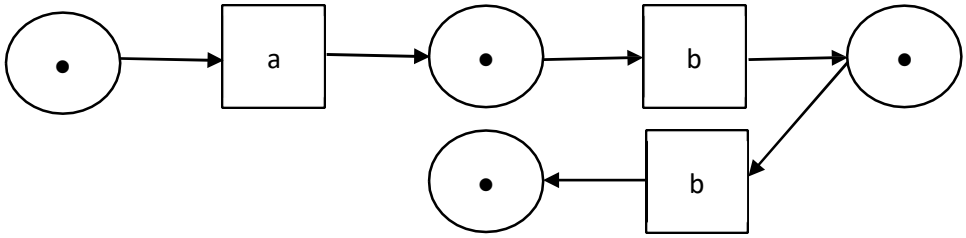
Ce qui suit illustre que les deux philosophies produisent des notions incomparables d'équivalence.

B:



L'exigence de b énoncée par la position au centre est superflue dans la philosophie symbolique collective, donc A doit être comparable à B. Mais parce que B n'a pas l'exécution $a \to b$, et ne sont pas entièrement équivalents de bisimulation simultanée. La bisimulation complètement concurrente de A, en revanche, est identique à C ci-dessous.

C:



Le réseau d'occurrences dérivé de A par dépliage [4,8] est en réalité C. Les exécutions $a \to b$ et $\begin{smallmatrix} a \\ b \end{smallmatrix}$ sont toutes deux incluses dans la philosophie des jetons individuels pour A et C. Cependant, selon la philosophie des jetons collectifs, A ne peut pas être comparable à C de quelque manière que ce soit qui préserve la causalité puisqu'il n'a pas de course $a \to b$.

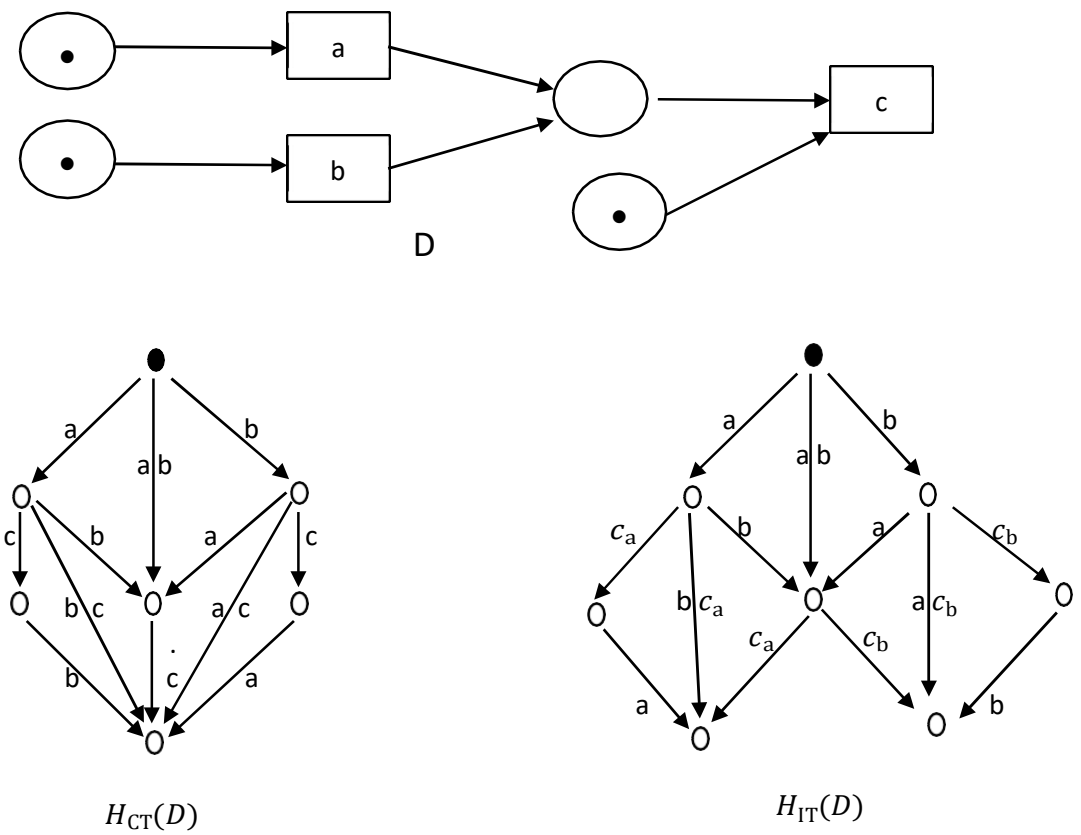


Figure 2.7: Réseau de causalité disjonctive

Le réseau de Petri *D* ci-dessus, sert d'exemple de la façon dont l'interprétation collective des jetons entraîne des liens de causalité qui ne peuvent pas être décrits par des ordres partiels. Ce réseau présente une causalité disjonctive selon l'interprétation symbolique collective : ce dépend causalement de $a \vee b$. En revanche, *D* est autorisé dans deux exécutions sous l'interprétation individuelle du jeton, l'une dans laquelle c ne dépend que de a et l'autre dans laquelle ce dépend uniquement de b.

En défendant l'interprétation collective de ce réseau, Antoni Mazurkiewicz a pris l'exemple de deux écoliers contribuants chacun £1 pour offrir un cadeau à leur professeur. ce représente l'acte d'acheter le cadeau, qui ne coûte que £1. Contrairement à l'interprétation symbolique collective, qui ne permet qu'une seule exécution complète dans laquelle l'achat du cadeau est causé par la disjonction des deux contributions, l'interprétation symbolique individuelle suggère que le cadeau est acheté avec une contribution soit d'un enfant soit de l'autre.

II.4.2 Une règle de tir pour l'interprétation individuelle des jetons

La définition normale d'un marquage et la règle de tir associée représentent l'interprétation symbolique collective par opposition à l'individuelle. Dans l'article « the individual and collective token interpretation of Petri Nets »[33], Rob van Glabbek a réinterprété ces idées d'une manière qui intègre l'interprétation subjective de chaque jeton. Pour

ce faire, ils ont décrit l'idée d'un jeton tel qu'il peut apparaître dans un réseau de Petri d'une manière qui donne à chaque occurrence potentielle de jeton un nom unique. Ils ont utiliser $t^{'} = *$ pour les jetons qui sont initialement dans s. Lorsque dépose s en n jetons, n est la quantité de jetons. Ces jetons se distinguent par les entiers ordinaux $k = 0, 1, 2, \ldots, n-1$. Ils ont spécifié simultanément les tirs de transition afin de définir les. Ce seront des paires (X, t) où X est l'ensemble des jetons consommés lors du déclenchement de la transition et t est la transition qui se déclenche. Les transitions qui peuvent s'exécuter plusieurs fois sur la même entrée sans utiliser de jetons sont appelées (k, t), où $k \in \mathbb{N}$ est utilisé à la place de $(\emptyset, t)$. Ils ont précisé les quatre fonctions. $\beta(x, k, s) = s$ Connecte les jetons à leurs emplacements, tandis que $\eta(x, t)$ connecte les déclenchements de transition à la transition qui se déclenche. La fonction β s'étend à une fonction des ensembles de jetons X aux multiensembles d'emplacements $\beta(X)$: $S \to \mathbb{N}$, par $\beta(X)(s) = |\{s' \in X \mid \beta(s') = s\}|$.[31]

Definition 2.4. Les ensembles de jetons S et de tirs de transition T d'un réseau de Petri $N\ (S, T, F, I, l)$ sont définis récursivement par
- $(*, k, s) \in S_\bullet$ pour $s \in S$ et $k < I(s)$;
- $(t', k, s)\ E\ S_\bullet$ pour $s \in S, t' \in T_\bullet\ et\ k < F(\eta(t'), s)$;
- $(X, t) \in T_\bullet$ pour $t \in T$ et $X \subseteq S$. tel que $\beta(X) = {}^\bullet t \neq 0$;
- $(k, t)\ T_\bullet$ pour $k \in \mathbb{N}$ et $t \in T$ tel que ${}^\bullet t = 0$.

La formule $l_\bullet(t) = l(\eta(t))$ donne la fonction d'étiquetage $l_\bullet : T_\bullet \to A$ sur les tirs de transition. Un multiensemble $M : S_\bullet \to \mathbb{N}$ de jetons constitue un marquage unique de N. $I_\bullet(*, k, s) = 1$ et $I_\bullet(t', k, s) = 0$ aboutit au marquage individuel initial $I_\bullet : S_\bullet \to \mathbb{N}$.
Filets réguliers.
Un réseau typique est celui qui a les propriétés suivantes : $\forall t \in T.\ {}^\bullet t > 0$. Chaque transition contient au moins un arc entrant. Si l'ensemble des transitions spontanées d'un réseau $To = \{(k, t) \in T\bullet \mid k \in \mathbb{N}$ est vide, le réseau est dit standard. Tout d'abord, on doit définis la règle de déclenchement pour les réseaux conventionnels qui incarne l'interprétation individuelle des jetons.

Definition 2.5. Laissez $U \subseteq T_\bullet$ représentent l'ensemble limité de tirs de transition dans un réseau typique.

$$^\bullet U = \sum_{(x,t)\in U} X\ et\ U^\bullet = \{(t', k, s) \mid t' \in U \wedge k < F(\eta(t'), s)\}$$

être à la fois l'ensemble des jetons de sortie de U et le multi-ensemble de jetons d'entrée. Si ${}^\bullet U \leq M$, l'ensemble U est activé sous une personne marquant $M \in \mathbb{N}^S$. Si c'est le cas, U peut tirer sous M, ce qui donne $M' = M - {}^\bullet U + {}^\bullet U \in \mathbb{N}^S$, parfois écrit $M \xrightarrow{U}_\bullet M'$.
Une séquence de tir est une chaîne $I_\bullet \xrightarrow{U_1}_\bullet M1 \xrightarrow{U_2}_\bullet \ldots \xrightarrow{U_n}_\bullet Mn$. S'il existe une telle séquence se terminant par $M = M_n$, alors il est possible de contacter la personne qui a marqué $M \in \mathbb{N}^{S\bullet}$.
L'affirmation qui suit affirme qu'ils ont réussi à nommer différemment chaque occurrence de jeton imaginable.

Proposition 2.1. Tout multi-ensemble accessible de jetons dans un réseau standard est un ensemble.

Preuve. Ils ont démontré que le multi-ensemble $I_\bullet - {}_\bullet M1 - {}_\bullet \ldots - {}_\bullet Mn$, qui contient Mn, est un ensemble dans la séquence de tir I Mn. En appliquant l'induction à n, le cas de base est vrai selon $I_\bullet$. Les définitions de $I_\bullet$ et $U^\bullet$ pour l'étape d'induction indiquent qu'un jeton a la forme (t', k, s) s'il apparaît deux fois dans $I_\bullet + \mathrm{Z}^{n}_{i=1} U_i$,et par conséquent, la transition tirant f' se produit deux fois dans $I_\bullet + \mathrm{Z}^{n}_{i=1} U^\bullet$ Puisque t' n'est pas spontané, il a la forme (X,t), où X est un ensemble de jetons non vide. Par le Un jeton dans X qui apparaît deux fois dans $I_\bullet + \mathrm{Z}^{n-1}_{i=1} U_i$; est défini comme ${}^\bullet U$.

La prop. 1 démontre que la mise à jour de la déf. 5 aux multi sets U ne sert à rien.

Filets non traditionnels. La définition de $U^\bullet$ pour les réseaux arbitraires reste la même, où U est une collection finie de transitions ; mais, dans la définition de ${}^\bullet U$, il faut choisir les circonstances d'entrée pour les tirs de transition spontanés. La méthode la plus simple serait d'inclure k dans la définition de ${}^\bullet U$, ou, alternativement, de considérer k comme $\emptyset$. Plusieurs copies de ses jetons de sortie dans le marquage accessible résultant. Ce problème peut être résolu en mettant à jour la définition d'une séquence de déclenchement pour inclure la restriction selon laquelle chaque déclenchement de transition spontanée ne peut avoir lieu qu'une fois à l'intérieur de celle-ci. Ici, avec les noms d'ensemble des futurs tirs de transition spontanés potentiels. L'ensemble des déclenchements de transition spontanés qui se sont déjà produits aurait été une source d'informations tout aussi riche, mais ils ont choisi l'option ci-dessus car elle combine les deux éléments d'un état en un seul ensemble de conditions préalables pour les déclenchements de transition.

Definition 2.6. N devrait être une boîte de Pétri. Supposons que $S^+_\bullet = S_\bullet \cup \{t_k \mid (k,t) \in T_0\}$ est l'ensemble des ressources de N. Un marquage individuel et une variété de noms de tirs de transition spontanés (k, t) se combinent pour former un seul état $M \in \mathbb{N}^{S^+_\bullet}$ de N. L'union de l'état initial I avec la liste des noms pour chaque tir de transition spontanée constitue l'état initial $I^+_\bullet = \{(*, k, s) \mid k < I(s)\} \cup \{t_k \mid (k,t) \in T_0$. La formule

$$U = \Sigma \quad X + \{t_k \mid (k,t) \in U \cap T_0\}$$

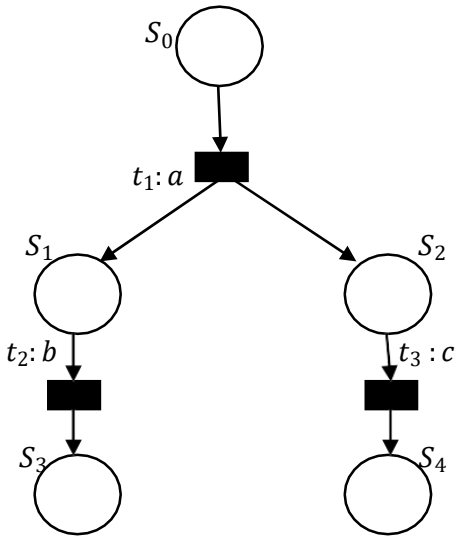
donne le multiensemble de ressources d'entrée pour un ensemble limité de tirs de transition $U \subseteq T_\bullet$. Tout en employant des états individuels plutôt que des marqueurs individuels et $I^+_\bullet$ plutôt que $I_\bullet$, les autres composants définitionnels de la clause cinq sont toujours valables.

II.5 Réseau de Petri réversible

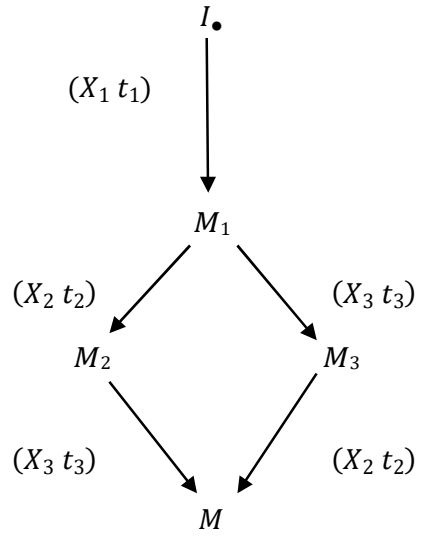
En raison de la réversibilité causale, un événement qui provoque d'autres événements ne peut être annulé qu'après que les événements qu'il a provoqués ont déjà été inversés. Ils y ont contribué en modélisant cette réversibilité au sein des IPN. Le tir vers l'avant de t peut être représenté comme la destruction de chaque jeton dans l'ensemble X et la fabrication subséquente de jetons dans l'ensemble $u^\bullet$. En supposant que N = (S , T , F , I , L) comme IPN, soit $u \in \mathrm{T}_\bullet$ tel que $u = (X, t)$. Par conséquent, l'annulation en arrière de u consomme $u^\bullet$ et produit X. Dans cette partie, ils montrent comment cette vision intuitive du défaire peut assurer la cohérence et la réversibilité flexible d'un système. [30]

Définition 2.7 : Pour un tir $u \in T_\bullet$ dans un réseau de Petri tel que $u = (X, t)$, soit ${}^\bullet u = X$ et $u^\bullet = \{(u, k, s) | K < F(\eta(u), s)\}$ soit l'ensemble des jetons d'entrée et l'ensemble des jetons de sortie jetons de u, respectivement.

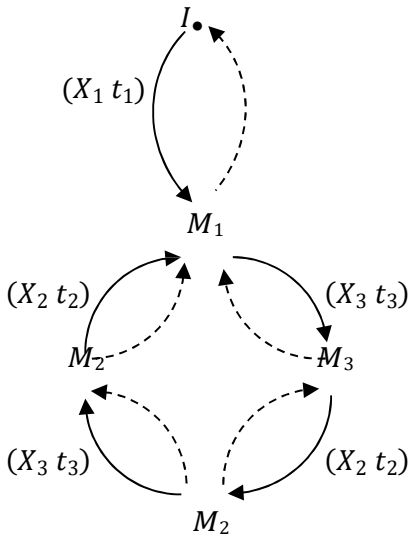
1. Tir vers l'avant : la transition t est activée sous un marquage individuel $M \in S_\bullet$ si ${}_\bullet u \subseteq M$. Dans ce cas, t peut tirer sous M, donnant $M' = (M \setminus {}^\bullet u) \cup u^\bullet$, écrit comme $M \xrightarrow{u}_{\bullet f} M'$ ou $M[u >_f M'$.
2. Tir vers l'arrière : la transition t peut être annulée sous un marquage individuel $M' \in S_\bullet$ si $u^\bullet \subseteq M'$. Dans ce cas, l'annulation de t peut se déclencher sous M', donnant $M = (M' \setminus u^\bullet) \cup^\bullet u$, noté $M' \xrightarrow{u}_{\bullet b} M$ ou $M'[u >_b M$.



(a) : réseau de Petri N



(b) : graphique de marquage habituel



(c) : graphe de marquage avec réversibilité causale

Figure 2.8: Réversibilité dans l'interprétation des jetons individuels réseau

Soit N l'IPN de la figure 2.4 à titre d'illustration, et soit $I_\bullet = \{(*,0, s_0)\}$ représente l'état de marquage initial de N. On ne peut exécuter $(X_1\ t_1)$ qu'avec la restriction que $X_1 = \{(*,0,\ s_0)\}$ en marquant $I_\bullet$. On a $I_\bullet$ [$(X_1\ t_1) >_{\mathbf{f}} M_1$ si M_1 tel que $M_1 = \{(\{(X_1\ t_1)\},0,\ s_1),(\{(X_1\ t_1)\},0,\ s_1)\}$.

Trois actions alternatives sont disponibles à l'état M_1 : exécuter $(X_2\ , t_2)$, exécuter (X_3, t_3) ou annuler (X_1, t_1) :

1. Si M_2 est tel que $X_2 = (X_1, t_1)\},0,\ s_1)\}$ et $M_2 = \{(\{(X_1, t_1)\},0,\ s_2),(\{(X_2, t_2)\},0,\ s_2)\}.$, alors M1$[(X_2, t_2) >_{\mathbf{f}} M_2$.
2. $M_1\ [(X_2, t_3) >_{\mathbf{f}} M_3$ tel que $M_3 = \{(\{(X_1\ t_1)\},0,\ s_2),(\{(X_3\ t_3)\},0,\ s_4)\}$.
3. $M_1\ [(X_1, t_1) >_{\mathrm{b}} I_\bullet$.

L'annulation de $(X_2\ , t_2)$, ou l'exécution de $(X_3\ , t_3)$, sera possible à partir de M_2 :

1. $M_2\ [(X_3\ , t_3) >_{\mathbf{f}} M_4$ tel que
$M_4 = \{(\{(X_2, t_2)\},0,\ s_3),(\{(X_3, t_2)\},0,\ s_4)\}$.
2. $M_2\ [(X_2, t_2) >_{\mathrm{b}} M_1$.

A partir de M_2 , nous avons :

1. $M_3\ [(X_2, t_2) >_{\mathbf{f}} M_4$ tel que
$M_4 = \{(\{(X_2,\ t_2)\},0,\ s_3),(\{(X_3, t_3)\},0,\ s_4)\}$.
2. $M_3\ [(X_3\ , t_3) >_{\mathrm{b}} M_1$

Le marquage M_4 permet de défaire $(X_3\ , t_3)$ ou $(X_2\ , t_2)$:

1. $M_3\ [(X_3, t_3) >_{\mathrm{b}} M_2$
2. $M_4\ [(X_2, t_2) >_{\mathrm{b}} M_3$

Ceux-ci seront tous récapitulés dans le graphique de marquage de la figure 4 (c), où un arc en pointillé indique l'annulation d'une action. Le graphe de marquage de N (sans réversibilité) est représenté sur la figure reffbIPN(b). Chacun peut voir que (i) les deux graphes précédents ont exactement le même jeu de marquage, et (ii) sur la Figure 4(c), l'annulation de la séquence $I_\bullet$ $(X_1\ , t_1) >_{\mathbf{f}} M_1\ [(X_2, t_2) >_{\mathbf{f}} M_2[(X_3, t_3) >_{\mathbf{f}} M_4$ est réalisé soit par retour arrière M_4 $[(X_3, t_3) >_{\mathrm{b}} M_2\ [(X_2, t_2) >_{\mathrm{b}} M_1[(X_1, t_1) >_{\mathrm{b}} I_\bullet$, soit par séquence $M_4\ [(X_2, t_2) >_{\mathrm{b}} M_3$ $[(X_3\ , t_3) >_{\mathrm{b}} M_1\ (X_1\ , t_1) >_{\mathrm{b}} I_\bullet$; c'est-à-dire par le retour arrière de sa séquence équivalente $I_\bullet\ (X_1\ , t_1) >_{\mathbf{f}} M_1\ [(X_3, t_3) >_{\mathbf{f}} M_2[(X_2, t_2) >_{\mathbf{f}} M_4$.

Il est évident de confirmer la cohérence et la réversibilité souple du système dans cette situation. On ne sait toujours pas si des IPN réversibles seront soumis à ces propriétés. La cohérence et la réversibilité flexible pour un IPN réversible spécifique sont établies dans les paragraphes qui suivent. De plus, la réversibilité à partir de n'importe quel état de marquage permet d'accéder au marquage initial.

Définition 2.**8**:Soit $u \in T_\bullet$. Relations $\mathfrak{F}, \mathfrak{B} \subseteq T_\bullet \times S_\bullet \times S_\bullet$ sont définis par

- $(u, M, M') \in \mathfrak{F}$ si et seulement si $M \xrightarrow[u]{}_{\bullet \mathbf{f}} M'$.
- $(u, M, M') \in \mathfrak{B}$ si et seulement si $M' \xrightarrow[u]{}_{\bullet \mathrm{b}} M$.

L'annulation d'un tir $(X\ ,\ t)$ dans un IPN réversible entraîne la génération de tout nouvel état de marquage. En termes simples, c'est l'opposé de l'état précédent de $(X,\ t)$. Il faut démontrer que $\mathfrak{B}$ est une fonction partielle inverse de $\mathfrak{F}$ pour le vérifier.

Proposition 2.2 : La fonction inverse de $\mathfrak{F}_{u}$ est $\mathfrak{B}_{u}$, disons donc $u \in T_{\bullet}$.

Proposition 2.3 : Dans un IPN réversible, l'annulation d'un tir (X, t) se traduit par un nouvel état de marquage. C'est simplement l'opposé du marquage précédent (X, t).

Définition 2.9. Ainsi, Soit $\sigma = u_1 . u_2 \ldots . u_{n}$ Un exemple dans $T_{\bullet}^{*}$ est le suivant : $M0 \xrightarrow{u1}_{f} M1 \xrightarrow{u2}_{f} M2 \xrightarrow{un}_{f}$ Mn . Fonction $\mathfrak{F} : T_{\bullet}^{*} \times S_{\bullet} \rightarrow S_{\bullet}\ est\ definit\ par\ F_{\sigma} = F_{un} \circ F_{un-1} \ldots . F_{u2} \circ F_{u1}$.

La partie suivante indique que $M[\sigma >_{f} M'\ pour\ \mathfrak{F}_{\sigma}(M) = M'$.

Poser $[M >_{f}$ est une liste complète de toutes les marques qui peuvent être utilisées en commençant par M pour toute marque M. Par conséquent, $[I_{\bullet} >_{f}$ est la collection de toutes les marques visibles de N.

Définition 2.10. Une autre progression dans $T_{\bullet}^{*}$ est la suivante : $M_{n} \xrightarrow{u_n}_{b} M_{n-1} \xrightarrow{u_{n-1}}_{b} M_{n-2} \longrightarrow_{b} M0$. Fonction $\mathfrak{B} : T_{\bullet}^{*} \times S_{\bullet} \rightarrow S_{\bullet}$ est défini comme suit :
$\mathfrak{B}$ est égal à $\mathfrak{B}_{\sigma} = \mathfrak{B}_{u_1} \circ \mathfrak{B}_{u_2} \ldots . \mathfrak{B}_{u_{n-1}} \circ \mathfrak{B}_{u_n}$.
Sur la note $M'[\sigma >_{b}$, versez $\mathfrak{B}_{\sigma}(M') = M$.

Ensemble $[M>_{b}$ est la somme de toutes les marques accessibles par réversibilité pour chaque marque M. de M, donc $[I_{\bullet} >_{b} = \{ M | \forall \sigma \in T_{\bullet}^{*}\ M[\sigma >_{b} I_{\bullet}\}$

Proposition 2. 4. Dans un IPN réversible, on a $\mathfrak{B}_{\sigma}(M') = M$ pour tout $\sigma \in T$ tel que $\mathfrak{F}_{\sigma}(M) = M'$.

Proposition 2.5. Dans un IPN réversible, $\mathfrak{B}_{\sigma}(M) = I_{\bullet}$ vaut pour tout $M \in S_{\bullet}$, tout $\sigma \in T_{\bullet}^{*}$.
Il est maintenant possible d'introduire la théorie de la cohérence des systèmes.

Theorem 2.1: Sachant que $N = (S, T, F, I, L)$ est un IPN et que les N états réversibles sont identiques, l'ensemble de tous les états possibles de N est : $[I_{\bullet} >= [I_{\bullet} >_{f} = [I_{\bullet} >_{b}$.

Théorème 2.2: La réversibilité est fluide
Si $\sigma \epsilon T_{\bullet}^{*}$ tel que $\sigma \sim \sigma'$ existe pour tout $\sigma \epsilon T_{\bullet}^{*}$ tel que $\mathfrak{F}_{\sigma}(M) = M'$, alors $\mathfrak{B}_{\sigma}(M') = \mathfrak{B}_{\sigma'}(M') = M$.
Selon cette théorie, la réversibilité d'une fonction peut être déterminée par son inverse ou par l'inverse de ses séquences correspondantes.

II.5.1. La réversibilité de réseau de Petri avec plusieurs jetons :

La réversibilité de réseau de Petri est un formalisme basé sur le réseau qui comprend des jetons individuels qui peuvent être liés entre eux par des liens. Il a été démontré que l'option de déclencher une transition plusieurs fois avec divers ensembles de jetons provoquait un non-déterminisme, communément appelé conflit en arrière, lors du déplacement en arrière. De plus, deux méthodes pour définir la sémantique réversible ont été trouvées face à ces conflits en

Arrière, qui étaient motivés par les interprétations individuelles et collectives des jetons développées pour expliquer la causalité dans les réseaux de Petri. Un type de réversibilité hors ordre causal a été produit dans l'étude du modèle RPN avec de nombreux jetons utilisant la technique du jeton collectif.

La réversibilité de réseau de Petri est une formalisation du réseau qui se compose de jetons individuels qui peuvent être connectés les uns aux autres par des liens. Il a été démontré que la possibilité de déclencher à plusieurs reprises une transition avec différents ensembles de boutons entraîne un non-déterminisme, également appelé conflit dans le dos, lors du mouvement dans le dos. Dans l'étude du modèle RPN à plusieurs jettons utilisant l'approche des jetons collectifs, une sorte de réversibilité causale dans le désordre s'est produite.
La réversibilité de réseau de Petri est une formalisation du réseau composé de jetons individuels qui peuvent être reliés entre eux par des liens. Il a été démontré que la possibilité d'initier une transition plusieurs fois avec diverses combinaisons de boutons entraîne un non-déterminisme, également appelé conflit dans la tête, lors du mouvement de la tête. Une forme de réversibilité causale dans le trouble se produit dans l'étude du modèle RPN à jets multiples utilisant l'approche des jetons collectifs.

En conséquence, après avoir interprété chaque jeton individuellement, faut-il inverser un calcul nécessite de tenir compte de l'activité précédente, dans ce cas, de différencier les jetons impliquant le stylo préexistant et les jetons utilisés pour déclencher chaque transition. Dans les sections qui suivent, ils ont mis en pratique cette stratégie d'ajout de nombreux jetons et examinons ses caractéristiques par rapport à la réversibilité de l'ordre causal. Ils ont construit également une connexion entre ce modèle et les RPN qui n'ont qu'un seul jeton. [34]

Le fait qu'il y ait plusieurs jetons de même nature dans ce nouvel environnement est un problème puisqu'il conduit aux phénomènes de non-déterminisme vers l'arrière lorsque les transitions sont inversées. Par exemple, comme le montre la figure 8(d), deux stylos construits seront présents à l'emplacement x ainsi qu'un composant i - c après l'exécution des transitions t_1 deux fois et t_2. Disons que la transition d'état t_1 est inversée dans ce cas. Toutes les occurrences de la liaison i - c sont considérées comme identiques dans l'interprétation du jeton collectif. Par conséquent, lors de l'inversion de la transition t_1, n'importe laquelle de ces liaisons peut être rompue. Les nombreux jetons d'encre et de gobelet sont identifiés sous l'interprétation individuelle des jetons, néanmoins, en fonction de leurs trajectoires causales. Étant donné que la composante assombrie des jetons de la figure est impliquée dans la première exécution de la transition t_1, qui aboutit au réseau de la figure 8(b), on pense que la transition t_2 a été initiée par la premières exécution de la transition t_1.Compte tenu de ces connexions causales entre les transitions, la composante i – c particulière ne doit pas être déconstruite tant que la transition t_2 n'est pas inversée, selon une sémantique de réversibilité causale. Semblable au stylo préexistant, aucune des transitions ne l'a généré, il ne doit donc pas être désassemblé en ses composants. Au lieu de cela, la liaison solitaire i - c dans le composant doit être rompue en inversant la transition t_1 dans le RPN de la figure 8(d). L'encre est enfermée à l'intérieur de la paire du gobelet et du bouton, ceci est donc compatible avec le concept selon lequel le

Démontage du produit ne permettrait pas la séparation de l'encre de l'intérieur du gobelet avantle retrait du bouton.

En suivant l'interprétation des jetons individuels, l'inversion d'un calcul nécessite de tenir compte de l'activité précédente, dans ce cas en différenciant les jetons impliquant le stylo préexistant et les jetons utilisés pour déclencher chaque transition. Dans les sections qui suivent, nous mettons en pratique cette stratégie d'ajout de nombreux jetons et examinons ses caractéristiques par rapport à la réversibilité de l'ordre causal. Nous construisons également une connexion entre ce modèle et les RPN qui n'ont qu'un seul jeton.

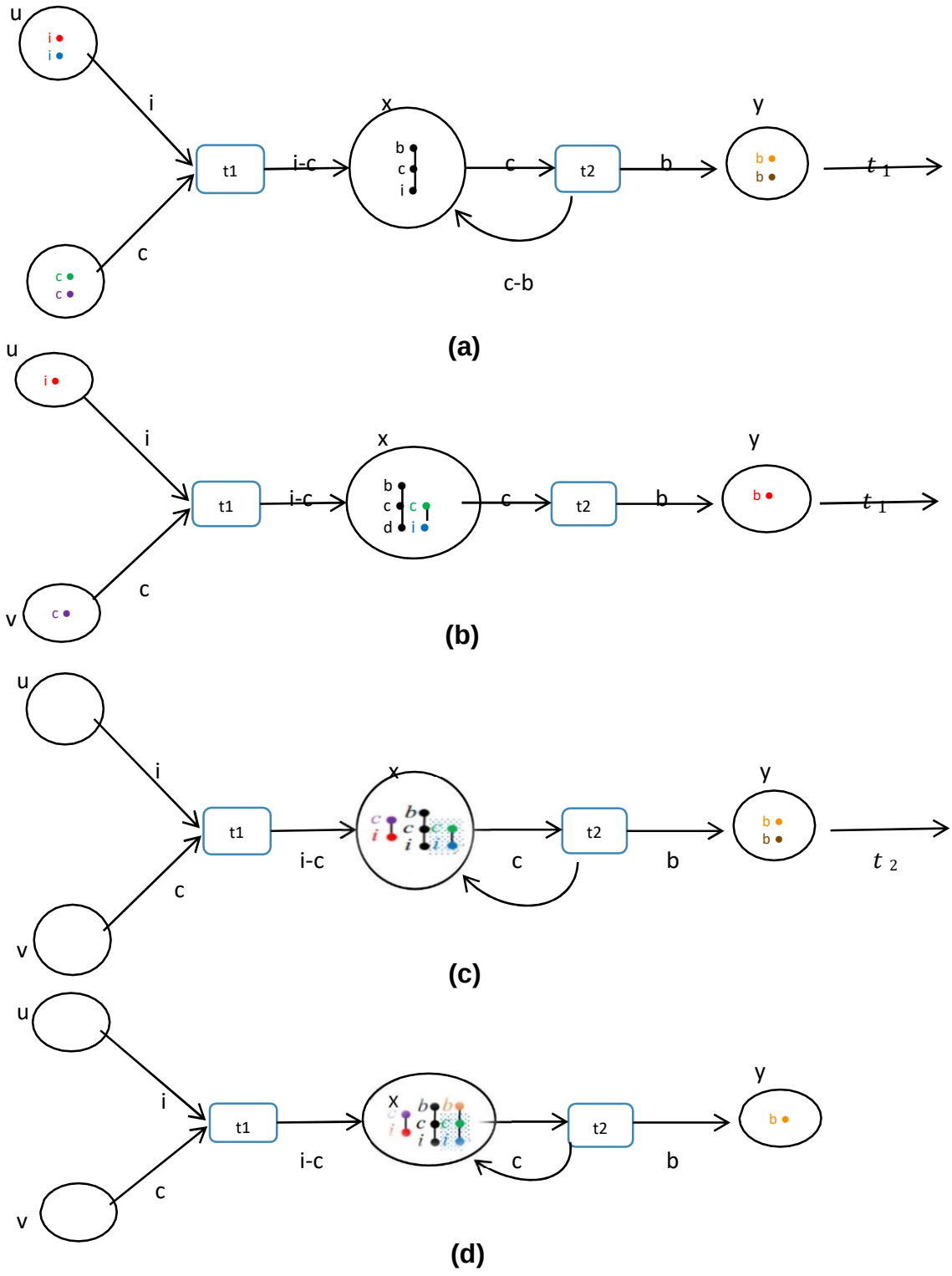


Figure 2.9: un système avec un stylo préassemblé

La Figure 2.9.(a) montre un système avec un stylo préassemblé en place x et deux pièces de chacun des composants d'encre, de gobelet et de bouton.

L'exécution de la transition t1 dans le réseau (a) peut produire le réseau en (b). Différentes sélections de jetons auraient pu être fait. Dans (b), la transition t1 est exécutée avec les (seuls) jetons disponibles menant à (c), où l'exécution de t2 avec le composant produit par la première exécution de t1 donne (d).

II.6 Conclusion

Dans ce chapitre, nous avons évoqué en détails le modèle réseau de Petri réversible. Nous avons présenté aussi la sémantique derrière ce modèle à savoir la sémantique de causalité. Il y a deux façon d'implémenter ce modèle, notre travail est basé sur le modèle qui utilise les jetons individuels de [33].

Chapitre 3

Conception et Implémentation

III.1 La conception

III.1.1 Positionnement

Dans cette première partie, nous aborderons, en premier lieu, l'interaction de notre module avec son environnement. Ce dernier c'est rien qu'un système prend en entrée un fichier "entrée.ipn" contient la description textuelle d'un réseau de Petri réversible, son objectif est de traiter ce fichier et de générer le graphe de marquage selon deux formats (i) textuelle via la sortie ".aut" (ii) graphique (image) via ".dot".

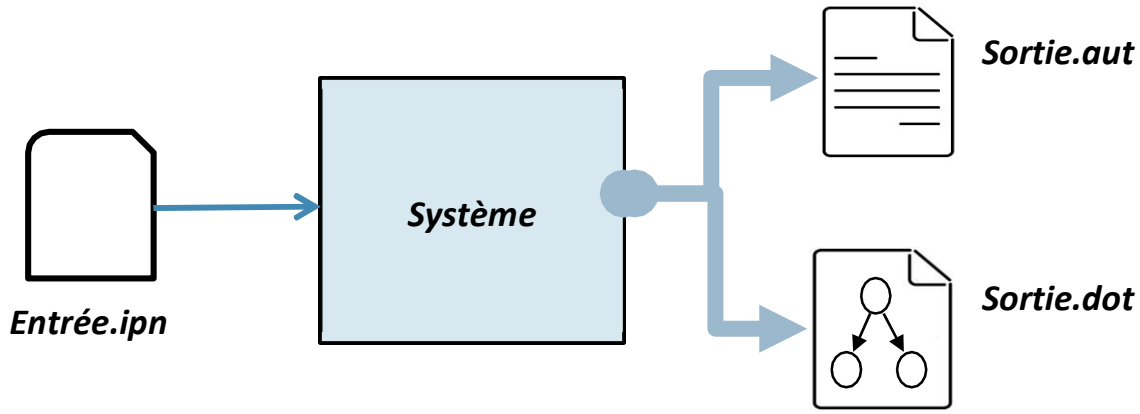


Figure 3.1: Environnement

Cet environnement contient différents modules interagissent entre eux tels que la pile d'analyseur, le générateur, et les transformateurs textuel et DOT.

Une fois le réseau de Petri réversible compilé avec succès, le système générera le graphe de marquage. Le fichier "sortie.aut" contiendra une représentation textuelle de graphe de marquage généré, tandis que le fichier "sortie.dot" fournira sa représentation graphique.

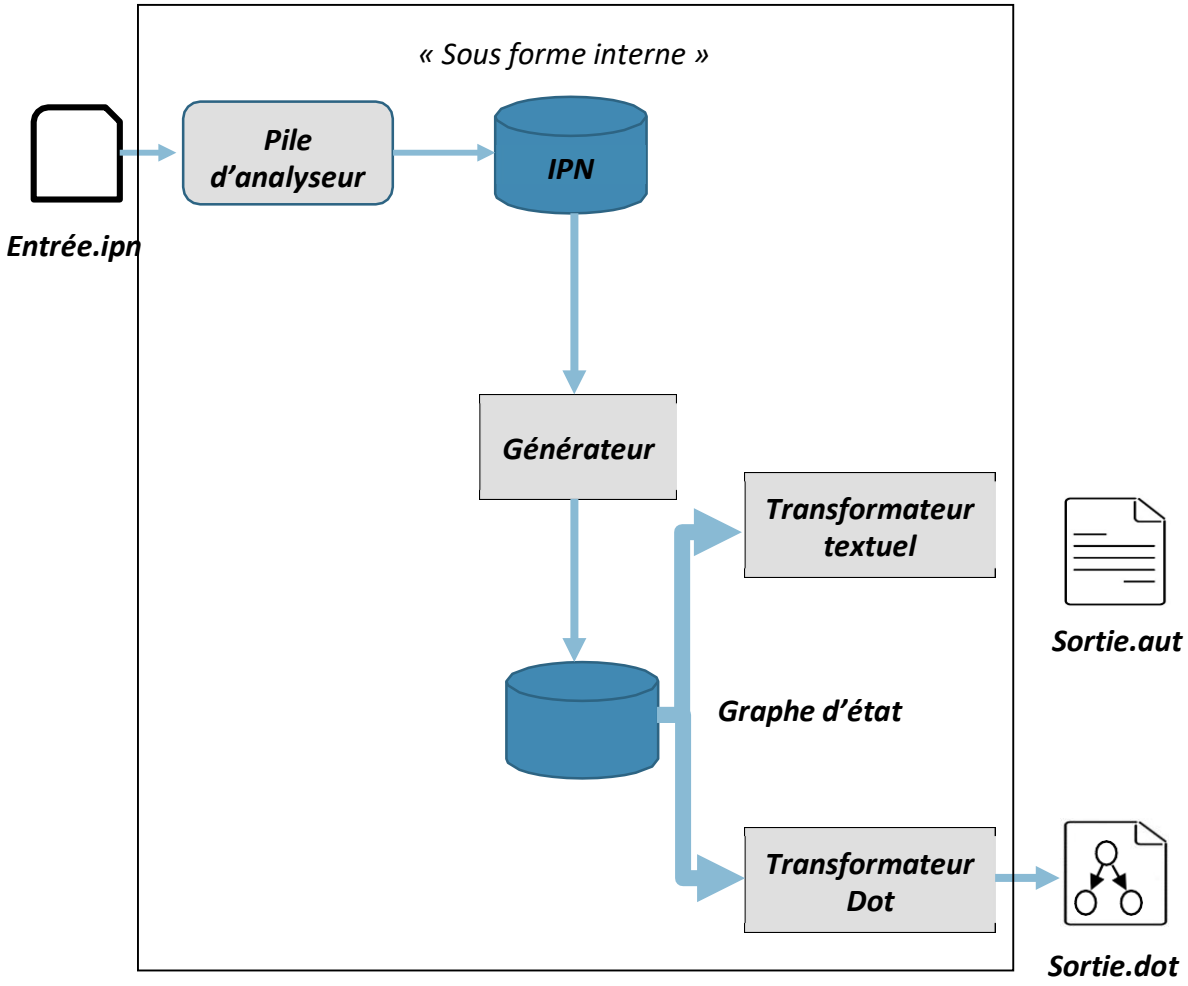


Figure 3.2: Architecture de l'environnement

Ce présent projet va implémenter seulement le module compilateur (Pile d'analyseur). Dans ce qui suit, on va proposer la grammaire d'un réseau de Petri réversible et l'implémentation du compilateur. Nous avons proposé la grammaire derrière le fichiers d'extension « .aut » .

III.2 Pile d'analyse

La pile d'analyse est un composant essentiel du système. Elle reçoit le fichier d'entrée "entrée.ipn" et effectue une analyse approfondie en utilisant les analyseurs lexicaux, syntaxique et sémantique. Elle produit un réseau de Petri sous forme une structure interne dynamique qui sera utilisé par les autres composants de l'environnement.

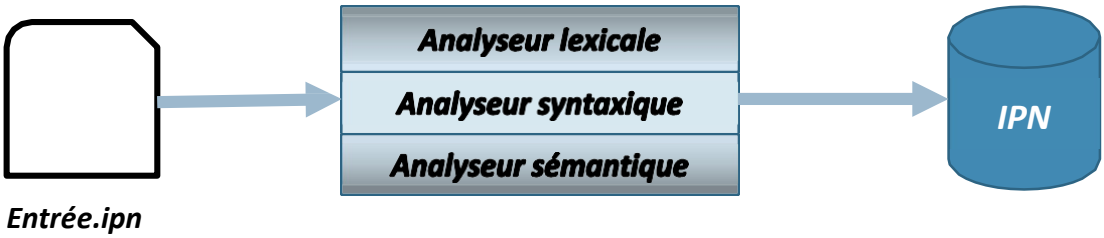


Figure 3.3: Architecture de la pile d'analyseur

L'analyse lexicale segmente le fichier d'entrée en lexèmes, qui sont les unités lexicales significatives du langage. Ensuite, l'analyse syntaxique sera effectuée pour vérifier la conformité du fichier d'entrée à la grammaire définie pour les réseaux de Petri réversibles. Enfin, l'analyse sémantique permettra de garantir la cohérence sémantique du réseau de Petri réversible, en vérifiant les règles et les contraintes spécifiées. Dans ce projet, On va ignoré l'analyseur sémantique car le fichier est produit par l'outil graphique (Figure 0.1).

IPN ->NB_Trans,NB_Place,Place,Token,Transition
NB_Trans ->'NB_Trans' ':' num ';'
NB_Place ->'NB_Place' ':' num ';'
Place ->'Place' ':' Identifier ';'
Tokens ->'Token' ':' '(' ListToken ')' ';'
ListToken ->'' ',' num ',' identifier*
Transition -> TRANSITIONS ,:''(, ' TransitionList')''}'
TransitionList ->'{,'(, LISTEpre ,)'',' identifier','(, LISTEpost')' ,}'
LISTEpre ->'(' identifier',' num ')'
LISTEpre ->'('num',' identifier ')'
Digit -> '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
*num -> Digit**
*Identifier -> Letter (Letter | Digit)**
Letter -> 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z' | 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z'

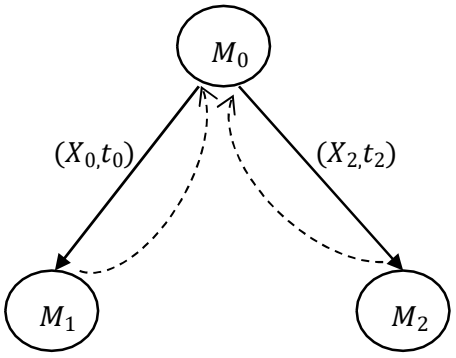
Figure 3.4:grammaire de réseau de Petri reversible

Transformateur textuel :

Le transformateur textuel prend en entrée le graphe d'état généré par le générateur et le transforme en un fichier de sortie au format textuelle "sortie.aut". Ce fichier contient la représentation textuelle du graphe de marquage. Le transformateur textuel utilise des règles de formatage et des langages de description pour générer le fichier de sortie de manière cohérente et lisible.

```
Transformateur ->NB_Trans,NB_Stats,System
System          ->'System' '{' Transition List '}' 'End System'
NB_Trans        ->'NB_Trans' Digit ';'
NB_Stats        ->'NB_Stats' Digit ';'
TransitionList   -> Transition TransitionList | Transition
Transition       -> '(' State ',' Token ',' State ',' Direction ')' ';'
State            -> 'S' Digit
Token            -> '(' Cause ',' Fire ')'
Cause             -> num
Fire             ->Identifier
Direction         -> 'F' | 'B'
Digit             -> '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
Num              ->Digit*
Identifier        -> Letter (Letter | Digit)*
Letter           -> 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' |
'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z' | 'a' | 'b' | 'c' | 'd' |
'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' |
'x' | 'y' | 'z'
```

Figure 3.5::Grammaire utilisé par le transformateur textuel




```
NB_Trans= 3;
NB_State= 2;

System {
        (S0, (2, t2), S2, F);
        (S0, (0, t0), S1, F);
        (S2, (2, t2), S0, F);


        } End System
```

Figure 3.6: Exemple d'un graphe de marquage vers une sortie textuelle avec la grammaire fournie

III.3 Implémentation

1. **Analyse lexicale** : L'analyse lexicale est la première étape du processus de compilation. Elle consiste à analyser le code source caractère par caractère et à le diviser en unités lexicales appelées lexèmes. Les lexèmes sont des symboles significatifs dans le langage de programmation, tels que les mots-clés, les identificateurs, les nombres, les opérateurs, les symboles de ponctuation dans cette phase nous avons utilisé l'outils lex.
2. **Analyse syntaxique** : L'analyse syntaxique est l'étape suivante après l'analyse lexicale. Elle consiste à vérifier la structure grammaticale du code source conformément à la grammaire du langage de programmation. L'analyse syntaxique est essentielle pour vérifier que le code source est bien formé et respecte les règles de la grammaire du langage. Elle permet également de détecter les erreurs de syntaxe et de produire des messages d'erreur informatifs en cas de non-conformité.
3. **Analyse sémantique** : L'analyse sémantique joue un rôle crucial dans la garantie de la cohérence et de la validité du programme sur le plan sémantique, en détectant les erreurs sémantiques telles que les incohérences de types, les variables non déclarées, les conflits de portée, etc. Elle facilite également la génération de code intermédiaire ou la construction de structures de données nécessaires pour les étapes ultérieures de la compilation. L'analyse syntaxique est essentielle pour vérifier que le code source est bien formé et respecte les règles de la grammaire du langage. Elle permet également de détecter les erreurs de syntaxe et de produire des messages d'erreur informatifs en cas de non-conformité

En combinant l'analyse lexicale, syntaxique et sémantique, nous pouvons construire un système de compilation solide pour le réseau de Petri réversibles en utilisant les outils lex et yacc.

III.3.1 Lex et yacc

Ces deux outils ont été initialement développés dans le cadre du projet UNIX. Lex a été créé par Mike Lesk et Eric Schmidt en 1975, tandis que Yacc a été créé par Stephen C. Johnson en 1970. Ils sont devenus très populaires pour la construction de compilateurs et ont été implémentés dans de nombreux systèmes d'exploitation et environnements de développement.

Dans notre travail nous avons utilisé fla version flex et Bison.

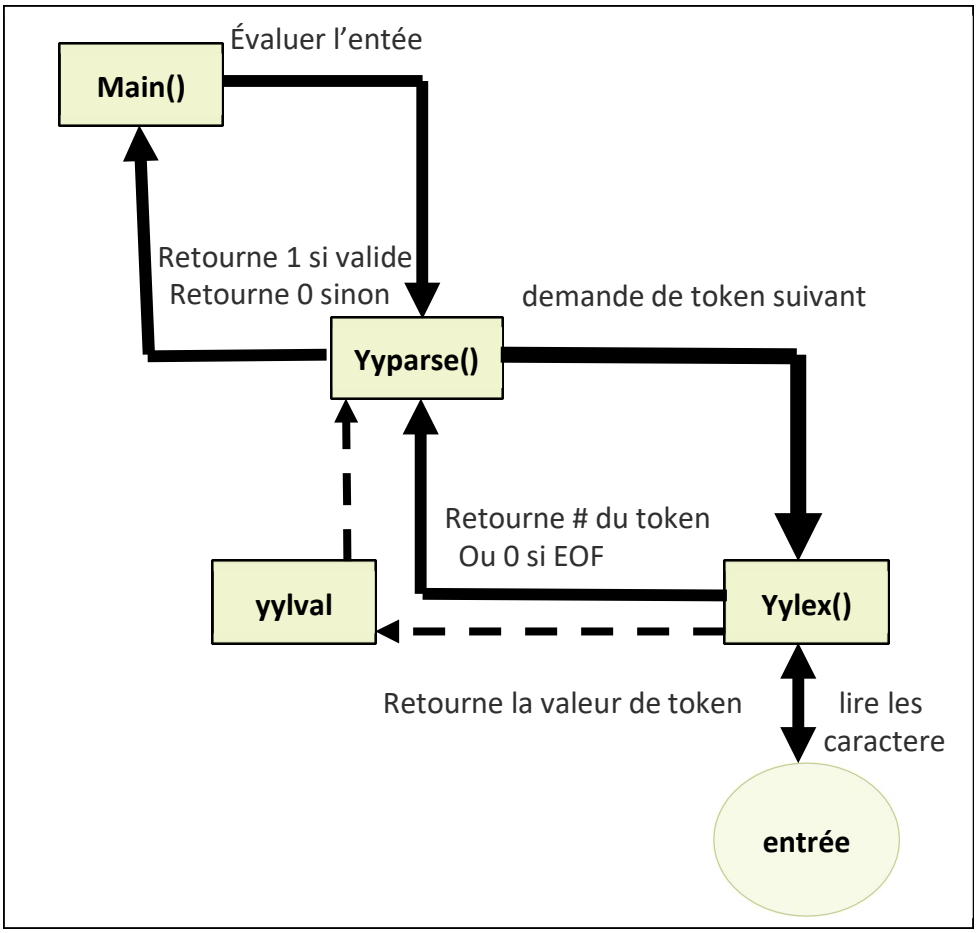


Figure 3.7: La démarche à suivre pour utiliser Lex/yacc

III.3.1.1 Lex

L'outil Lex est couramment utilisé pour effectuer l'analyse lexicale. Il permet de spécifier des expressions régulières qui définissent les motifs de correspondance des différents lexèmes du langage. En utilisant les règles définies dans le fichier Lex, l'outil génère un analyseur lexical qui parcourt le code source et reconnaît les lexèmes en fonction des expressions régulières spécifiées.

L'analyse lexicale est importante car elle prépare le terrain pour les étapes suivantes de la compilation en fournissant une séquence de lexèmes qui seront utilisés par l'analyse syntaxique.

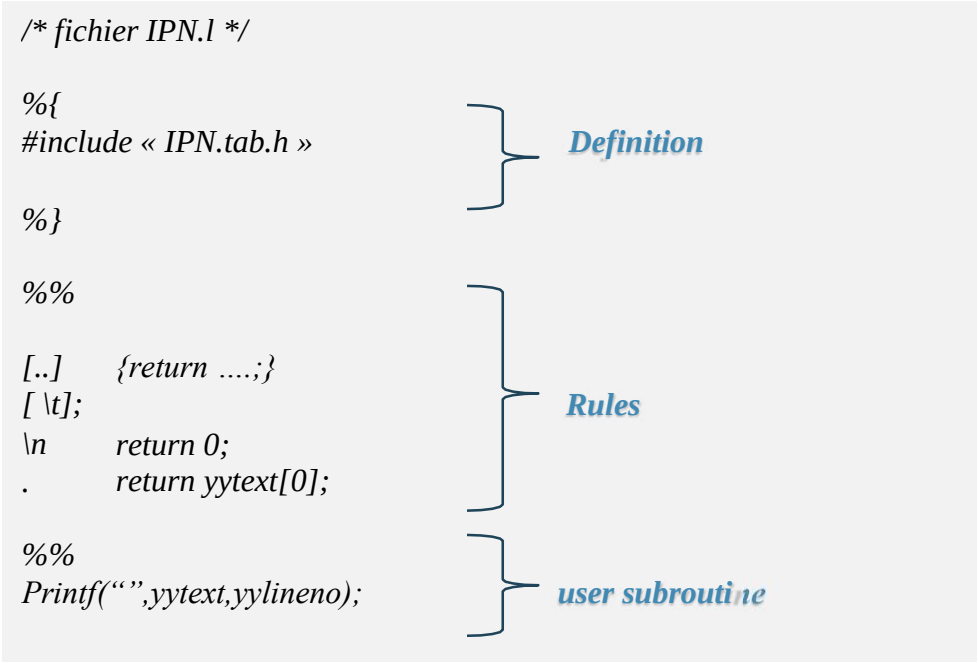


Figure 3.8: Le squelette d'un fichier Lex

III.3.1.2 Yacc

L'outil Yacc (ou Bison) est couramment utilisé pour effectuer l'analyse syntaxique. Il utilise des règles de production qui décrivent les structures syntaxiques valides du langage. Les règles de production sont écrites dans un fichier Yacc (.y) et spécifient comment les lexèmes reconnus par l'analyse lexicale peuvent être combinés pour former des constructions syntaxiquement valides.

Yacc génère un analyseur syntaxique qui parcourt les lexèmes et construit un arbre syntaxique ou une structure de données équivalente qui représente la structure hiérarchique du code source. L'arbre syntaxique permet de capturer la hiérarchie des expressions, des instructions et des blocs de code du programme.

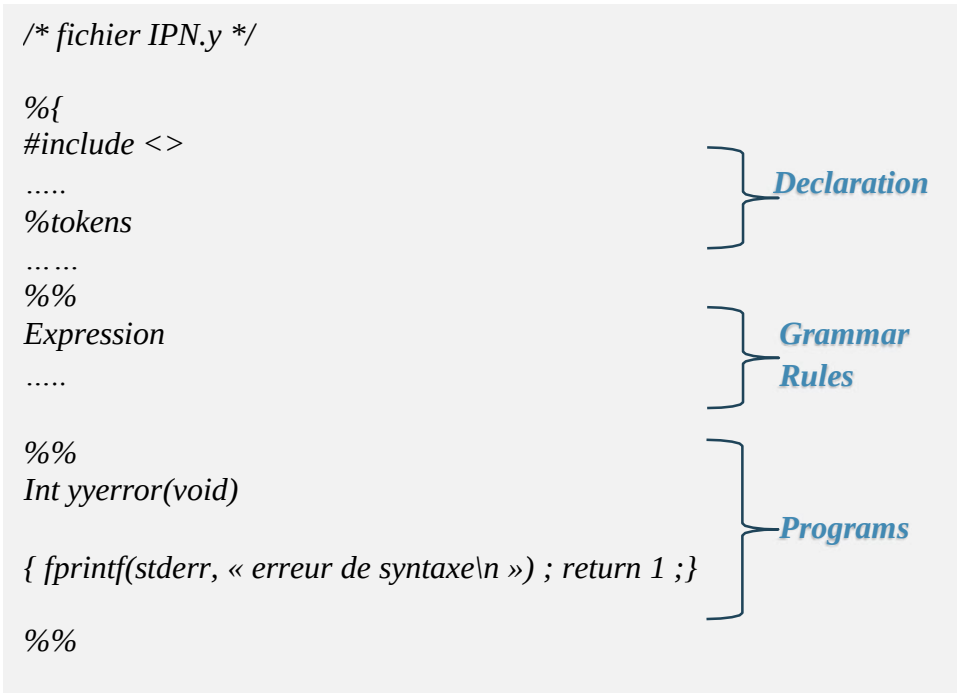


Figure 3.9: Le squelette d'un fichier Yacc

III.3.2 L'implémentation de la pile d'analyseur :

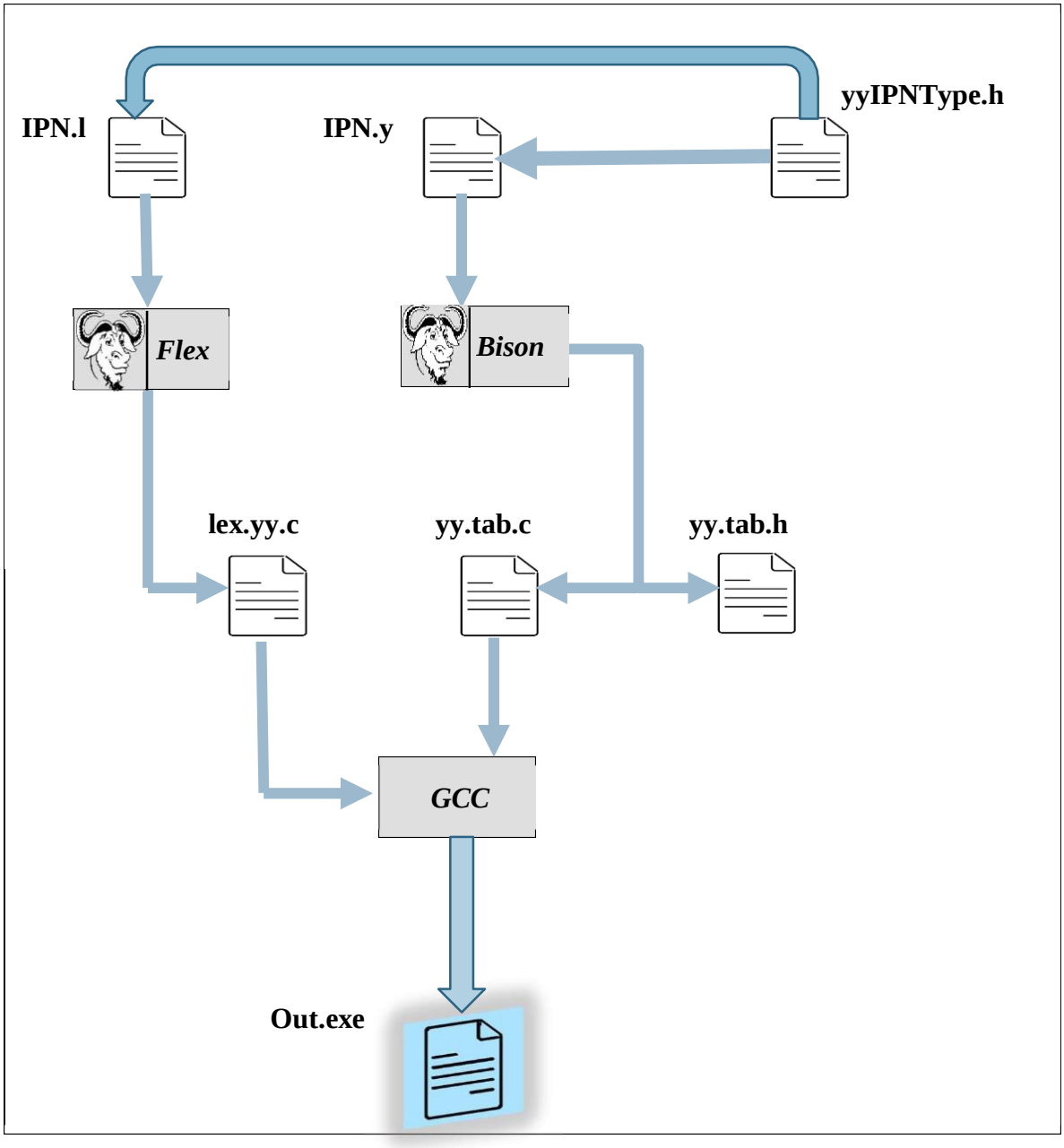


Figure 3.10: Les étapes à suivre à fin d'avoir notre exécutable

Compilation :

1. Installer les outils Lex et Yacc (flex\ bison).
2. Concevoir les fichier IPN.l et IPN.y et yyIPNType.h .
3. Générez le fichier C correspondant au fichier Lex en exécutant la commande dans le terminal, Cela générera un fichier lex.yy.c.
4. Ajouter l'en-têtes nécessaires IPN.tab.h dans le fichiers IPN.l qui est généré par yacc.

5. Générez le fichier C correspondant à le fichier Yacc en exécutant la commande dans le terminal, Cela générera les fichiers IPN.tab.c et IPN.tab.h.
6. Compilez les fichiers C générés (lex.yy.c, IPN.tab.c) avec les commandes de compilation appropriées. Cela produira les fichiers objets lex.yy.o et IPN.tab.o.
7. Liez les fichiers objets ensemble pour créer l'exécutable final.
8. Exécutez l'exemple avec le compilateur en utilisant la commande appropriée.

```
lex IPN.l
gcc -c IPN.tab.c

gcc lex.yy.o IPN.tab.o -o out

./ out <  test.txt
```

```
%{
#include "IPN.tab.h"
%}
%option noyywrap
%option yylineno

idf [a-zA-Z0-9_]+

%%

"NB_places"              return NBPLACES;
"NB_transitions"         return NBTRANSITIONS;
"Tokens"                 return TOKENS;
"Places"                 return PLACES;
"Transitions"            return TRANSITIONS;
"{"                      return AO;
","                      return VIR;
"}"                      return AF;
":"                      return TP;
";"                      return PV;
[0-9]+                   { yylval.num = atoi(yytext); return
DIGITAL; }
{idf}                    { yylval.id = strdup(yytext); return
ID; }
"("                      return PO;
")"                      return PF;
"*"                      return ETOILE;
[ \t\n]+                 /* ignorer les espaces */
.

%%
//. printf("errer lex (%s)) a ligne %d\n", yytext, yylineno);
```

Figure 3.11: Le fichier IPN.l

```
%{
#include <stdlib.h>
#include <stdio.h>
void yyerror(char *s);
%start PROGRAM

%token AO VIR AF TP PV  PO PF ETOILE
%token NBPLACES
%token NBTRANSITIONS
%token TOKENS
%token PLACES
%token TRANSITIONS
%%
PROGRAM : NBplace NBTransitions TOCKENs Places Transitions ;
NBplace : DIGITAL
NBTransitions : NBTRANSITIONS TP DIGITAL
TOCKENs :TOKENS TP AO LSTtocken AF PV ;
Places :PLACES TP AO Pliste AF PV ;
Pliste : ID
      | Pliste VIR ID
      ;
Transitions : TRANSITIONS TP AO ListEtransition AF PV
            ……

%%

void yyerror(char *s) {
fprintf(stdout, "%s\n", s);
}
```

Figure 3.12: Une partie du fichier IPN.y

Conclusion Générale

Le calcul réversible est un paradigme de calcul où chaque opération est réversible, c'est-à-dire qu'il est possible de retrouver l'état précédent à partir de l'état actuel. Dans un système de calcul réversible, toutes les informations nécessaires pour inverser une opération sont conservées, permettant ainsi de remonter le temps et d'annuler les calculs effectués. Cela diffère du calcul classique, où certaines informations peuvent être perdues lors d'une opération et où il n'est pas possible de retrouver l'état précédent de manière unique.

Les réseaux de Petri réversibles sont une extension des réseaux de Petri, qui sont des outils de modélisation graphique utilisés pour représenter les systèmes concurrents et distribués. Dans un réseau de Petri réversible, les transitions sont réversibles, ce qui signifie qu'il est possible d'inverser le marquage du réseau et de revenir à un état antérieur. Cela permet de modéliser des systèmes où les actions peuvent être annulées, les ressources récupérées et les états précédents restaurés.

Les réseaux de Petri réversibles et le calcul réversible sont tous deux utilisés dans des domaines tels que la conception de circuits logiques réversibles, la simulation de systèmes physiques réversibles, la conception de systèmes informatiques à faible consommation d'énergie et la rétro-ingénierie. Ces concepts sont souvent utilisés lorsque la réversibilité et la récupération d'information sont importantes, par exemple dans les systèmes de calcul quantique, où la réversibilité est une propriété fondamentale des opérations quantiques.

Développer un environnement formel adoptant le calcul réversible via réseaux de Petri est quasiment indispensable pour aller plus loin dans cet axe de recherche. Dans ce présent projet nous avons conçu sa première pierre qui consiste à développer un module (compilateur) qui prend en entrée une description textuelle d'un Réseau de Petri Réversible (RPR) et la transformes en structure de données spécifique pour être utilisée ensuite par d'autre module de l'environnement. Précisément, nous avons (i) proposé une grammaire adéquate aux réseaux de Petri réversibles (ii) développé l'analyseur lexical (iii) développé l'analyseur syntaxique.

L'implémentation de cet environnement vient tout juste d'être lancée, il y un ensemble de modules à développer à savoir le générateur de graphe de marquage, l'éditeur graphique et les outils de la vérification.

Bibliographie

[1] R. Landauer, « Irreversibility and Heat Generation in the Computing Process », *IBM J. Res. Dev.*, vol. 5, nº 3, p. 183-191, juill. 1961, doi: 10.1147/rd.53.0183.

[2] L. Burgholzer et R. Wille, « Improved DD-based Equivalence Checking of Quantum Circuits », in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, janv. 2020, p. 127-132. doi: 10.1109/ASP-DAC47756.2020.9045153.

[3] D. Della Giustina, C. Piazza, B. Riccardi, et R. Romanello, « Directed Graph Encoding in Quantum Computing Supporting Edge-Failures », in *Reversible Computation*, C. A. Mezzina et K. Podlaski, Éd., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, p. 75-92. doi: 10.1007/978-3-031-09005-9_6.

[4] P. Niemann, L. Müller, et R. Drechsler, « Finding Optimal Implementations of Non-native CNOT Gates Using SAT », in *Reversible Computation*, S. Yamashita et T. Yokoyama, Éd., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, p. 242-255. doi: 10.1007/978-3-030-79837-6_15.

[5] B. Aman et G. Ciobanu, « Bonding calculus », *Nat. Comput.*, vol. 17, nº 4, p. 823-832, déc. 2018, doi: 10.1007/s11047-018-9709-7.

[6] S. Kuhn, « Simulation of Base Excision Repair in the Calculus of Covalent Bonding », in *Reversible Computation*, J. Kari et I. Ulidowski, Éd., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, p. 123-129. doi: 10.1007/978-3-319-99498-7_8.

[7] V. Danos et J. Krivine, « Reversible Communicating Systems », in *CONCUR 2004 - Concurrency Theory*, P. Gardner et N. Yoshida, Éd., in Lecture Notes in Computer Science, vol. 3170. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, p. 292-307. doi: 10.1007/978-3-540-28644-8_19.

[8] V. Danos et J. Krivine, « Transactions in RCCS », in *CONCUR 2005 – Concurrency Theory*, M. Abadi et L. de Alfaro, Éd., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, p. 398-412. doi: 10.1007/11539452_31.

[9] T.-S. Yu et M. Pinedo, « Flow shops with reentry: Reversibility properties and makespan optimal schedules », *Eur. J. Oper. Res.*, vol. 282, nº 2, p. 478-490, avr. 2020, doi: 10.1016/j.ejor.2019.09.036.

[10] A. J. Benavides et A. Vera, « The reversibility property in a job-insertion tiebreaker for the permutational flow shop scheduling problem », *Eur. J. Oper. Res.*, vol. 297, nº 2, p. 407-421, mars 2022, doi: 10.1016/j.ejor.2021.05.014.

[11] C. Linshu, W. Jiayang, L. I. U. Yuanhui, et M. A. Qing, « Study on Reversibility of Quotient Space Granularity », *J. Front. Comput. Sci. Technol.*, vol. 13, nº 5, p. 884, mai 2019, doi: 10.3778/j.issn.1673-9418.1804039.

[12] D. Bhattacharjee, S. S. Roy, et R. Sadhu, « ENTROPY REVERSIBILITY SCENARIO – OVER PROJECTIONS AND SIMULATIONS », *EPRA Int. J. Res. Dev. IJRD*, vol. 7, nº 7, Art. nº 7, juill. 2022.

[13] A. Bhattacharjee, C. Bandyopadhyay, et H. Rahaman, « A Novel Heuristic Method for Linear Nearest Neighbour Realization of Reversible Circuits », *IETE J. Res.*, vol. 0, nº 0, p. 1-19, févr. 2022, doi: 10.1080/03772063.2022.2027822.

[14] G. Zhang, « Lower-energy conformers search of TPP-1 polypeptide via hybrid particle swarm optimization and genetic algorithm », 2018, Consulté le: 20 juin 2023. [En ligne]. Disponible sur: https://shareok.org/handle/11244/316277

[15] K. Liu *et al.*, « Shift-Collapse Acceleration of Generalized Polarizable Reactive Molecular Dynamics for Machine Learning-Assisted Computational Synthesis of Layered Materials », in *2018 IEEE/ACM 9th Workshop on Latest Advances in Scalable*

Algorithms for Large-Scale Systems (scalA), nov. 2018, p. 41-48. doi: 10.1109/ScalA.2018.00009.
[16] A. Benamira, « CAUSAL REVERSIBILITY IN INDIVIDUAL TOKEN INTERPRETATION OF PETRI NETS », *Comput. Sci.*, vol. 21, nov. 2020, doi: 10.7494/csci.2020.21.4.3728.
[17] A. Philippou et K. Psara, « Token Multiplicity in Reversing Petri Nets Under the Individual Token Interpretation », *Electron. Proc. Theor. Comput. Sci.*, vol. 368, p. 131-150, août 2022, doi: 10.4204/EPTCS.368.8.
[18] B. Aman *et al.*, « Foundations of Reversible Computation », in *Reversible Computation: Extending Horizons of Computing*, Springer, Cham, 2020, p. 1-40. doi: 10.1007/978-3-030-47361-7_1.
[19] J. Esparza et M. Nielsen, « Decidability Issues for Petri Nets », *BRICS Rep. Ser.*, vol. 1, nº 8, mai 1994, doi: 10.7146/brics.v1i8.21662.
[20] A. Finkel et Z. Bouziane, « Cyclic Petri Net Reachability Sets are Semi-linear Effectively Constructible », *Electr Notes Theor Comput Sci*, vol. 9, p. 15-24, déc. 1997, doi: 10.1016/S1571-0661(05)80423-2.
[21] T. Murata, « Petri nets: Properties, analysis and applications », *Proc. IEEE*, vol. 77, nº 4, p. 541-580, avr. 1989, doi: 10.1109/5.24143.
[22] M. Colange, S. Baarir, F. Kordon, et Y. Thierry-Mieg, « Crocodile: A Symbolic/Symbolic Tool for the Analysis of Symmetric Nets with Bag », juin 2011, p. 338-347. doi: 10.1007/978-3-642-21834-7_20.
[23] K. Barylska, E. Erofeev, M. Koutny, Ł. Mikulski, et M. Piątkowski, « Reversing Transitions in Bounded Petri Nets », *Fundam. Informaticae*, vol. 157, nº 4, p. 341-357, janv. 2018, doi: 10.3233/FI-2018-1631.
[24] E. Badouel, L. Bernardinello, et P. Darondeau, *Petri Net Synthesis*. in Texts in Theoretical Computer Science. An EATCS Series. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. doi: 10.1007/978-3-662-47967-4.
[25] D. Frutos Escrig, M. Koutny, et Ł. Mikulski, « An Efficient Characterization of Petri Net Solvable Binary Words », 2018, p. 207-226. doi: 10.1007/978-3-319-91268-4_11.
[26] H. Melgratti, C. Mezzina, et I. Ulidowski, « Reversing P/T Nets », 2019, p. 19-36. doi: 10.1007/978-3-030-22397-7_2.
[27] A. Philippou et K. Psara, « Reversible Computation in Petri Nets: 10th International Conference, RC 2018, Leicester, UK, September 12-14, 2018, Proceedings », 2018, p. 84-101. doi: 10.1007/978-3-319-99498-7_6.
[28] K. Barylska, M. Koutny, Ł. Mikulski, et M. Piątkowski, « Reversible Computation vs. Reversibility in Petri Nets », juill. 2016, p. 105-118. doi: 10.1007/978-3-319-40578-0_7.
[29] A. Philippou, K. Psara, et H. Šiljak, « Controlling Reversibility in Reversing Petri Nets with Application to Wireless Communications: Work-in-Progress Paper », 2019, p. 238-245. doi: 10.1007/978-3-030-21500-2_15.
[30] A. Benamira, « CAUSAL REVERSIBILITY IN INDIVIDUAL TOKEN INTERPRETATION OF PETRI NETS », *Comput. Sci.*, vol. 21, nov. 2020, doi: 10.7494/csci.2020.21.4.3728.
[31] R. Glabbeek, « The Individual and Collective Token Interpretations of Petri Nets », présenté à Lecture Notes in Computer Science, août 2005, p. 323-337. doi: 10.1007/11539452_26.
[32] L. Popova-Zeugmann, *Time and Petri Nets*. 2013. doi: 10.1007/978-3-642-41115-1.
[33] R. Glabbeek, « The Individual and Collective Token Interpretations of Petri Nets », présenté à Lecture Notes in Computer Science, août 2005, p. 323-337. doi: 10.1007/11539452_26.

[34] A. Philippou et K. Psara, « Token Multiplicity in Reversing Petri Nets Under the Individual Token Interpretation », *Electron. Proc. Theor. Comput. Sci.*, vol. 368, p. 131-150, août 2022, doi: 10.4204/EPTCS.368.8.