

Chapitre 6

Réseaux de Petri et logique linéaire

6.1. Introduction

Les méthodes d'analyse des réseaux de Petri sont très souvent des méthodes « orientées états », c'est-à-dire qu'elles utilisent le graphe d'états, ou par exemple le graphe des classes d'états dans le cas des réseaux de Petri temporels (chapitre 5 en [DIA 01]), comme support de l'analyse. Le chapitre 1 a aussi montré comment exprimer et vérifier des propriétés en utilisant la logique temporelle. L'une des difficultés de ces approches étant l'explosion combinatoire du graphe d'états, des méthodes ont aussi été développées dans cet ouvrage pour tenter de le réduire (graphe des pas couvrants, symétries, voir chapitres 2 et 4).

L'approche développée dans ce chapitre est relativement différente dans le sens où elle propose une méthode d'analyse « orientée événements », les événements d'un réseau de Petri étant, bien entendu, les franchissements des transitions. En effet, par leur définition, les réseaux de Petri offrent une quasi totale symétrie entre états (places) et événements (transitions). Cette symétrie se retrouve dans les méthodes d'analyse structurelle développées au chapitre 3 de [DIA 01] puisqu'il est tout aussi aisé de calculer des invariants de transitions que des invariants de places. Par contre les différents outils d'analyse d'accessibilité se basent en général sur les états : nous allons voir comment il est possible d'utiliser une approche orientée événements pour examiner certaines questions d'accessibilité. Pour cela nous utiliserons un « raisonnement logique » qui s'appuie directement sur la structure du réseau de Petri, sans générer le graphe d'états. De plus, nous verrons comment cette analyse orientée actions (franchissements de transitions) nous permet également d'obtenir des informations temporelles

précises concernant les séquences de transitions ou plutôt les ensembles de transitions franchies : ainsi nous pourrions calculer leur durée.

Pourquoi ce choix de travailler en n'utilisant pas le graphe d'états ? La raison première est certainement le traitement du parallélisme. L'une des caractéristiques des graphes d'états est de représenter le parallélisme par l'entrelacement, c'est-à-dire par un choix. Mais le graphe d'états contient déjà d'autres types de choix, ceux qui sont issus de la résolution des conflits par exemple : il existe donc un risque de confusion. Une autre façon d'envisager l'entrelacement est de s'intéresser aux liens de causalité. En considérant que deux événements a et b peuvent avoir lieu en parallèle, on spécifie qu'il n'y a, *a priori*, aucun lien de causalité entre eux : ce sont des événements non ordonnés. Par contre l'entrelacement induit non pas un ordre, mais des ordres possibles : a suivi de b ou bien b suivi de a . Le débat porte donc sur le fait que « sans ordre » n'est pas toujours équivalent à « dans un ordre quelconque ». Ce point est important puisque nous nous intéressons à la durée d'ensembles de franchissements de transitions : selon qu'ils sont ordonnés ou non, la durée associée sera différente. C'est donc une caractérisation correcte du parallélisme que nous souhaitons : exprimer tous les liens de causalité existants et seulement ceux-là. C'est pourquoi nous recherchons l'aide de la logique à la fois comme outil de représentation de cette causalité et comme outil de raisonnement.

En plus des applications de la logique temporelle qui suppose de construire le graphe d'états, l'idée d'associer réseaux de Petri et logique est apparue il y a plusieurs années déjà. Mais, dans tous les cas, ces approches basées sur la logique propositionnelle classique ne permettent pas de traiter n'importe quel type de réseaux de Petri : ils sont tantôt saufs, tantôt non bornés, non cycliques, etc. Ces difficultés sont liées à la monotonie de la logique classique : si une proposition est prouvée comme étant vraie, elle le reste, quel que soit l'usage qui en est fait. Or, pour les problèmes que nous représentons à l'aide des réseaux de Petri, nous avons besoin de caractériser des ressources qui sont produites puis consommées ou bien allouées puis libérées. De plus, il est nécessaire de pouvoir les compter ce que ne permet pas la logique classique puisque, par définition, $a \text{ ET } a$ est équivalent à a (cette équivalence, issue de la « règle de contraction » peut aussi être vue comme l'idempotence du ET). C'est pourquoi nous utilisons une logique non classique, la logique linéaire, capable de représenter toutes ces notions liées aux ressources. Les propositions ne sont plus des vérités éternelles : elles peuvent être consommées par le raisonnement, d'où la non-monotonie.

Après avoir consacré un paragraphe à la présentation de la logique linéaire, nous présenterons une méthode reliant logique linéaire et structure des réseaux de Petri. Nous présenterons ensuite notre correspondance entre franchissements de transitions (ce sont les événements) et formules de logique linéaire (appelées séquents) dont l'objectif est de caractériser le fonctionnement du réseau et non pas seulement la structure. Un des points essentiels de cette démarche est l'équivalence entre la prouvabilité des séquents de logique linéaire et l'accessibilité dans le réseau de Petri correspondant. C'est sur ce résultat que nous basons notre travail. Enfin nous expliciterons comment construire

des preuves de séquents qui correspondent à ce que nous appelons des scénarios, c'est-à-dire des ensembles de franchissements de transitions. Il est ainsi possible d'obtenir les liens de causalité existant entre les franchissements de transitions inclus dans le scénario : ce sont ces liens de causalité qui nous permettent d'obtenir la durée du scénario. Il faut noter que le raisonnement temporel est conduit sous forme symbolique, ce qui le rend plus général.

6.2. Logique linéaire

6.2.1. *Les bases : une logique qui manipule les ressources*

La logique linéaire a été proposée par J.Y. Girard [GIRAR 87] comme une variante de la logique propositionnelle : cette logique s'avère être « non monotone ». Elle est basée sur le « calcul des séquents » qui a été initialement introduit par Gentzen (en 1934) pour la logique classique : Girard l'a choisi comme outil privilégié de définition et de conduite de preuve en logique linéaire. Le point fondamental est l'abandon des deux règles de la logique propositionnelle classique que sont la « règle de contraction » et la « règle d'affaiblissement » dans le but, notamment, de mieux exprimer la causalité. Le résultat en est une redéfinition des connecteurs ET, OU et IMPLIQUE : ainsi le calcul des séquents associé conduit à un système capable de traiter de la dynamique des systèmes. Les propositions de logique linéaire sont en fait des ressources, qui peuvent être consommées ou produites. Une déduction de logique linéaire consomme les propositions de ses prémisses et produit les propositions de sa conclusion. Si une proposition est utilisée deux fois dans une déduction, il faut qu'elle ait été disponible ou produite deux fois avant cette déduction. Cet aspect dynamique introduit bien entendu une notion sous-jacente de temps et donc d'état des ressources. C'est cette capacité à raisonner sur les notions de ressources, d'exemplaires multiples d'une même ressource, de production/consommation, d'état, etc. qui rend la logique linéaire *a priori* attrayante pour l'étude des réseaux de Petri : ces notions sont l'essence même de ces réseaux.

6.2.2. *Les connecteurs et leur interprétation*

Comme indiqué ci-dessus, la logique linéaire abandonne deux règles essentielles de la logique propositionnelle classique. La règle de contraction conduit à l'idempotence du ET puisqu'il est possible de dériver $a \text{ ET } a$ à partir de a . Dans une optique de représentation de ressources disponibles, on conçoit aisément qu'il faille différencier deux exemplaires, d'une même source, d'un exemplaire unique. La seconde règle qui est abandonnée est celle dite d'affaiblissement. Elle permet notamment de déduire a à partir de $a \text{ ET } b$ et donc de traiter de causes sans effet, c'est-à-dire d'oublier des hypothèses dans un raisonnement, d'où son rejet. La disparition de ces deux règles conduit à une refonte complète des connecteurs classiques ET et OU. Ces changements entraînent aussi une nouvelle sémantique pour l'implication et la négation. L'ensemble

des connecteurs de la logique linéaire est en fait réparti en trois groupes : les connecteurs multiplicatifs, additifs et exponentiels mais nous n'allons présenter ici que ceux utilisés pour notre travail actuel sur les réseaux de Petri, les connecteurs multiplicatifs *FOIS* et *IMPLICATION LINEAIRE*. Les connecteurs additifs peuvent être utilisés pour exprimer les situations de choix mais le travail essentiel sur l'accessibilité dans les réseaux de Petri repose sur l'utilisation des connecteurs multiplicatifs. Le lecteur intéressé pourra trouver une présentation détaillée (orientée réseaux de Petri) dans [GIRAU 97a] ou bien dans l'un des articles de synthèse consacrés à la logique linéaire par Girard [GIRAR 95a, GIRAR 95b].

Le connecteur \multimap (*IMPLICATION LINEAIRE*) exprime la causalité entre la production et la consommation de ressources. Par exemple $A \multimap B$ traduit le fait qu'en consommant la proposition A , la proposition B est produite. C'est donc le résultat d'une action qui est ainsi modélisé.

Le connecteur \otimes (*FOIS*) est une forme de conjonction (correspondant en partie au connecteur ET de la logique classique) qui exprime l'accumulation de ressources : ainsi la proposition $A \otimes A$ représente la présence de deux exemplaires de A . Cette proposition n'est pas équivalente à la proposition A .

Pour illustrer quelques conséquences de ce changement par rapport à la logique propositionnelle classique, considérons deux exemples simples. A partir de la proposition $A \otimes A$ et de $A \multimap B$ il sera possible de déduire $A \otimes B$ mais pas B : le second exemplaire de A ne peut pas être « oublié ». Symétriquement, partant de A et de $A \otimes A \multimap B$ il ne sera pas possible de déduire B car un seul exemplaire de A n'est pas suffisant à l'exécution de l'action. Dans les deux cas, l'usage des connecteurs classiques (ET et IMPLIQUE) aurait permis ces déductions.

6.2.3. Calcul des séquents

Toutes les déductions de logique linéaire peuvent se faire dans un cadre clair et formel, le calcul des séquents. Un séquent est une formule : $\Gamma, X \vdash Y, \Delta$ dans laquelle les parties gauche et droite sont constituées par des suites finies de formules. La virgule est un métasybole dont le sens est fourni par les règles associées : elle représente la conjonction dans la partie gauche et la disjonction dans la partie droite (la conjonction « Γ et X » permet de déduire « Δ ou Y »). Dans le cadre du travail présenté ici, le membre droit sera toujours réduit à une seule formule, ce qui place notre travail dans le cadre d'un sous-ensemble de la logique linéaire, appelé MILL (*Multiplicative Intuitionistic Linear Logic*). Prouver un séquent, c'est montrer qu'il est syntaxiquement correct. Cette preuve consiste à montrer qu'il peut être entièrement construit à partir d'un ensemble de règles introduisant les atomes (propositions) et les connecteurs. Les règles présentées ci-dessous sont donc celles du fragment MILL.

Pour chaque règle, le séquent à prouver est écrit en dessous de la barre tandis que le (ou les) séquent(s) utilisé(s) pour cette preuve est (sont) écrit(s) au-dessus. Le calcul des séquents linéaires est entièrement défini par un ensemble de règles, réparties en trois groupes : le « groupe identité » et le « groupe structurel » expriment des propriétés intrinsèques à la logique tandis que le « groupe logique » définit une règle d'introduction à gauche et une règle d'introduction à droite pour chacun des connecteurs spécifiques à la logique considérée (à savoir ici \otimes et \multimap).

$$\begin{array}{c}
 \text{Groupe identité} \\
 \frac{}{A \vdash A} \text{Identité} \qquad \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B} \text{Cut} \\
 \\
 \text{Groupe structurel} \\
 \frac{\Gamma, A, B \vdash \Delta}{\Gamma, B, A \vdash \Delta} \text{Echange} \\
 \\
 \text{Groupe logique} \\
 \frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \otimes B \vdash \Delta} \otimes L \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \otimes R \\
 \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C} \multimap L \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \multimap R
 \end{array}$$

Figure 6.1. Les règles du calcul des séquents de MILL

Partant de ces règles, une preuve est conduite de bas en haut en posant comme racine de l'arbre de preuve le séquent à démontrer. Ce séquent est prouvé si chaque feuille de l'arbre se termine par un séquent axiome (dans le sous-ensemble MILL, le seul axiome est le séquent identité). Par exemple, la démonstration du séquent $A, A \multimap B \vdash B$ est présentée dans l'arbre suivant. Il est à noter que ce séquent exprime la consommation de A mais aussi celle de $A \multimap B$ pour produire B. La formule implicative est ici vue comme une ressource consommable ou, plus exactement, comme une instance d'action [GIRAU 97a].

$$\frac{\frac{}{A \vdash A} Id \quad \frac{}{B \vdash B} Id}{A, A \multimap B \vdash B} \multimap L$$

Figure 6.2. L'arbre de preuve de $A, A \multimap B \vdash B$

L'un des points importants du calcul des séquents est l'usage de la règle de coupure (règle CUT). Comme la logique propositionnelle classique, la logique linéaire vérifie un théorème important : « la règle de coupure est redondante » [GIRAR 87]. Donc, si un séquent est prouvable, il existe une démonstration qui n'utilise pas la règle CUT. Pourquoi cet intérêt vis-à-vis de la non utilisation de la coupure ? Paradoxalement, c'est

l'une des règles qui apparaît la plus naturelle puisqu'elle explicite le fait qu'une preuve peut se faire par enchaînements de déductions élémentaires. Mais, lue de bas en haut, la règle de coupure est la seule qui introduise dans les prémisses une formule exogène (la formule A dans la règle de la figure 6.1) au séquent à prouver : on retrouve donc le même effet que celui de l'introduction d'un lemme intermédiaire dans une preuve mathématique. Une preuve sans coupure est donc plus directe. Nous allons examiner dans le paragraphe suivant quelques-unes des approches reliant réseaux de Petri et logique linéaire et nous verrons que, suivant les cas, elles utilisent ou non la règle de coupure.

6.3. Réseaux de Petri et logique linéaire

6.3.1. Les différentes approches

Dès les premières publications sur la logique linéaire, un certain nombre d'auteurs se sont intéressés aux liens existant avec les réseaux de Petri, mais il faut noter que l'essentiel de ces publications était plutôt destiné à la communauté des logiciens, les réseaux de Petri ne constituant alors qu'une interprétation de la logique linéaire. D'ailleurs, ce n'est que dans les publications de 1995 que Girard emploie explicitement les termes « états » et « transitions » quand il parle de la logique linéaire. Une analyse exhaustive et argumentée de ces publications est disponible dans [GIRAU 97a]. Ici, nous n'allons nous intéresser qu'aux approches qui ont explicitement cherché à modéliser le fonctionnement des réseaux de Petri, et, par là même, ont tenté de fournir un outil d'analyse supplémentaire à cette théorie. L'une des approches fréquemment suivies consiste à modéliser la structure du réseau de Petri, sans représentation explicite du marquage. Cela conduit à des résultats intéressants mais qui restent au niveau structurel. Notre proposition de représentation, présentée ensuite, diffère en ce sens que ce sont les marquages et les franchissements de transitions qui sont modélisés dans les séquents.

6.3.2. Approche sans marquage

Le travail le plus représentatif dans cette voie est celui de Gehlot et Gunter [GEH 92, GUN 89]. Pour un réseau de Petri donné :

- un atome propositionnel P est associé à toute place p du réseau,
- un séquent axiome est défini pour chaque transition t du réseau de Petri, à partir des vecteurs Pre(t) et Post(t), et rajouté à la liste des axiomes du calcul des séquents. Si Pre(pi,t) vaut n, la proposition Pi apparaît n fois dans la formule de gauche et il en est de même pour les places de sortie.

$$\bigotimes_{i \in \text{Pre}(pi,t)} P_i \vdash \bigotimes_{o \in \text{Post}(po,t)} P_o$$

Chaque axiome traduit ainsi la variation de marquage due au franchissement de la transition. Remarquons que la logique linéaire caractérise les boucles élémentaires avec exactitude. Par exemple, la transition t_1 ayant A et B comme places d'entrée et B et C comme places de sortie sera caractérisée par le séquent $A \otimes B \vdash B \otimes C$. Par les règles du connecteur \otimes , la preuve de ce séquent ne peut en aucune façon fournir une information pour la preuve de $A \vdash B$. En d'autres termes, le séquent $A \otimes B \vdash B \otimes C$, correspondant à la transition, « n'oublie » pas la boucle élémentaire sur la place B et n'est pas « simplifiable ».

Le principe de cette méthode est donc de rajouter autant d'axiomes propres que de transitions présentes dans le réseau. Cette représentation, très structurée, se justifie par le fait qu'une transition n'est pas une ressource consommable et donc n'est pas modélisée par une proposition qui est, par définition, consommable. La preuve d'un séquent se termine lorsque toutes les feuilles conduisent à un axiome qui est, soit l'axiome *Identité*, soit l'un des axiomes générés par le réseau de Petri.

L'objectif du travail de Gehlot et Gunter est double : ainsi il peut tout d'abord prouver l'accessibilité entre certains états du réseau mais il peut aussi caractériser correctement le parallélisme contenu dans un réseau de Petri.

Pour illustrer son approche, considérons le réseau de Petri de la figure 6.3. Partant des formules de logique linéaire, il est par exemple possible de prouver le séquent $A \otimes D \vdash C \otimes F$ qui exprime l'accessibilité entre les états $A \otimes D$ et $C \otimes F$: la preuve par le calcul des séquents est représentée figure 6.4. Bien évidemment, ce réseau contient un parallélisme structurel : nous allons voir comment il peut être exhibé lors de la preuve du séquent. Nous verrons aussi par la suite que cette méthode ne permet pas de caractériser le parallélisme induit par le marquage.

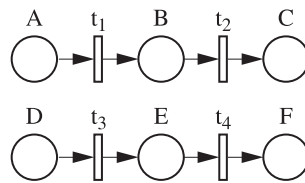


Figure 6.3. Un réseau de Petri avec parallélisme structurel

Dans la preuve du séquent (figure 6.4), la règle $\otimes R$ est utilisée lorsque les deux transitions (ou ensembles de transitions) ne sont pas structurellement en séquence et peuvent donc être franchies simultanément (par exemple t_1 et t_3 , noté $(t_1//t_3)$). La règle $\otimes L$ est une simple règle de ré-écriture tandis que la règle *Cut* est utilisée pour exprimer la mise en séquence de (la causalité entre) deux ensembles de transitions (ici $(t_1//t_3)$ est mis en séquence avec $(t_2//t_4)$). L'arbre de preuve ainsi développé fournit donc une

information supplémentaire à cette seule preuve : en l'analysant, il est possible d'en déduire l'ordre de franchissement des transitions.

$$\frac{\frac{\frac{A \vdash B \quad D \vdash E}{A, D \vdash B \otimes E} \otimes R \quad \frac{B \vdash C \quad E \vdash F}{B, E \vdash C \otimes F} \otimes R}{A \otimes D \vdash B \otimes E} \otimes L \quad \frac{B \vdash C \quad E \vdash F}{B \otimes E \vdash C \otimes F} \otimes L}{A \otimes D \vdash C \otimes F} \text{Cut}$$

Figure 6.4. L'arbre de preuve de $A \otimes D \vdash C \otimes F$

L'arbre ci-dessus prouve l'accessibilité entre les deux marquages considérés et exprime un ordre partiel $(t1//t3); (t2//t4)$ entre les transitions (où l'opérateur « ; » note la mise en séquence). Mais, dans bien des cas, cet arbre de preuve n'est pas unique.

L'auteur propose ensuite, dans un deuxième temps, des règles de ré-écriture de l'arbre de preuve afin qu'il exprime « le parallélisme maximal ». Si nous reprenons l'exemple précédent, ces règles conduisent à l'arbre de la figure 6.5, où l'ordre partiel est alors $(t1; t2)//(t3; t4)$. C'est effectivement celui qui exprime le mieux le parallélisme structurel du réseau.

$$\frac{\frac{A \vdash B \quad B \vdash C}{A \vdash C} \text{Cut} \quad \frac{D \vdash E \quad E \vdash F}{D \vdash F} \text{Cut}}{A \otimes D \vdash C \otimes F} \otimes R$$

Figure 6.5. L'arbre de preuve avec parallélisme maximal de $A \otimes D \vdash C \otimes F$

Ces premiers résultats sont très intéressants mais ils comportent deux défauts : tout d'abord, les preuves ne se font pas dans le cadre strict de la logique linéaire puisqu'elle est systématiquement augmentée par autant d'axiomes que de transitions dans le réseau. Cette extension des axiomes conduit à une preuve qui ne peut plus se faire sans utiliser la règle de coupure, car elle n'est plus redondante, et cela peut donc compliquer le mécanisme de preuve.

Le second défaut de cette méthode est une conséquence de la représentation choisie : la notion de marquage n'est jamais directement traduite dans les séquents considérés, et n'est donc jamais prise en compte. Ceci ne remet pas en cause les résultats concernant la prouvabilité des séquents, et donc l'accessibilité dans le réseau de Petri, mais par contre les informations sur l'ordre partiel, issues de l'arbre de preuve, s'avèrent erronées : la méthode proposée ne sait détecter que les seules relations de parallélisme structurel. Les relations de parallélisme induites par le marquage, que nous appelons « parallélisme dynamique du réseau », ne peuvent pas être caractérisées car elles

dépendent du marquage. Reprenons le réseau de la figure 6.3 et considérons le séquent $A \otimes B \otimes D \vdash B \otimes C \otimes F$: pour atteindre le marquage $B \otimes C \otimes F$ à partir de $A \otimes B \otimes D$, il faut franchir chaque transition du réseau exactement une fois. En utilisant cette méthode, ce séquent sera prouvé mais le processus de ré-écriture pour déterminer le parallélisme maximal conduira au même ordre partiel que celui de $A \otimes D \vdash C \otimes F$. Pourtant, suivant la présence ou non d'un jeton dans la place B, les liens de causalité entre les franchissements sont différents : si un jeton est présent, la transition t_2 peut être franchie indépendamment de t_1 . Puisque nous sommes intéressés par une caractérisation temporelle, il est indispensable de caractériser aussi bien le parallélisme dynamique que le parallélisme structurel d'un réseau. C'est ce qui nous a conduits à proposer une autre traduction du fonctionnement des réseaux de Petri en logique linéaire.

6.3.3. Approche avec marquage

L'approche que nous avons développée [GIRAU 97a, GIRAU 97b, KUN 97], reste dans le strict cadre de la logique linéaire (pas d'axiomes propres rajoutés) et essaie de modéliser au mieux le comportement d'un réseau de Petri. La conséquence de ce cadre est que les preuves sont conduites sans utiliser la règle de coupure. Pour ce faire, le premier changement concerne les transitions : elles ne sont plus représentées par un séquent mais par une proposition implicative qui exprime le fait que la transition est franchie. Nous représentons ainsi des instances de franchissement de transitions et non pas des transitions, et les propositions implicatives pourront être consommées au cours de la preuve, ce qui indiquera que la transition est effectivement franchie. Second changement important, le marquage dans lequel a lieu ce franchissement est explicitement spécifié.

Pour un réseau de Petri donné :

- un atome propositionnel P est associé à toute place p du réseau,
- une formule en \otimes est associée à tout marquage ainsi qu'à toute précondition (Pre) ou postcondition (Post) de transition,
- une formule implicative est définie pour chaque transition t du réseau de Petri, à partir des vecteurs $Pre(t)$ et $Post(t)$. Si $Pre(p_i, t)$ vaut n , la proposition P_i apparaît n fois dans la partie gauche et il en est de même pour les places de sortie.

$$t : \bigotimes_{i \in Pre(p_i, t)} P_i \multimap \bigotimes_{o \in Post(p_o, t)} P_o$$

Tout séquent de la forme $M, t_1, \dots, t_n \vdash M'$ exprime l'accessibilité entre deux marquages M et M' , en spécifiant quelles sont les transitions franchies (t_i représente la formule implicative correspondant à la transition). La preuve est ensuite conduite, de manière canonique, dans le strict cadre de la logique linéaire : nous allons l'illustrer

au travers d'un exemple. Considérons le réseau de Petri de la figure 6.6 et le séquent exprimant l'accessibilité entre $A \otimes B$ et $B \otimes C$ grâce au franchissement de chacune des transitions t_1 et t_2 .

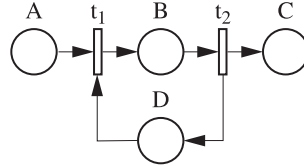


Figure 6.6. Un réseau de Petri sans parallélisme structurel

L'arbre, donné en figure 6.7, permet d'illustrer comment sont conduites les preuves. Le franchissement effectif d'une transition est toujours matérialisé par l'application de la règle $\multimap L$: le séquent généré en partie gauche représente les jetons consommés lors de ce franchissement tandis que le séquent droit représente ce qui reste à prouver. Si la transition franchie a une seule place d'entrée, le séquent gauche est une identité ($B \vdash B$ par exemple pour la transition t_2 qui est la première franchie) mais il est du type $A, D \vdash A \otimes D$ si elle comporte plus d'une place en entrée. C'est le seul cas où nous utiliserons la règle $\otimes R$ (phénomène de synchronisation en entrée de la transition). La règle $\otimes L$ est une simple règle de réécriture. La preuve est terminée lorsque le séquent droit ne comporte plus de formules de transitions et se termine par une (des) identité(s) (de même que précédemment, si le marquage final contient plusieurs jetons, une application de la règle $\otimes R$ permet de terminer la preuve pour ce séquent droit).

$$\begin{array}{c}
 \frac{\frac{\frac{}{A \vdash A} Id \quad \frac{}{D \vdash D} Id}{A, D \vdash A \otimes D} \otimes R \quad \frac{\frac{}{B \vdash B} Id \quad \frac{}{C \vdash C} Id}{B, C \vdash B \otimes C} \otimes R}{A, C, D, A \otimes D \multimap B \vdash B \otimes C} \multimap L \\
 \frac{\frac{}{B \vdash B} Id \quad \frac{}{A, C \otimes D, A \otimes D \multimap B \vdash B \otimes C} \otimes L}{A, B, A \otimes D \multimap B, B \multimap C \otimes D \vdash B \otimes C} \multimap L \\
 \frac{}{A \otimes B, A \otimes D \multimap B, B \multimap C \otimes D \vdash B \otimes C} \otimes L
 \end{array}$$

Figure 6.7. L'arbre de preuve canonique de $A \otimes B, t_1, t_2 \vdash B \otimes C$

La procédure pour conduire la preuve est donc relativement simple, puisque nous n'utilisons que très peu de règles du calcul des séquents et, surtout, parce que nous travaillons sans la règle de coupure : la taille de cet arbre est strictement proportionnelle au nombre de franchissements inclus dans le séquent.

L'arbre canonique est obtenu par l'algorithme suivant :

APPLIQUER la règle $\otimes L$ autant de fois que nécessaire pour transformer le marquage de départ en une liste d'atomes séparés par le meta-connecteur « , »

TANT QUE la règle $\multimap O L$ est applicable (*c'est-à-dire si le marquage d'entrée d'une ou plusieurs formules de transitions est inclus dans la liste des atomes de l'étape courante*)

- appliquer la règle $\multimap O L$ à la formule implicative candidate de plus petit ordre lexicographique,
- terminer la preuve du séquent gauche généré en utilisant, si nécessaire, la règle $\otimes R$,
- appliquer, si nécessaire, la règle $\otimes L$ au marquage produit dans le séquent droit (partie gauche de celui-ci).

FIN TANT QUE

Cet algorithme conduit à une preuve du séquent de départ si toutes les formules implicatives correspondant aux franchissements de transitions ont été éliminées et donc si toutes les feuilles de l'arbre se terminent par le séquent identité.

D'autres séquents intéressants sont prouvables pour ce réseau. Par exemple, le séquent $A \otimes A \otimes D, t1, t1, t2 \vdash B \otimes C$ exprime ce qui se passe lorsque $t1$ est franchie deux fois alors que $t2$ n'est franchie qu'une seule fois. C'est donc un multi-ensemble de franchissements de transitions qui est exprimé par un séquent.

6.3.4. *Equivalence entre l'accessibilité dans un réseau de Petri et la prouvabilité d'un séquent en logique linéaire*

Le point important dans cette démarche de preuve est bien sûr la canonicité mais elle n'aurait aucun intérêt si cette preuve n'était pas reliée à la notion d'accessibilité pour les réseaux de Petri.

Il a été montré que tout séquent du type $M, t1, \dots, tn \vdash M'$ est prouvable si et seulement si l'accessibilité entre les marquages M et M' est vérifiée. Au multi-ensemble considéré correspond un ensemble de séquences (différent seulement par leur ordre) [GIRAU 97a] tel que pour toute séquence, notée *sequence*, de cet ensemble, on a :

$$M \xrightarrow{\text{sequence}} M' \equiv M, t1, \dots, tn \vdash M'$$

La question est donc de savoir ce qu'apporte cette méthode par rapport aux techniques usuelles des réseaux de Petri. La première façon de traiter l'accessibilité dans

la théorie des réseaux est de résoudre l'équation caractéristique :

$$M' = M + C \cdot \bar{s}$$

Malheureusement, cette équation ne fournit qu'une condition nécessaire (mais non suffisante) d'accessibilité. De plus, le vecteur caractéristique s ne fournit que le multi-ensemble des transitions concernées : aucun ordre de franchissement ne peut être déterminé. L'information obtenue pour s est donc seulement nécessaire et non ordonnée.

La seconde méthode pour obtenir une information sur l'accessibilité est de déterminer une séquence seq telle que :

$$M \xrightarrow{seq} M'$$

Dans ce cas, l'information obtenue est totalement ordonnée et la condition est nécessaire et suffisante. Mais cet ordre total (quelles que soient les relations de causalité réelles, les franchissements sont effectués les uns après les autres) ne permet pas de distinguer les liens de causalité réels que nous recherchons.

Que pouvons-nous attendre de cette nouvelle caractérisation en logique linéaire ? Tout d'abord, la preuve par le calcul des séquents permet d'obtenir rapidement et simplement des réponses exactes (condition nécessaire et suffisante) pour l'accessibilité entre deux marquages, en fournissant simultanément le multi-ensemble des transitions impliquées. De plus, il est possible d'extraire des informations sur l'ordre de ces franchissements à partir de l'arbre de preuve canonique. Ces informations de causalité ne sont pas fournies par la nature des règles du calcul des séquents utilisées mais par les jetons produits et consommés. Pour obtenir ces liens de causalité, nous allons associer à chaque jeton un label : ce label sera temporel et nous permettra ainsi de calculer la durée de franchissement du multi-ensemble. Notre démarche ne comporte donc aucune étape de ré-écriture de l'arbre pour déterminer l'ordre partiel. Quel que soit l'arbre développé, l'information apportée par les labels serait identique : c'est pourquoi nous ne construisons que l'arbre canonique.

Considérons à nouveau le réseau de la figure 6.6 et le séquent dont l'arbre de preuve est fourni figure 6.7, $A \otimes B, t1, t2 \vdash B \otimes C$: nous voyons, grâce à cet arbre, que le franchissement de $t1$ (règle \multimap L pour la formule $A \otimes D \multimap B$) ne peut se faire qu'après celui de $t2$ (règle \multimap L pour la formule $B \multimap D \otimes C$) puisque le jeton de la place D , nécessaire au tir de $t1$, est produit par le franchissement de $t2$. Suivant le marquage, ces liens de causalité peuvent être différents. Considérons le séquent $A \otimes B \otimes D, t1, t2 \vdash B \otimes C \otimes D$: il traduit lui aussi un seul franchissement de chacune des transitions, mais le marquage de départ est différent. Dans ce cas, aucun lien de causalité ne relie ces deux franchissements (cas de parallélisme dynamique) : en développant l'arbre de preuve, on s'apercevrait que chacune des deux transitions consomme des jetons déjà présents au marquage initial. Ces informations de causalité sont capitales pour une

évaluation temporelle et nous allons les décrire au moyen de labels. Nous n'allons pas, comme le fait Gehlot, réécrire l'arbre pour déterminer une formule exprimant l'ordre partiel sous la forme d'une expression sur les transitions car cette méthode est limitée aux capacités d'expression des ordres série parallèle [VAL 82]. Le réseau de Petri de la figure 6.10 est un exemple typique (ordre dit « en N ») des limitations des ordres série parallèle. Nous allons montrer qu'en associant des labels aux jetons nous pouvons quand-même obtenir des relations causales exactes pour ce type de réseaux.

6.4. Évaluation temporelle de scénarios

6.4.1. Définitions : scénario, labels, étape courante

Nous appelons scénario d'un réseau de Petri, un multi-ensemble de franchissements de transitions associé aux marquages M et M' tels que M' soit accessible depuis M par le tir des différentes transitions considérées. L'ensemble de ces informations est représenté dans un séquent du type $M, t_1, \dots, t_n \vdash M'$. Les réseaux de Petri considérés sont des réseaux temporisés : une durée est associée à chaque transition et nous notons d_i la durée associée à toute transition t_i . Pour déterminer les liens de causalité, un label temporel est associé à chaque jeton d'un marquage : il correspond à sa date de production et, dans le séquent nous noterons par exemple $A(dk)$ pour indiquer que ce jeton de la place A a été produit à la date dk .

La preuve conduite à la figure 6.7 montre que sont manipulés des marquages (jetons reliés par le connecteur \otimes) mais aussi des listes de jetons reliés par le métaconnecteur « , » (A, C, D par exemple). Ces listes de jetons avec leurs labels associés sont appelées « étapes courantes » car elles correspondent à des étapes de raisonnement mais elles diffèrent des marquages [PRA 99b]. L'existence d'une étape notée $A(d_i)$, $B(d_j)$, $C(d_k)$ ne garantit pas que le marquage $A \otimes B \otimes C$ soit effectivement atteint : cela dépend des valeurs des labels temporels. Par contre l'existence de cette étape garantit qu'un jeton est produit dans la place A à la date d_i , qu'un autre est produit en B à la date d_j et un troisième en C à la date d_k sans qu'aucune relation de causalité ne lie ces productions.

La méthode sera détaillée dans la suite, mais, pour illustrer ce travail, reprenons la preuve de la figure 6.7 en y adjoignant les labels temporels (le marquage initial ayant été produit à la date 0, ce qui est noté $(A \otimes B)(0)$). Seul le séquent à démontrer contient un marquage en partie gauche, les séquents intermédiaires de l'arbre contiennent des étapes courantes.

Cet exemple permet de donner un premier aperçu de la méthode. En lisant cet arbre de bas en haut, on repère d'abord le franchissement de t_2 par consommation de B puis celui de t_1 . Lors du franchissement d'une transition (règle $\multimap L$), la date de production du marquage atteint est égale à la durée de la transition plus la date de consommation au plus tôt des jetons des places d'entrée. Ces jetons ayant éventuellement été produits

$$\begin{array}{c}
\frac{\frac{\frac{}{A \vdash A} \text{Id} \quad \frac{}{D \vdash D} \text{Id}}{A(0), D(d2) \vdash A \otimes D} \otimes R \quad \frac{\frac{}{B \vdash B} \text{Id} \quad \frac{}{C \vdash C} \text{Id}}{B(d2+d1), C(d2) \vdash B \otimes C} \otimes R}{\frac{\frac{}{B \vdash B} \text{Id} \quad \frac{A(0), C(d2), D(d2), A \otimes D \multimap B \quad \vdash \quad B \otimes C}{A(0), (C \otimes D)(d2), A \otimes D \multimap B \quad \vdash \quad B \otimes C} \otimes L}{A(0), B(0), A \otimes D \multimap B, B \multimap C \otimes D \quad \vdash \quad B \otimes C} \multimap L} \otimes L} \\
\frac{}{(A \otimes B)(0), A \otimes D \multimap B, B \multimap C \otimes D \quad \vdash \quad B \otimes C} \otimes L
\end{array}$$

Figure 6.8. L'arbre de preuve précédent avec labels

à des dates différentes, cette date de consommation est égale au maximum de leurs dates de production. Par exemple, le tir de t_1 consomme $A(0)$ et $D(d_2)$: la date de franchissement au plus tôt est $\max(0, d_2) = d_2$. Le jeton B sera donc produit à la date $(d_2 + d_1)$. La durée de ce scénario est égale au maximum des dates associées aux jetons B et C, c'est-à-dire $(d_2 + d_1)$. Lors de l'application de la règle $\otimes L$, le label qui était associé au marquage est associé individuellement à chaque jeton. Cet exemple simple ne comporte pas de conflits mais, dans le cas général, vont apparaître des conflits de transitions mais aussi des conflits de jetons.

6.4.2. Conflits de transitions et de jetons

La notion de conflit de transitions est bien connue des utilisateurs des réseaux de Petri. Nous la retrouvons très naturellement au niveau du séquent linéaire suivant :

$$Etape, M_{e1} \multimap M_{s1}, \dots, M_{en} \multimap M_{sn} \vdash \text{Marquage}$$

DÉFINITION 6.1.— *Il y a conflit entre deux transitions t_i et t_j si l'un des atomes de l'étape courante Etape est inclus dans les préconditions de deux transitions du séquent (par exemple M_{ei} et M_{ej}). Les conflits considérés ici ne sont pas forcément des conflits effectifs puisque aucune condition n'est requise quant à la possibilité d'appliquer immédiatement la règle $\multimap L$ aux transitions considérées. En effet, l'ordre des franchissements de transitions peut devenir significatif en faisant apparaître un conflit qu'il soit ou non effectif.*

Un conflit effectif nécessite de choisir la transition tirée puisque le comportement temporel du système diffère d'une alternative à l'autre. En termes d'arbre de preuve, tout conflit effectif de transitions entraîne la construction de deux arbres différents. Prenons pour exemple le simple réseau de la figure 6.9a et le scénario conduisant du marquage $A \otimes B$ à la date 0 au marquage $C \otimes D$ en tirant exactement une fois chacune des transitions. Puisque t_2 et t_3 sont en conflit effectif, il sera nécessaire de construire deux arbres de preuve. La durée des scénarios correspondants sera $\max(d_2, d_1 + d_3)$ si le conflit est résolu par le tir de t_2 en premier et $\max(d_3, d_1 + d_2)$ dans le cas contraire.

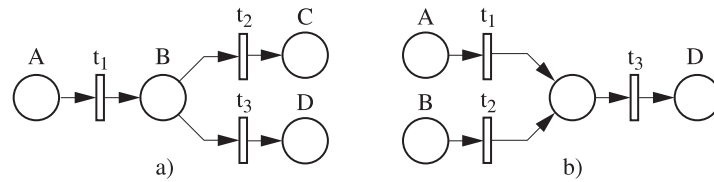


Figure 6.9. *Conflicts de transitions et de jetons*

Il peut paraître surprenant de détecter aussi bien les conflits potentiels (transitions présentes dans le scénario mais non encore candidates à l'application de la règle —oL) que les conflits réels mais cela s'avère nécessaire pour obtenir des informations temporelles valides. En effet les travaux de [SIL 90] ont montré que dans le cas d'un système de production, par exemple, la durée la plus courte n'est pas toujours obtenue par le tir au plus tôt des transitions. Il peut être nécessaire de réserver une ressource critique pour une opération qui n'est pas encore possible mais qui le deviendra bientôt. Ce cas correspond bien au fait de retarder le franchissement d'une transition franchissable mais appartenant à un conflit potentiel.

Le cas dual du conflit de transitions est celui du conflit de jetons dont un exemple est fourni à la figure 6.9b. Considérons à nouveau le scénario conduisant du marquage $A \otimes B$ à la date 0 au marquage $C \otimes D$ en tirant exactement une fois chacune des transitions. Le tir de t_3 pourra se faire soit avec le jeton issu du tir de t_1 , soit avec celui issu du tir de t_2 . Il faudra également construire deux arbres de preuve différents dont les durées seront $\max(d_2, d_1 + d_3)$ et $\max(d_1, d_2 + d_3)$. Ce type de conflit intervient donc lorsqu'il n'est pas possible d'ordonner dans le temps les labels de deux jetons d'une même place.

DÉFINITION 6.2.— *Il y a conflit de jetons si l'étape courante Etape contient un nombre de jetons dans la place P supérieur au nombre de jetons nécessaires au tir d'une transition t du scénario et si les labels temporels de ces jetons sont non totalement ordonnés.*

Cette situation peut provenir de plusieurs situations : comme dans le cas de la figure 6.9b, il peut s'agir de plusieurs transitions en amont d'une place commune et franchissables indépendamment l'une de l'autre. Cela ne peut se produire que pour des réseaux non saufs et où existent des transitions multisensibilisées.

6.4.3. Scénarios complètement spécifiés

Les arbres de preuves comprenant des conflits résolus aléatoirement restent valables et permettent de prouver l'accessibilité des marquages considérés. En ce qui concerne les labels temporels associés aux jetons, toute situation de conflit doit être détectée afin que la validité de l'information temporelle issue de ces labels soit clairement

associée à une résolution particulière de chacun des conflits rencontrés. Pour simplifier la construction de l'arbre de preuve avec des labels temporels, nous restreignons dans un premier temps cette recherche au cas des scénarios « complètement spécifiés », c'est-à-dire ceux pour lesquels cet arbre est unique. Nous les appelons ainsi car ils correspondent à des cas où aucune décision externe n'a à être prise.

Une classe simple de réseaux de Petri permet de s'assurer que les scénarios sont complètement spécifiés, ce sont les scénarios qui sont des graphes d'événements : dans cette classe, le tir au plus tôt des transitions est toujours celui qui permet le fonctionnement le plus rapide. Cette contrainte permet d'obtenir un algorithme de calcul de durée de scénario simple. Mais il faut noter que ce ne sont pas les réseaux eux-mêmes qui sont ainsi contraints : le réseau peut être plus complexe mais, si avant de démarrer la construction de l'arbre temporel, nous nous assurons que le scénario lui-même est un graphe d'événements (deux transitions de ce scénario n'ont pas de places d'entrée ou de sortie communes) nous n'aurons pas à rechercher les conflits lors de la construction de l'arbre de preuve. Nous traiterons alors sans problème particulier le cas de marquages non saufs mais aussi de scénarios contenant plusieurs franchissements de la même transition.

6.4.4. Calcul des labels temporels

L'algorithme pour l'obtention de l'arbre de preuve canonique avec labels temporels est le même que pour l'arbre sans label en rajoutant quelques règles pour les labels :

VERIFIER que le fragment du réseau correspondant au scénario est un graphe d'événements,

APPLIQUER la règle $\otimes L$: le label temporel des atomes de l'étape courante est égal à 0,

TANT QUE la règle $\text{—}oL$ est applicable

– appliquer la règle $\text{—}oL$ à l'une des transitions candidates : *le label temporel associé au marquage produit est égal au maximum des labels des atomes consommés augmenté de la durée associée à cette transition ; les autres labels sont inchangés,*

– terminer la preuve du séquent gauche généré en utilisant, si nécessaire, la règle $\otimes R$,

– appliquer, si nécessaire, la règle $\otimes L$ au séquent droit : *le label temporel des atomes est égal à celui du marquage.*

FIN TANT QUE

Suivant le cas :

– toutes les transitions ont été traitées : le séquent est prouvé (accessibilité validée) et la durée du scénario est égale au maximum des labels associés aux atomes du marquage final (date de production au plus tôt de ce marquage) ;

– certaines transitions n’ont pu être franchies : le séquent n’est pas prouvable car l’accessibilité entre les marquages par le multi-ensemble considéré n’est pas possible. Le blocage et sa date peuvent être repérés grâce aux labels.

6.4.5. Exemples

Pour illustrer l’intérêt de cette approche nous allons considérer le réseau de Petri de la figure 6.10 : nous examinerons plusieurs scénarios comportant tantôt du parallélisme structurel, tantôt du parallélisme structurel et dynamique. Ce réseau de Petri est particulièrement intéressant car il fait partie des réseaux dits « en N » pour lesquels aucune relation d’ordre partiel du type série-parallèle n’est capable de décrire exactement son parallélisme structurel.

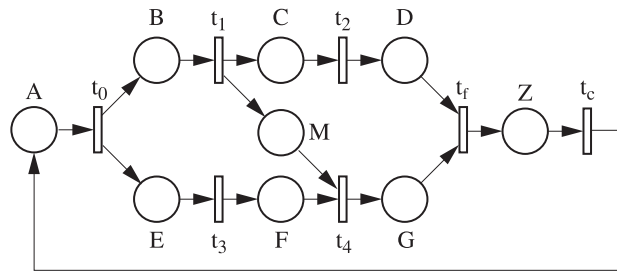


Figure 6.10. Réseau de Petri « en N »

Considérons un premier scénario qui contient un tir de chaque transition à partir du marquage A et conduit au marquage Z : les liens de causalité ne sont dans ce cas que des relations structurelles pourtant aucune relation de type série-parallèle ne peut les exprimer. Par exemple :

- l’ordre $t_0 ; ((t_1 ; t_2) // (t_3 ; t_4)) ; t_f$ n’exprime pas le lien de causalité entre t_1 et t_4 ,
- l’ordre $t_0 ; ((t_1 // t_3) ; (t_2 // t_4)) ; t_f$ rajoute un lien de causalité inexistant $(t_3 ; t_2)$,
- l’ordre $t_0 ; ((t_1 ; t_2) // t_3) ; t_4 ; t_f$ rajoute un lien de causalité inexistant $(t_2 ; t_4)$,
- l’ordre $t_0 ; t_1 ; (t_2 // (t_3 ; t_4)) ; t_f$ rajoute un lien de causalité inexistant $(t_1 ; t_3)$.

Nous allons traiter cet exemple en construisant pas à pas l’arbre de preuve du scénario $A, t_0, t_1, t_2, t_3, t_4, t_f \vdash Z$. Le premier pas de cette preuve est :

$$\frac{A \vdash AId \frac{B(d_0), E(d_0), t_1, t_2, t_3, t_4, t_f \vdash Z}{(B \otimes E)(d_0), t_1, t_2, t_3, t_4, t_f \vdash Z} \otimes L}{A(0), A \multimap B \otimes E, t_1, t_2, t_3, t_4, t_f \vdash Z} \multimap L$$

Les transitions t1 et t3 sont candidates à l'application de la règle —oL mais elles ne sont pas en conflit : c'est donc t1 qui va être considérée (ordre lexicographique) :

$$\frac{B \vdash BId \frac{E(d0), C(d0 + d1), M(d0 + d1), t2, t3, t4, tf \vdash Z}{E(d0), (C \otimes M)(d0 + d1), t2, t3, t4, tf \vdash Z} \otimes L}{B(d0), E(d0), B\text{—o}C \otimes M, t2, t3, t4, tf \vdash Z} \text{—oL}$$

Désormais ce sont t3 et t2 qui sont candidates : la règle —oL est appliquée à t2 :

$$\frac{C \vdash CIdE(d0), M(d0 + d1), D(d0 + d1 + d2), t3, t4, tf \vdash Z}{E(d0), C(d0 + d1), M(d0 + d1), C\text{—o}D, t3, t4, tf \vdash Z} \text{—oL}$$

La règle —oL est appliquée à t3 puisque c'est la seule formule pour laquelle elle est applicable :

$$\frac{E \vdash EIdF(d0 + d3), M(d0 + d1), D(d0 + d1 + d2), t4, tf \vdash Z}{E(d0), M(d0 + d1), D(d0 + d1 + d2), E\text{—o}F, t4, tf \vdash Z} \text{—oL}$$

La règle —oL est ensuite appliquée à t4 :

$$\frac{\frac{\dots \otimes R}{F, M \vdash F \otimes M} \quad G(d0 + \max(d3, d1) + d4), D(d0 + d1 + d2), tf \vdash Z}{F(d0 + d3), M(d0 + d1), D(d0 + d1 + d2), F \otimes M\text{—o}G, tf \vdash Z} \text{—oL}$$

La preuve se termine alors par le traitement de tf :

$$\frac{\frac{\dots \otimes R}{D, G \vdash D \otimes G} \quad Z(\max((d0 + \max(d3, d1) + d4), (d0 + d1 + d2)) + df \vdash Z)^{Id}}{G(d0 + \max(d3, d1) + d4), D(d0 + d1 + d2), D \otimes G\text{—o}Z, \vdash Z} \text{—oL}$$

La durée du scénario est donc égale à $d0 + \max(d3 + d4, d1 + d4, d1 + d2) + df$. Cette formule traduit bien l'existence de trois contraintes de causalité qui sont respectivement t3 avant t4, t1 avant t4 et t1 avant t2, le tout étant précédé de t0 et suivi de tf.

Sur ce même réseau de Petri, considérons quelques autres scénarios qui traduisent des ordres partiels différents. Nous n'allons donner ici que les résultats obtenus sans détailler les arbres de preuve. Le premier scénario considéré contient les mêmes franchissements que le précédent mais les marquages de départ et de fin sont différents : $A \otimes M, t0, t1, t2, t3, t4, tf \vdash Z \otimes M$. La présence du jeton en M supprime la contrainte de causalité entre t1 et t4, ce que traduit la durée obtenue qui est $d0 + \max(d3 + d4, d1 + d2) + df$.

Un autre scénario est intéressant : $A \otimes E, t_0, t_1, t_2, t_3, t_4, t_f \vdash Z \otimes E$. La présence d'un jeton en E détruit le lien de causalité entre t_0 et t_3 . La durée est désormais égale à $\max(d_3 + d_4, d_0 + d_1 + d_4, d_0 + d_1 + d_2) + d_f$.

Bien entendu, cette méthode est aussi capable de traiter des scénarios cycliques : la durée de $A, t_0, t_1, t_2, t_3, t_4, t_f, t_c \vdash A$ est $d_0 + \max(d_3 + d_4, d_1 + d_4, d_1 + d_2) + d_f + d_c$.

6.4.6. Exemple de scénario incomplètement spécifié

Nous n'allons pas ici développer complètement le cas des scénarios incomplètement spécifiés mais seulement nous intéresser à un exemple qui illustre le dédoublement des arbres de preuve. Soit le réseau de la figure 6.11 pour lequel nous allons considérer successivement un scénario sans conflit, un scénario avec conflit de transitions et un scénario comportant simultanément un conflit de jetons et un conflit de transitions. Dans le premier cas l'arbre de preuve sera unique tandis que le second cas conduira à deux arbres et le dernier cas à quatre arbres. La présence de plusieurs arbres indique que le comportement temporel n'est pas déterministe même si le maximum de leurs durées fournit une borne supérieure.

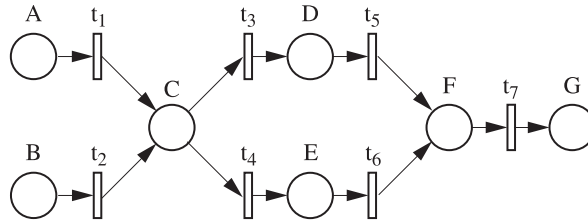


Figure 6.11. Scénarios incomplètement spécifiés

Le scénario $B \otimes C, t_2, t_3, t_3, t_5 \vdash D \otimes F$ est sans conflit : bien que certaines places contiennent plusieurs jetons, ceux-ci sont temporellement ordonnés et il n'y a pas de conflit de transition car t_4 n'est pas franchie. La durée de ce scénario est alors $\max(d_2 + d_3, d_3 + d_5)$.

Le scénario $B \otimes E, t_2, t_3, t_5, t_6, t_7 \vdash F \otimes G$ comporte un conflit de jetons en F mais ne contient aucun conflit de transitions. Les deux arbres ont respectivement pour durée $\max(d_2 + d_3 + d_5, d_6 + d_7)$ et $\max(d_2 + d_3 + d_5 + d_7, d_6)$.

Enfin, le scénario $A \otimes C, t_1, t_3, t_4, t_5, t_6, t_7 \vdash F \otimes G$ comporte un conflit de jetons en F et un conflit de transitions. Les quatre arbres qui en découlent ont pour durées respectives : $\max(d_3 + d_5, d_1 + d_4 + d_6 + d_7)$, $\max(d_3 + d_5 + d_7, d_1 + d_4 + d_6)$, $\max(d_1 + d_3 + d_5, d_4 + d_6 + d_7)$ et $\max(d_1 + d_3 + d_5 + d_7, d_4 + d_6)$.

6.4.7. Conclusion

La démarche proposée dans ce chapitre s'inscrit dans la complémentarité par rapport aux méthodes existantes pour l'analyse des systèmes représentés par réseaux de Petri : elle fournit des outils orientés événements mais ne prétend pas pour autant remplacer celles qui sont orientées états. La recherche d'une propriété inévitable n'est pas envisageable en logique linéaire.

Au vu des résultats illustrés tout au long de ce chapitre, la logique linéaire s'avère être un outil intéressant pour les réseaux de Petri : l'originalité des notions exprimées (ressources, exemplaires multiples de ressources, production, consommation) permet de modéliser le fonctionnement d'un réseau et notamment de trancher simplement les questions d'accessibilité sans construire complètement le graphe d'états. Elle permet de travailler directement à partir de la description du réseau.

Le travail initial de Gehlot [GEH 92] a permis de vérifier sa parfaite adéquation avec la structure d'un réseau mais pas avec son fonctionnement. L'accessibilité vérifiée est seulement structurelle mais il n'est pas nécessaire de connaître le multi-ensemble des transitions avant la preuve : il est obtenu par le calcul des séquents, avec cependant le risque d'une mauvaise utilisation de la règle de coupure. Les franchissements multiples ne pourront pas être clairement traités. Le but visé concernant le parallélisme est intéressant (exprimer le parallélisme par un ordre partiel série-parallèle sur le franchissement des transitions) mais il est trop restrictif : beaucoup de relations de causalité ne peuvent se traduire ainsi.

L'approche qui représente dans un « séquent scénario » à la fois des marquages et des franchissements de transitions apporte une meilleure solution à ce problème. Les résultats sur l'accessibilité permettent de compléter les outils existants : ainsi on peut imaginer que la preuve d'un séquent scénario ne soit entreprise qu'après vérification que le multi-ensemble satisfait l'équation caractéristique puisque celle-ci est une condition nécessaire mais non suffisante. La taille de la preuve est linéaire par rapport au nombre de franchissements sans perdre pour cela la possibilité de caractériser le parallélisme : l'un des buts annoncés, à savoir disposer d'un outil gérant le parallélisme sans entrelacement, est atteint. Les labels utilisés sont symboliques alors que dans une approche graphe d'états, on travaille en général avec des valeurs réelles : le graphe doit être reconstruit en cas de changement de valeurs.

Pour parvenir à ce résultat, exprimer le comportement et pas seulement la structure, un changement important est intervenu dans les formules de logique manipulées : l'arbre de preuve contient des étapes de raisonnement et non pas des marquages. La différence est essentielle : alors qu'un marquage est une vision instantanée de l'état d'un système, une étape est une collection d'informations dont les étiquettes temporelles sont différentes. C'est ce point qui donne toute sa force à la méthode et qui, malgré les apparentes similitudes, constitue une différence fondamentale avec le joueur de réseau de Petri.

Si les scénarios complètement spécifiés sont traités très facilement, il faut encore développer les cas pour lesquels existent des conflits de jetons et de transitions, pour être certain de générer tous les arbres de preuve correspondants. Ce point n'est pas complètement trivial pour les conflits de jetons notamment, car il a des conséquences sur l'ordre d'application des règles : pour faire apparaître des conflits, on sera amené à retarder le traitement de certains franchissements. Le grand avantage de la logique linéaire est qu'elle traite des réseaux non saufs, cycliques, etc. De plus, elle dispose de connecteurs supplémentaires que nous n'avons pas abordé ici : les connecteurs additifs qui expriment des choix internes ou externes peuvent sans doute aider au traitement des choix et des conflits.

6.5. Bibliographie

- [DIA 01] DIAZ M. (dir.), *Les réseaux de Petri*, Hermès, Traité IC2, Paris, 2001.
- [GEH 92] GEHLOT V., A proof theoretic approach to semantics of concurrency, PhD Thesis, Université de Pennsylvanie, 1992.
- [GIRAR 87] GIRARD J.Y., *Linear Logic*, Theoretical Computer Science, vol. 50, Pays-Bas, 1987.
- [GIRAR 95a] GIRARD J.Y., « Linear Logic: a survey », dans *The Curry-Howard Isomorphism, Cahiers du Centre de Logique*, n° 8, Academia Press, 1995.
- [GIRAR 95b] GIRARD J.Y., *Linear Logic: its syntax and semantics*, Advances in Linear Logic, London Mathematical Society, Lecture Notes Series, n° 222, Cambridge University Press, 1995.
- [GIRAU 97a] GIRAULT F., Formalisation en logique linéaire du fonctionnement des réseaux de Petri, Thèse de doctorat, n° 2870, Université Paul Sabatier, Toulouse, France, 1997.
- [GIRAU 97b] GIRAULT F., PRADIN-CHEZALVIEL B., VALETTE R., « A logic for Petri nets », *JESA* vol. 31, n° 3, Hermès, 1997.
- [GUN 89] GUNTER G., GEHLOT V., « Nets as tensor theory », dans *10th International Conference on Applications and Theory of Petri Nets*, Bonn, Germany, 1989.
- [KUN 97] KUNZLE L.A., Raisonement temporel basé sur les réseaux de Petri pour des systèmes manipulant des ressources, Thèse de doctorat, n° 2753, Université Paul Sabatier, Toulouse, France, 1997.
- [PRA 99a] PRADIN-CHEZALVIEL B., KUNZLE L.A., GIRAULT F., VALETTE R., « Evaluation temporelle de scénarios de réseaux de Petri incluant du parallélisme », *MSR'99 (Modélisation des Systèmes Réactifs)*, Cachan, France, 24-26 mars 1999.
- [PRA 99b] PRADIN-CHEZALVIEL B., VALETTE R., KUNZLE L.A., « Scenario durations characterization of t-timed Petri nets using linear logic », *PNPM'99 (Petri Nets Performance Modeling)*, Zaragoza, Espagne, 8-10 septembre 1999.
- [SIL 90] SILVA M., VALETTE R., « Petri nets and flexible manufacturing », *Advances in Petri nets 1989*, LNCS 424, p. 374-417, Springer-Verlag, 1990.
- [VAL 82] VALDES J., TARJAN R.E., LAWLER E.L., « The recognition of Series-Parallel digraphs », *SIAM Journal of Computing*, vol. 11, n° 2, 1982.