

# *Projet PARAGRAPHÉ*

*Parallélisme et graphes*

*Rennes*

THÈME 1C



*Rapport  
d'Activité*

1999



## Table des matières

<b>1</b>	<b>Composition de l'équipe</b>	<b>3</b>
<b>2</b>	<b>Présentation et objectifs généraux</b>	<b>3</b>
<b>3</b>	<b>Fondements scientifiques</b>	<b>4</b>
3.1	Théorie des régions et synthèse des réseaux de Petri . . . . .	4
3.1.1	Un aperçu des réseaux de Petri classiques . . . . .	4
3.1.2	Théorie des régions . . . . .	5
3.2	Réseaux de Petri étendus . . . . .	7
3.3	Graphes reconnaissables . . . . .	8
3.3.1	Graphes reconnaissables et bisimulations . . . . .	8
3.3.2	Domaines d'événements reconnaissables . . . . .	9
<b>4</b>	<b>Domaines d'applications</b>	<b>11</b>
4.1	Panorama . . . . .	11
4.1.1	Distribution semi-automatique d'automates . . . . .	11
4.1.2	Synthèse de contrôleurs asynchrones . . . . .	12
4.1.3	Modélisation et simulation de services coopératifs . . . . .	12
<b>5</b>	<b>Logiciels</b>	<b>12</b>
<b>6</b>	<b>Résultats nouveaux</b>	<b>12</b>
6.1	Graphes de marquages, théorie des régions, synthèse . . . . .	12
6.1.1	Synthèse des réseaux de Petri non bornés . . . . .	13
6.1.2	Graphes de marquages de systèmes d'addition de vecteurs . . . . .	14
6.2	Réseaux de Petri étendus . . . . .	14
6.2.1	Réseaux reconfigurables réversibles . . . . .	14
6.2.2	Automates coopérants . . . . .	15
6.2.3	Réseaux et programmes synchrones . . . . .	16
6.3	Réseaux de Petri, logique linéaire . . . . .	17
6.3.1	Réseaux de Petri et logique linéaire . . . . .	17
6.3.2	Grammaires minimalistes et logique linéaire . . . . .	18
6.4	Graphes reconnaissables et dépliage . . . . .	19
6.4.1	Graphes réguliers, reconnaissables, rationnels . . . . .	19
6.4.2	Une théorie du dépliage . . . . .	20
<b>7</b>	<b>Actions régionales, nationales et internationales</b>	<b>21</b>
7.1	Actions régionales . . . . .	21
7.1.1	Application des réseaux de Petri aux services coopératifs . . . . .	21
7.2	Actions nationales . . . . .	21
7.2.1	Synthèse de commande des systèmes à événements discrets à l'aide des réseaux de Petri . . . . .	21
7.3	Actions européennes . . . . .	21

7.4	Actions internationales . . . . .	21
7.5	Visites, et invitations de chercheurs . . . . .	21
<b>8</b>	<b>Diffusion de résultats</b>	<b>22</b>
8.1	Animation de la Communauté scientifique . . . . .	22
8.2	Enseignement . . . . .	22
8.3	Rapports de thèse . . . . .	22
8.4	Séminaires . . . . .	22
<b>9</b>	<b>Bibliographie</b>	<b>23</b>

## 1 Composition de l'équipe

### Responsable scientifique

Philippe Darondeau [DR Inria]

### Personnel Inria

Eric Badouel [CR - en détachement à partir du 1/09/99]

Boubakar Gamatié [CR]

Christian Retoré [CR - intégré au projet le 1/09/98]

### Personnel CNRS

Didier Caucal [CR]

### Personnel INSA

Danielle Quichaud [maître de conférences]

### Personnel université de Rennes 1

Gilles Lesventes [maître de conférences - intégré au projet le 1/09/99]

Vincent Schmitt [Ater - présent jusque au 1/06/99]

### Chercheurs doctorants

David Harlet [allocataire MENRT]

Christophe Morvan [allocataire MENRT]

Yannick Le Nir [allocataire MENRT à partir du 1/10/99]

### Chercheurs post-doctorants

Andrew Tokmakoff [boursier CIES du 1/10/98 au 31/07/99]

## 2 Présentation et objectifs généraux

Le projet étudie des algorithmes de synthèse de réseaux de Petri applicables à la parallélisation, à la distribution ou au contrôle des systèmes de transitions. L'objectif est de mettre à la disposition des ingénieurs qui ont à traiter ces problèmes des outils de résolution reposant implicitement sur l'algèbre linéaire, qui est au coeur des réseaux de Petri et qui fournit souvent des solutions effectives échappant à l'intuition. Le projet étudie des réseaux de Petri étendus servant à modéliser la coordination d'agents mobiles ou humains, coopérant à un service commun ou communiquant par un système sophistiqué. L'objectif est de faciliter le développement des systèmes coopératifs en vérifiant leur bon fonctionnement dès la spécification. Le projet étudie les propriétés fondamentales des graphes à structure régulière, tels les graphes de transitions des automates à pile ou les structures d'événements associées aux réseaux de Petri saufs. L'objectif est de délimiter les familles de graphes ou de systèmes de transitions dont les propriétés logiques sont décidables, donc susceptibles d'une vérification automatisée.

### 3 Fondements scientifiques

#### 3.1 Théorie des régions et synthèse des réseaux de Petri

**Mots clés** : réseau de Petri, régions, synthèse.

**Glossaire** :

**synthese** Le problème de synthèse des réseaux de Petri consiste, étant donné un système de transitions initialisé, à construire un réseau de Petri dont le graphe des marquages accessibles soit isomorphe à ce système de transitions.

**region** Les régions d'un système de transitions sont les morphismes qui l'envoient dans le graphe de Cayley des entiers, restreint aux noeuds positifs; ces régions sont interprétées comme des places dans les réseaux synthétisés.

**Résumé** : *On trouvera ci-dessous une introduction aux réseaux de Petri avec quelques renvois à la littérature, puis une présentation de la théorie des régions d'Ehrenfeucht et Rozenberg, étendue ici à des types arbitraires de réseaux.*

##### 3.1.1 Un aperçu des réseaux de Petri classiques

On consultera Murata <sup>[Mur89]</sup> pour une présentation d'ensemble du domaine. Les réseaux de Petri servent à la spécification, à la simulation, à l'analyse, et à la mise en oeuvre de systèmes, destinés à la conduite de dispositifs plus ou moins complexes (ateliers de production, trafic aérien,...) ou à la coordination d'agents qui communiquent selon des schémas formels (protocoles de communication) ou informels (processus coopératifs). Les réseaux sont à l'origine des graphes bipartis reliant par des arcs dirigés, places à valeurs booléennes et transitions, la dynamique étant définie par un jeu de jetons, représentant les conditions valides: les places en entrée (resp. en sortie) d'une transition sont ses pre-conditions (resp. post-conditions). Il s'est par la suite avéré utile de permettre l'accumulation des jetons dans les places, vues dès lors comme de simples réservoirs de ressources banalisées. Afin de faciliter la notation de ces réseaux étendus, des poids entiers ont été inscrits sur les arcs, mesurant les flots de ressources correspondants. Le modèle obtenu est le modèle classique des réseaux de Petri, dans lesquels les places prennent des valeurs entières.

Les travaux sur les réseaux de Petri classiques ont surtout été consacrés à l'étude de leurs graphes de marquages. Ces travaux reposent sur l'algèbre linéaire et les invariants <sup>[MR80]</sup> <sup>[STC99]</sup>), sur la théorie des graphes et les symétries <sup>[HITZ95]</sup>, ou sur le mélange de ces deux

- 
- [Mur89] T. MURATA, « Petri Nets: Properties, Analysis and Applications », *Proc. IEEE* 77(4), 1989.
  - [MR80] G. MEMMI, G. ROUCAIROL, « Linear algebra in net theory », *in: Proc. Net Theory and Applications*, 84, Springer Verlag LNCS, 1980.
  - [STC99] M. SILVA, E. TERUEL, J. COLOM, « Linear Algebraic and Linear Programming Techniques for the Analysis of Place/Transition Net Systems », *in: Lectures on Petri Nets I: Basic Models*, 1491, Springer-Verlag LNCS, 1999, p. 309–373.
  - [HITZ95] S. HADDAD, J. ILIÉ, M. TAGHELIT, B. ZOUARI, « Symbolic reachability graph and partial symmetries », *in: Proc. ICATPN*, 935, Springer Verlag LNCS, 1995.

disciplines (synthèse de réseaux). Un problème central est l'obtention de méthodes de vérification capables de contourner les obstacles posés par la taille des graphes de marquages. Entre autres problèmes de décision célèbres résolus pour les réseaux de Petri, citons la finitude du graphe des marquages accessibles [KM69], l'accessibilité des marquages [May84], et la vivacité en corollaire de l'accessibilité.

La théorie des réseaux de Petri est liée à la théorie classique des domaines par les travaux sur les structures d'événements [NPW81] [HKT96]. Ces travaux donnent aux réseaux un éclairage complémentaire en leur associant des graphes acycliques ou des langages partiels comme les langages de traces [HKT92]. L'un des problèmes qui se posent dans ce cadre est de caractériser les familles de graphes ou de langages correspondant à une classe de réseaux donnée. La théorie des régions, présentée de façon assez détaillée dans [13], est l'un des éléments de réponse à ce problème.

### 3.1.2 Théorie des régions

Une présentation uniforme de toutes les familles existantes de réseaux de Petri peut être obtenue en posant les définitions suivantes. Un *type de réseaux* est un système de transitions  $\tau = (LS, LE, \tau)$  dans lequel  $LS$  et  $LE$  sont deux ensembles finis ou infinis, et  $\tau \subseteq LS \times LE \times LS$  est un ensemble de transitions  $ls \xrightarrow{le} ls'$  satisfaisant la condition de déterminisme:  $ls \xrightarrow{le} ls' \wedge ls \xrightarrow{le} ls'' \Rightarrow ls' = ls''$ . Intuitivement, les ensembles  $LS$  et  $LE$  définissent respectivement les états locaux des agents atomiques d'un système et les changements d'états de ces agents sous l'effet d'événements locaux issus de la décomposition des interactions globales. Un *réseau* marqué de type  $\tau$  est une structure  $N = (P, E, W, M)$ , où  $P$  est un ensemble de *places* figurant les agents d'un système dans les états locaux donnés par la fonction de *marquage*  $M : P \rightarrow LS$ , et  $E$  est un ensemble d' *événements* représentant des interactions globales qui se décomposent en événements locaux conformément aux colonnes de la matrice  $W : P \times E \rightarrow LE$  (*matrice des poids*). Un événement  $e \in E$  est *tirable* dans le marquage  $M$ , et sa mise à feu conduit à un marquage  $M'$  si et seulement si, pour toute place  $p \in P$ ,  $M(p) \xrightarrow{W(p,e)} M'(p)$  dans  $\tau$ . Les transitions d'un réseau s'obtiennent donc en synchronisant les transitions locales de ses sous-réseaux atomiques.

Fixons un type de réseaux  $\tau = (LS, LE, \tau)$ . Soit  $A = (S, E, T, s_0)$  un automate fini ou infini dont l'ensemble des transitions  $T \subseteq S \times E \times S$  satisfait la condition de déterminisme, et

- 
- [KM69] R. KARP, R. MILLER, « Parallel Program Schemata », *JCSS* 3, 1969.
  - [May84] E. MAYR, « An algorithm for the general Petri net reachability problem », *Siam J. Comput.* 13, 1984.
  - [NPW81] M. NIELSEN, G. PLOTKIN, G. WINSKEL, « Petri nets, event structures and domains », *TCS* 13, 1981.
  - [HKT96] P. HOOGERS, H. KLEIJN, P. THIAGARAJAN, « An event structure semantics for general Petri Nets », *TCS* 153, 1996.
  - [HKT92] P. HOOGERS, H. KLEIJN, P. THIAGARAJAN, « A trace semantics for Petri nets », *in: Proc. ICALP*, 623, Springer Verlag LNCS, 1992.

dont les états sont tous accessibles à partir de  $s_0$  (l'état initial). Une *région* <sup>[ER90a]</sup> dans  $A$  vis à vis du type  $\tau$  est un morphisme de systèmes de transitions  $(\sigma, \eta) : (S, E, T) \rightarrow (LS, LE, \tau)$ , c'est à dire une paire de fonctions  $\sigma : S \rightarrow LS$  et  $\eta : E \rightarrow LE$  envoyant toute transition  $s \xrightarrow{e} s'$  de  $T$  sur une transition  $\sigma(s) \xrightarrow{\eta(e)} \sigma(s')$  de  $\tau$ . Si  $A$  est le graphe de marquages d'un réseau  $N = (P, E, W, M_0)$  (notation:  $A = N^*$ ), on voit que chaque place  $p \in P$  détermine une région associée, donnée par  $\sigma_p(M) = M(p)$  et  $\eta_p(e) = W(p, e)$  (pour tout marquage  $M$  et pour tout événement  $e \in E$ ). Réciproquement, on peut associer à l'automate  $A = (S, E, T, s_0)$  le réseau  $A^* = (P, E, W, M_0)$  défini de la façon suivante: l'ensemble  $P$  des places du réseau est l'ensemble des régions du système de transitions  $(S, E, T)$ , et pour toute région  $p = (\sigma, \eta)$  on pose  $W(p, e) = \eta(e)$  et  $M_0(p) = \sigma(s_0)$ . On peut montrer que le graphe des marquages  $A^{**}$  du réseau  $A^*$  est isomorphe à l'automate  $A$  si et seulement si les deux conditions de *séparation* suivantes sont vérifiées <sup>[ER90b]</sup>:

- i) pour toute paire d'états distincts  $s, s' \in S$ , il existe dans  $A$  une région *séparatrice*  $(\sigma, \eta)$  telle que  $\sigma(s) \neq \sigma(s')$
- ii) pour tout état  $s \in S$  et pour tout événement  $e \in E$  non tirable en  $s$ , il existe dans  $A$  une région *inhibitrice*  $(\sigma, \eta)$  telle qu'aucune transition de la forme  $s \xrightarrow{\eta(e)} s'$  ne soit tirable dans  $\tau$ .

Lorsque  $A$  est un automate fini, le nombre des instances des deux conditions de séparation à vérifier dans  $A$  pour assurer l'isomorphisme de  $A$  et du graphe des marquages accessibles  $A^{**}$  de son réseau dual  $A^*$  a une borne quadratique en la taille de cet automate <sup>[DR96]</sup>. On en tire pour tout type  $\tau$  fini une procédure de décision du problème  $A \cong A^{**}$ , mais la décision demande en général un temps exponentiel en la taille de l'automate: pour le type des réseaux élémentaires, qui est un type fini, chaque instance de l'une des deux conditions de séparation peut poser un problème NP-complet, et le problème  $A \cong A^{**}$  est de fait NP-complet [2]. Par contre, lorsque  $A$  est isomorphe au graphe des marquages de  $A^*$ , il est toujours isomorphe au graphe des marquages d'un sous-réseau de  $A^*$  dont l'ensemble de places (obtenu en restreignant l'ensemble des régions de  $A$  à un sous ensemble irrédundant vis à vis des conditions de séparation) est de taille polynomiale en la taille de  $A$ . Pour le type des réseaux de Petri purs, donné par le graphe de Cayley des entiers restreint à ses noeuds positifs, soit  $\tau = (\mathbb{N}, \mathcal{Z}, \rightarrow)$  avec  $n \xrightarrow{z} n' \iff n + z \geq 0 \wedge n' = n + z$ , la décision du problème  $A \cong A^{**}$  prend un temps polynomial en la taille de  $A$  [1]. La synthèse d'un réseau  $N$  tel que  $A \cong N^*$  est possible si et seulement si  $A \cong A^{**}$ . Le calcul d'un réseau  $N$  irrédundant prend alors un temps polynomial. Le principe de cette synthèse est la résolution de systèmes d'inéquations linéaires *homogènes* en nombres entiers dans des modules sur  $\mathcal{Z}$  de dimensions finies, représentant toutes les composantes  $\eta$  des régions entières de  $A$ , ou morphismes de  $A$  dans  $\tau = (\mathbb{N}, \mathcal{Z}, \rightarrow)$ .

- 
- [ER90a] A. EHRENFUCHT, G. ROZENBERG, « Partial (Set) 2-Structures; Part 1: Basic Notions and the Representation Problem », *Acta Informatica* 27, 1990.
  - [ER90b] A. EHRENFUCHT, G. ROZENBERG, « Partial (Set) 2-Structures; Part 2: State Spaces of Concurrent Systems », *Acta Informatica* 27, 1990.
  - [DR96] J. DESEL, W. REISIG, « The Synthesis Problem of Petri Nets », *Acta Informatica* 33, 1996.

### 3.2 Réseaux de Petri étendus

**Mots clés :** réseau automodifiant, réseau zero-sauf.

**Glossaire :**

**réseau automodifiant** réseau de Petri dans lequel les poids des arcs sont des combinaisons linéaires entières des variables associées aux places (représentant le contenu de ces places dans le marquage courant).

**réseau zero-sauf** réseau de Petri dans lequel on distingue un sous-ensemble de places (les zero-places) dont la vacuité conditionne l'observabilité des marquages.

**Résumé :** *On trouvera ci-dessous une introduction aux réseaux de haut niveau et aux réseaux à structure dynamique, avec quelques renvois à la littérature.*

Le modèle des réseaux de Petri a reçu vers la fin des années 70 les aménagements nécessaires à son utilisation dans des applications complexes. L'aménagement le plus important a été la distinction de couleurs de jetons, représentant des données [Jen92] [Vau86]. Les réseaux colorés ne sont qu'une forme comprimée de réseaux de Petri ordinaires, dans lesquels on peut toujours les traduire, mais ils combinent utilement flots de données et contrôle de ressources (le contenu d'une place est un multi-ensemble de données). Les réseaux algébriques sont une classe très générale de réseaux colorés dans lesquels les places sont typées par des types algébriques, définissant des ensembles de données (ou de couleurs) et les fonctions sur ces ensembles par lesquels opèrent les transitions afin de transformer leurs jetons d'entrée en jetons de sortie [Vau87] [Rei91]. Plus loin dans la même voie ont été proposés divers modèles de réseaux orientés objets, faisant intervenir la création dynamique de sous réseaux. Citons par exemple les réseaux de Sibertin-Blanc [SB94], dont la structure évolue au cours de l'exécution par création dynamique de sous-réseaux mobiles figurés par des jetons. Un dynamisme plus limité apparaît dans les réseaux automodifiants de Valk [Val78]. Dans ces réseaux, les ensembles de places et de transitions sont fixes, mais les relations de flot liant les unes aux autres varient en fonction des marquages, car les arcs sont pondérés par des combinaisons linéaires entières de variables, associées aux places et représentant leur contenu dans le marquage courant. Les propriétés des réseaux colorés sont semblables à celles des réseaux de Petri lorsque les ensembles de couleurs sont finis, mais doivent être réanalysées dans le cas contraire. Les propriétés des réseaux à objets et des réseaux automodifiants sont moins bien connues. Ces formalismes ont en général la puissance des machines de Turing, et le problème est d'en déterminer des sous-ensembles bien taillés

- 
- [Jen92] K. JENSEN, *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, Volume 1, Basic Concepts*, EATCS-Monographs, Springer Verlag, 1992.
- [Vau86] J. VAUTHERIN, « Non-linear invariants for coloured Petri nets with interdependent tokens; Application to the proof of parallel programs », *in: Advances in Petri nets 1985*, 222, Springer Verlag LNCS, 1986.
- [Vau87] J. VAUTHERIN, « Parallel systems specification with coloured Petri nets and algebraic abstract data types », *in: Proc. 7th European Workshop on ATPN*, 266, Springer Verlag LNCS, 1987.
- [Rei91] W. REISIG, « Petri nets and algebraic specifications », *TCS 80*, 1991.
- [SB94] C. SIBERTIN-BLANC, « Cooperative nets », *in: Proc. ICATPN*, 815, Springer Verlag LNCS, 1994.
- [Val78] R. VALK, « Self-modifying nets, a natural extension of Petri nets », *in: Proc. ICALP*, 62, Springer Verlag LNCS, 1978.

ayant au moins quelques propriétés décidables. La finitude du graphe des marquages est ainsi décidable pour les réseaux automodifiants dans lesquels les poids qui figurent sur les arcs en entrée des transitions sont des poids ordinaires, c'est à dire des constantes, indécidable dans le cas général.

### 3.3 Graphes reconnaissables

**Mots clés :** reconnaissabilité, décidabilité, dépliage.

**Glossaire :**

**reconnaissabilité** propriété, pour une structure discrète, d'être finie ou d'être engendrée par des transformations itérées à partir d'une structure finie.

**decidabilité** une théorie est décidable s'il existe une machine qui l'énumère et une autre qui énumère son complément.

**depliage** opération produisant à partir d'une structure donnée une structure acyclique équivalente.

**Résumé :** *Selon la définition classique, un langage est reconnaissable en tant que partie d'un monoïde s'il est l'image inverse d'une partie d'un monoïde fini par un morphisme de monoïdes. Dans le cas d'un monoïde libre, les langages reconnaissables coïncident avec les langages réguliers et aussi avec les langages engendrés par les grammaires linéaires droites. En s'inspirant de ces notions classiques, on essaie de définir une notion plus générale de reconnaissabilité pour des structures discrètes qui ne sont plus des langages mais des graphes. Deux approches sont possibles. L'une des approches consiste, en partant d'un ensemble d'objets (finis ou infinis) dans lesquels on a choisi un axiome ou un sous-ensemble d'axiomes, à équiper ces objets d'un ensemble de transformations et à définir comme objets reconnaissables les objets qui peuvent être obtenus par des suites finies de transformations appliquées à l'un des axiomes. La seconde approche, un peu plus fidèle à la définition classique, part de la notion de morphisme de pliage dans une catégorie d'objets finis ou infinis, et définit comme reconnaissables les objets qui se plient sur des objets finis.*

#### 3.3.1 Graphes reconnaissables et bisimulations

Les systèmes de récritures de graphes, en particulier les grammaires déterministes de graphes [Cou90], sont des représentations finies et exécutables de graphes infinis. Ces représentations servent à la décision d'une vaste famille de propriétés des graphes décrits: toute propriété définissable au second ordre monadique dans le langage d'un tel graphe est décidable. Ceci ne donne pas de moyen direct pour vérifier les propriétés d'un système ou d'un programme concurrent. Il faudrait encore, toute considération d'efficacité mise à part, savoir produire à partir d'un programme concurrent un système de récritures ou une grammaire qui engendre un graphe isomorphe au graphe des transitions de ce programme. Cette transformation n'est

---

[Cou90] B. COURCELLE, *Graph rewriting: an algebraic and logic approach*, Elsevier, 1990.

actuellement possible qu'à partir d'automates à pile <sup>[MS85]</sup> ou de systèmes de récritures préfixes sur les mots [6] ou les termes [5]. Dans le cas très particulier de programmes concurrents se comportant comme des automates à pile, on sait aussi décider si deux programmes ont le même comportement (c'est à dire si deux automates à pile ont des graphes de transitions isomorphes). Une équivalence de comportements moins rigoureuse que l'isomorphisme de graphes est la bisimulation, ou isomorphisme de graphes à l'identification près des sous-graphes accessibles isomorphes. On sait là encore, dans le cas de processus dont les transitions sont décrites par des récritures préfixes selon des grammaires algébriques sur les mots, décider de la bisimilarité de leurs comportements en passant par des grammaires de graphes décrivant ces comportements (elles même produites à partir des grammaires sur les mots).

Les graphes engendrés par les grammaires déterministes de graphes sont les graphes *réguliers* de Courcelle <sup>[Cou90]</sup>. Ces graphes sont un sous-ensemble strict des graphes *reconnaissables* de Caucal [8]. Ils peuvent en effet être obtenus par des transformations appliquées à l'arbre binaire complet  $B$ , compatibles avec la logique du second ordre monadique: à toute transformation  $T$  de cette famille correspond une transformation effective  $T^*$  sur les formules, telle que pour toute formule  $\phi$  du langage du graphe  $T(B)$ ,  $T^*(\phi)$  soit une formule du langage de l'arbre binaire complet et que  $T(B) \models \phi$  si et seulement si  $B \models T^*(\phi)$ . L'arbre binaire complet ayant, par un théorème dû à Rabin, une théorie du second ordre monadique décidable <sup>[Rab69]</sup>, l'ensemble des graphes reconnaissables hérite de cette propriété fondamentale.

La bisimilarité de deux sous-graphes accessibles d'un même graphe reconnaissable ne pouvant pas être exprimée par une formule du second ordre monadique, la propriété précédente n'est d'aucune aide pour résoudre le problème de décision de la bisimilarité des graphes reconnaissables, qui est un problème ouvert. La restriction de ce problème aux graphes réguliers est aussi un problème ouvert. Par contre, la bisimilarité est décidable pour le sous-ensemble des graphes réguliers à étiquetage déterministe (deux arcs issus d'un même sommet portent des étiquettes distinctes). En effet, la bisimilarité des graphes réguliers à étiquetage déterministe et l'équivalence des automates à pile déterministes sont deux problèmes interréductibles [7], et l'équivalence des automates à pile déterministes est décidable <sup>[Sén97]</sup>.

### 3.3.2 Domaines d'événements reconnaissables

Un automate concurrent est un automate déterministe, fini ou infini, muni en chaque état d'une relation binaire, symétrique et irréflexive, sur le sous-ensemble des événements tirables en cet état, appelée relation d'*indépendance*. La famille des relations d'indépendance, indexée par les états, sert à définir une équivalence sur les suites d'événements acceptées par l'automate,

- 
- [MS85] D. MÜLLER, P. SCHUPP, « The theory of ends, pushdown automata, and second order logic », *TCS* 37, 1985.
- [Rab69] M. RABIN, « Decidability of second-order theories and automata on infinite trees », *Trans. AMS* 141, 1969.
- [Sén97] G. SÉNIZERGUES, « The equivalence problem for deterministic pushdown automata is decidable », in : *Proc. ICALP*, 1256, Springer Verlag LNCS, 1997.

fondée sur la non distinction de deux suites  $uabv$  et  $ubav$  qui diffèrent seulement par la permutation de deux événements  $a$  et  $b$  indépendants dans l'état auquel l'automate est conduit par l'acceptation de  $u$ . Afin que chaque classe de cette équivalence puisse être interprétée comme l'ensemble des observations séquentielles d'un processus non-séquentiel, un axiome de cohérence locale est posé sur la famille des relations d'indépendance de l'automate, à savoir l'axiome du cube <sup>[Dro90]</sup>. Le langage reconnu par un automate concurrent est alors l'ensemble des classes d'équivalence des mots acceptés par l'automate sous jacent. Un tel langage est appelé un langage de traces.

Les notions centrales de la théorie des automates, comme celles de simulation d'un automate par un autre ou de réalisation minimale d'un langage par un automate, s'étendent aux automates concurrents et aux langages de traces. La théorie des automates concurrents n'est cependant pas une simple extension de la théorie des automates classiques: on peut voir un automate concurrent comme la représentation d'un ensemble ordonné, à savoir l'ensemble de ses traces ordonnées par le quotient de l'ordre préfixe. Ces ensembles ordonnés, appelés domaines d'événements, sont des domaines de Scott. La caractérisation axiomatique et la classification des divers types de domaines d'événements dans la classe des domaines de Scott ont fait l'objet de nombreux travaux <sup>[Bed87] [Dro90] [Kus94]</sup>, récemment prolongés dans [10]. Il n'existe par contre que très peu de résultats sur la caractérisation et la classification des domaines d'événements *reconnaisables* c'est à dire représentables par des automates concurrents *finis*.

Le domaine d'événements associé à un automate concurrent peut être vu comme un *dépliage* de cet automate dans la catégorie des automates concurrents <sup>[BD94]</sup> et des morphismes de pliage <sup>[BD92]</sup>. La question qui se pose alors est de caractériser les dépliages d'automates concurrents finis. Une question un peu plus générale est de mettre en évidence une notion robuste de reconnaissabilité dont les instanciations pour divers types d'automates concurrents ou de réseaux de Petri soient compatibles. Il existe aujourd'hui peu de résultats sur ces questions qui posent des problèmes combinatoires. Quelques résultats ont été obtenus sur les étiquetages finis des structures d'événements premières <sup>[ABCR94]</sup>. On sait également que les domaines d'événements *context free* (i.e. dont la réduction transitive est engendrée par une grammaire déterministe de graphes) sont reconnus par des automates concurrents finis, en l'occurrence des automates trace [3].

- 
- [Dro90] M. DROSTE, « Concurrency, automata and domains », *in: Proc. ICALP*, 443, Springer Verlag LNCS, 1990.
  - [Bed87] M. BEDNARCZYK, *Categories of asynchronous systems*, thèse de doctorat, university of Sussex, 1987.
  - [Kus94] D. KUSKE, « Non deterministic automata with concurrency relations », *in: Proc. ICALP*, 787, Springer Verlag LNCS, 1994.
  - [BD94] F. BRACHO, M. DROSTE, « Labelled domains and automata with concurrency », *TCS 135*, 1994.
  - [BD92] E. BADOUEL, P. DARONDEAU, « Trace nets », *in: Proc. REX workshop "Semantics: Foundations and Applications"*, 666, Springer Verlag LNCS, 1992.
  - [ABCR94] M. ASSOUS, V. BOUCHITTÉ, C. CHARRETON, B. ROZOY, « Finite labelling problem in event structures », *TCS 123*, 1994.

## 4 Domaines d'applications

### 4.1 Panorama

**Résumé :** *Les recherches menées sur les réseaux de Petri au sein de Paragraphe ont plusieurs domaines d'application. Un domaine étudié en coopération avec le projet EP-ATR est la vérification et la distribution de programmes synchrones, traduits en des réseaux de Petri zero-saufs. Des éléments encourageants ont été obtenus. Un domaine étudié en coopération avec le projet Pampa est la distribution semi-automatique d'automates, incluant les protocoles de communication. Un domaine voisin, étudié dans le cadre de l'action coopérative Mars (à laquelle participe aussi le projet Pampa) est la synthèse de contrôleurs distribués. Ces deux domaines, dans lesquels des résultats probants ont été obtenus en appliquant les techniques de synthèse de réseaux à des cas d'étude encore simples mais déjà réalistes, portent nos espoirs et font l'objet de nos efforts pour le montage de coopérations internationales. Un autre domaine dans lequel nous avons pu travailler grâce au soutien de la région Bretagne est l'aide à la conception de systèmes d'agents dynamiques interactifs (communiquant à travers un service de télécommunication, ou coopérant sous le contrôle d'un système de conduite de travail).*

#### 4.1.1 Distribution semi-automatique d'automates

La conception de logiciel distribué repose aujourd'hui beaucoup sur l'expertise pour adapter et assembler selon les besoins quelques algorithmes bien connus. Allant dans une autre direction, nous cherchons à développer des méthodes semi-automatiques permettant de produire directement et sans effort de bonnes implémentations distribuées de systèmes simples comme les protocoles de communication. Synet, l'outil de synthèse de réseaux de Petri qui a été développé dans le projet Pampa, fournit un support possible pour la conception assistée de logiciel distribué. Cet outil est capable de décider à partir d'un système de transitions étiquetées dans un alphabet distribué s'il existe un réseau distribuable qui le réalise. Lorsque la décision est positive, une implémentation distribuée est produite sous la forme d'une famille d'automates (un par site) communiquant par émissions / réceptions de messages asynchrones. Lorsque la décision est négative, les raisons de l'échec sont fournies, indiquant les modifications à apporter aux spécifications. On trouvera dans <sup>[Cai99]</sup> une description de l'application de cette méthode au protocole de connexion / déconnexion.

---

[Cai99] B. CAILLAUD, « Applications des techniques de synthèse de réseaux de Petri bornés à la répartition d'automates réactifs », *in*: Actes du deuxième congrès "Modélisation des systèmes réactifs", Hermes Science Publications, 1999.

### 4.1.2 Synthèse de contrôleurs asynchrones

Une théorie du contrôle superviseur fondée sur les automates et les langages formels a été développée par Ramadge et Wonham [RW87] [WR87]. L'objectif, étant donné un système (installation) et une description de ses comportements admissibles (trajectoires viables), est de construire un superviseur qui maintienne le système dans des bornes sûres sans le restreindre de façon indue. Dans la pratique, la supervision par des machines d'états finis souffre de limitations sérieuses : l'espace d'états des contrôleurs croît exponentiellement avec le degré d'asynchronisme des installations. La construction de contrôleurs asynchrones, seuls applicables à des installations largement réparties, est par ailleurs un problème ouvert. L'action coopérative Mars de l'Inria, conduite par Xiaolan Xie (Inria-Lorraine) avec l'ENS-Cachan et l'Université de Bordeaux I, a pour objet de remédier à ces problèmes en fournissant une nouvelle méthode de synthèse de contrôleurs applicable à des installations décrites par des réseaux de Petri, assortie de vérifications des systèmes contrôlés. Les travaux menés dans ce cadre par M. Uzam (post-doctorant à l'Inria-Rennes et à l'Inria-Lorraine) montrent que la technique de synthèse des réseaux de Petri trouve ici un champ d'application privilégié.

### 4.1.3 Modélisation et simulation de services coopératifs

Lorsque on doit concevoir une application coopérative et la valider par un modèle, se pose le problème de choisir un formalisme pour décrire ce modèle. Les langages à objets, qui peuvent servir à cet effet, ne semblent adaptés ni à la simulation interactive ni à la vérification des systèmes décrits. Les réseaux de Petri classiques, appropriés à la simulation et à la vérification, ne permettent pas de décrire le contrôle de systèmes dynamiques d'agents concurrents qui coopèrent en lançant des transactions servies à l'aide de ressources partagées. Un compromis entre les réseaux de Petri et les langages à objets a été recherché dans le cadre d'une convention avec la région Bretagne, aboutissant à un cadre de description pratique et de simulation interactive de tels systèmes.

## 5 Logiciels

## 6 Résultats nouveaux

### 6.1 Graphes de marquages, théorie des régions, synthèse

**Participants** : Eric Badouel, Philippe Darondeau.

**Mots clés** : réseau de Petri, synthèse, région.

**Résumé** : *Une solution algorithmique a été apportée au problème de la synthèse des réseaux de Petri non bornés à partir de graphes infinis, donnés par des*

---

[RW87] P. RAMADGE, W. WONHAM, « Supervisory Control of Classes of Discrete Event Processes », *SIAM Journal on Control and Optimisation* 25, 1, 1987.

[WR87] W. WONHAM, P. RAMADGE, « On Supremal Controllable Sublanguage of a Given Language », *SIAM Journal on Control and Optimisation* 25, 3, 1987.

*grammaires déterministes de graphes ou par des automates à pile déterministes [22]. Ce problème est sensiblement plus difficile que le problème de la synthèse des réseaux de Petri non bornés à partir de langages d'automates à pile déterministes (traité l'année précédente [9]). Les relations entre réseaux de Petri et groupes ont par ailleurs été examinées. Une classification des représentations des automates comme sous-graphes pleins de graphes de Schreier a été produite et appliquée aux graphes de marquages des réseaux de Petri [12].*

### 6.1.1 Synthèse des réseaux de Petri non bornés

**Participant :** Philippe Darondeau.

Dans un travail antérieur, nous avons étudié la synthèse des réseaux de Petri non bornés à partir de langages réguliers ou de langages d'automates à pile déterministes [9]. Etant donné un langage fermé par préfixes, le problème était alors de décider s'il s'agit du langage d'un réseau, et si oui de construire le réseau.

Le travail mené cette année a abouti à la solution algorithmique d'un problème plus difficile : étant donné un graphe infini régulier, défini par une grammaire déterministe de graphes ou par un automate à pile déterministe, décider s'il est isomorphe au graphe des marquages accessibles d'un réseau de Petri, et si oui construire le réseau. La solution proposée dans [22] étend les méthodes utilisées dans [1] pour la synthèse des réseaux de Petri à partir de graphes finis, mais elle en diffère car la base de cycles d'un graphe régulier est généralement infinie. Elle réutilise certaines méthodes employées dans [9] pour la synthèse des réseaux à partir de langages (l'ensemble des régions  $y$  est représenté par un cône polyédral), mais elle introduit des idées neuves pour traiter le problème de la séparation des états par les régions du graphe. Ce problème constitue la difficulté principale. On parvient à la solution en combinant les techniques d'algèbre linéaire utilisées dans [1] et dans [9] et les propriétés de décision des ensembles semi-linéaires.

Ce travail sur l'extension de la synthèse à des *systèmes infinis*, comme le travail mené l'an dernier sur l'extension de la synthèse à des *spécifications incomplètes*, vise à explorer les frontières des domaines d'application dans lesquels on peut raisonnablement envisager de résoudre des problèmes non académiques en utilisant la synthèse des réseaux de Petri.

Une application significative de la synthèse de réseaux à partir de spécifications incomplètes a ainsi émergé de la réflexion menée dans l'action coopérative Mars : la synthèse à partir d'un encadrement rationnel des comportements admissibles permet de décider si le contrôle optimal visant à éviter le blocage d'un système fini peut être réalisé par un réseau de Petri (le cas échéant par un réseau distribuable).

### 6.1.2 Graphes de marquages de systèmes d'addition de vecteurs

**Participant** : Eric Badouel.

Lorsque elle s'applique à des graphes ou à des automates, la synthèse de réseaux peut se concevoir comme la représentation d'un automate par un produit synchronisé d'automates types (*atomes* de la représentation). Dans le cas des réseaux élémentaires et dans celui des systèmes d'addition de vecteurs, ces automates types sont des restrictions pleines de graphes de Cayley, respectivement ceux des groupes  $\mathbb{Z}/3\mathbb{Z}$  et  $\mathbb{Z}$ . Une classification des représentations des automates comme sous-graphes pleins de graphes de Schreier a été donnée dans [12]. Un automate ainsi représenté doit évidemment être réversible, au sens que chaque événement  $y$  définit une bijection partielle de l'ensemble des états. Les conditions que doit satisfaire l'automate vis à vis du groupe le représentant sont semblables aux propriétés de séparation définies par Ehrenfeucht et Rozenberg pour les systèmes de transitions élémentaires. On décrit aussi le calcul de la représentation canonique d'un automate commutatif, c'est à dire d'un automate qui se plonge dans le graphe de Cayley d'un groupe commutatif.

En interprétant chaque place d'un système d'addition de vecteurs comme une contrainte synchrone, on peut caractériser les graphes de marquages des systèmes d'addition de vecteurs comme les graphes réduits obtenus par réduction de graphes polyédriques. Il s'agit d'automates commutatifs et on conjecture qu'ils sont sans torsion, c'est à dire que leur représentation canonique ne contient pas d'élément d'ordre fini. En utilisant la représentation canonique des automates commutatifs on obtient un algorithme de synthèse de systèmes d'addition de vecteurs à partir de graphes finis.

## 6.2 Réseaux de Petri étendus

**Mots clés** : réseau automodifiant, travail coopératif, réseau zero-sauf.

**Participants** : Eric Badouel, Philippe Darondeau, David Harlet, Danièle Quichaud, Andrew Tokmakoff.

**Résumé** : *Avec les réseaux reconfigurables réversibles [21], nous proposons un modèle simple du changement dynamique dans les systèmes de gestion de flux de tâches, dans lequel toutes les propriétés classiques des réseaux de Petri restent décidables. Nous proposons par ailleurs, pour la description et pour la simulation interactive de systèmes coopératifs ayant un fort degré de dynamisme, un modèle d'automates coopérants intermédiaire entre les réseaux de Petri et les langages à objets. Sur un autre registre, nous poursuivons l'étude des réseaux de Petri zero-sauf et de leurs relations avec la programmation synchrone.*

### 6.2.1 Réseaux reconfigurables réversibles

**Participant** : Eric Badouel.

Les plus simples et les mieux maîtrisés des systèmes de travail coopératif sont les systèmes de gestion de flux de tâches (Workflow Management Systems), d'un usage assez répandu dans les grands services d'administration. Ces systèmes se prêtent très naturellement à la modélisation par des réseaux de Petri. En suivant van der Aalst [vdA97], on peut par exemple vérifier le bon fonctionnement d'un WMS en s'assurant que le rebouclage du réseau de Petri associé est un réseau vivant et borné. L'un des problèmes critiques des WMS est celui du *changement dynamique*, c'est à dire de la modification du système à chaud sans perturber la suite des travaux en cours [EKR95].

L'une des façons d'aborder cet aspect est de considérer des réseaux de Petri augmentés de règles de réécriture de sous-réseaux, représentant des changements de modes. E. Badouel et J. Oliver avaient montré l'an dernier que les techniques proposées par van der Aalst pour la vérification des Workflows peuvent être étendues à ces réseaux reconfigurables [BO98]. Ils ont poursuivi cette année l'étude de ces réseaux en considérant plus particulièrement la sous classe des réseaux reconfigurables *réversibles* c'est-à-dire tels que chacune de leurs règles de modification structurelle ait un inverse formel. Un réseau reconfigurable réversible peut être vu comme la composition en cascade d'un automate et d'un réseau de Petri, et équivaut sous certaines hypothèses à un réseau de Petri stratifié.

E. Badouel et J. Oliver ont montré dans [21] que toutes les propriétés classiques des réseaux de Petri (accessibilité, caractère borné d'une place, non blocage, vivacité, couverture, etc.) sont décidables pour les réseaux reconfigurables réversibles.

### 6.2.2 Automates coopérants

**Participants :** Eric Badouel, Philippe Darondeau, Danièle Quichaud, Andrew Tokmakoff.

Dans le cadre d'une convention avec la région Bretagne, E. Badouel, Ph. Darondeau, D. Quichaud et A. Tokmakoff ont travaillé à la définition d'un modèle pratique pour la description de systèmes dynamiques d'agents coopérants, et à la construction d'un outil associé de simulation interactive.

Le modèle des automates coopérants [15] [20] s'articule autour de trois notions de base : celles d'état, d'action et d'appartenance d'un agent à un groupe. Un système d'automates coopérants se présente sous la forme d'un pool d'agents et d'un ensemble de règles régissant

- 
- [vdA97] W. VAN DER AALST, « Verification of Workflow Nets », *in : Proc. ICATPN*, 1248, Springer Verlag LNCS, 1997.
- [EKR95] C. ELLIS, K. KEDDARA, G. ROZENBERG, « Dynamic Change within Workflow Systems », *in : Proc. Conference on Organizational Computing Systems*, ACM Press, 1995.
- [BO98] E. BADOUEL, J. OLIVER, « Reconfigurable Nets, A Class of High Level Petri Nets Supporting Dynamic Changes », *in : WFM'98, CSR 98/07*, Eindhoven University of Technology, Department of Mathematics and Computing Science, p. 129-145, Lisbon, 1998.

leurs interactions. Chaque agent est représenté par un automate fini, décrivant les actions qu'il peut exécuter en fonction de son état courant (requêtes ou services), et par une liste d'attributs dont les valeurs décrivent ses relations avec les autres agents. Les valeurs des attributs n'ont pas de signification absolue: seules les égalités entre ces valeurs sont significatives. Les groupes d'agents sont ainsi définis par les valeurs d'attributs partagées. Les transactions sont décrites par des règles comportant chacune une garde, un vecteur d'actions synchronisées, et un effet de bord. La garde précise les relations qui doivent exister entre les agents qui s'enrôlent dans la transaction. Le vecteur d'actions synchronisées précise le service dévolu à chacun de ces agents. L'effet de bord précise la modification des relations entre agents à l'issue de la transaction. La création et la destruction d'agents sont rendues possibles par l'introduction d'un état fictif commun à tous les automates: une action issue de cet état figure la création d'un agent, une action y conduisant figure sa destruction.

Le plein pouvoir expressif du modèle est atteint dès que les agents ont deux attributs: on a la puissance du calculable et ceci enlève toute possibilité de vérifications automatisées sur la classe générale des automates coopérants. Le nombre des attributs par agent permet de classer les automates coopérants en trois niveaux de hiérarchie. La base de la hiérarchie est constituée des systèmes qui peuvent être définis sans utiliser d'attributs. Ce premier niveau correspond à la notion de coordination commune aux automates synchronisés d'Arnold et Nivat, au formalisme Gamma et aux réseaux de Petri. Le second niveau de la hiérarchie est constitué des systèmes d'agents à un attribut. Cet attribut sert à matérialiser l'appartenance d'un agent à un groupe, ce qui permet entre autres de représenter les dépendances logiques entre tâches dans les systèmes de gestion de flux de tâches. Les systèmes décrits restent susceptibles de certaines vérifications: une variante de la construction de l'arbre de couverture permet par exemple de vérifier s'ils sont bornés. Le troisième niveau de la hiérarchie est constitué des systèmes d'automates coopérants à deux attributs ou plus. On ne peut espérer vérifier ces systèmes par des procédures automatisées, à moins que le diamètre des relations entre agents n'y reste borné. Nous pensons que tel est bien le cas dans les systèmes d'automates coopérants qui modélisent des services coopératifs.

Un outil de simulation des systèmes d'automates coopérants a été écrit en Java. Le simulateur prend en entrée deux fichiers de texte, décrivant respectivement les automates, et les vecteurs d'actions synchronisées avec leurs gardes et leurs effets de bord. Le simulateur fournit deux fenêtres interactives: l'une montre la configuration courante du système (liste des agents représentés par leur état et leurs attributs), l'autre indique les vecteurs d'actions synchronisées tirables dans l'état courant. L'utilisateur peut sélectionner un à un les agents qu'il veut activer: la liste des vecteurs de synchronisation tirables est réduite au fur et à mesure. L'utilisateur peut choisir de tirer l'un de ces vecteurs. Le simulateur affiche alors la liste des agents restant à choisir pour compléter la transaction. Les configurations sont visualisées sous forme de graphes. Une configuration antérieure peut être restaurée afin d'explorer de nouveaux choix.

### 6.2.3 Réseaux et programmes synchrones

**Participants :** Philippe Darondeau, David Harlet.

Dans le cadre d'une action concertée des projets EP-ATR et Paragraphe, nous étudions la traduction des programmes du langage Signal en des réseaux de Petri équivalents. L'un des buts poursuivis est d'appliquer à Signal les outils de vérification de réseaux de Petri. Il importe que les réseaux de Petri considérés soient des réseaux ordinaires. Sans modifier la règle d'évolution des réseaux, il reste cependant possible de considérer que certains marquages sont inobservables. En s'inspirant des réseaux zero-saufs de Bruni et Montanari <sup>[BM97]</sup>, nous utilisons des réseaux de Petri dans lesquels nous distinguons un sous-ensemble de places, appelées zero-places, dont la vacuité conditionne l'observabilité des marquages. Un pas d'exécution est alors une suite quelconque de transitions menant d'un marquage observable à un autre. La différence avec les réseaux zero-saufs est que tout jeton peut ici être consommé dans le pas d'exécution au cours duquel il a été produit, afin que la désynchronisation par oubli de zero-place ait un sens (dans le cas des réseaux zero-saufs, ceci ne vaut que pour les jetons présents dans les zero-places). Etant donné un réseau de Petri, on dit qu'un ensemble de zero-places est *admissible* si toute exécution maximale du réseau est décomposable en une suite de pas. La propriété, pour un ensemble de zero-places, d'être admissible, est décidable. Il est assez facile de traduire les instructions de base de Signal en des réseaux équivalents, dont les ensembles de zero-places sont admissibles. La difficulté est de traduire la composition parallèle de ces instructions au niveau des réseaux : la réunion des ensembles de zero-places lors de la fusion des réseaux n'est pas sémantiquement correcte. La solution passe par l'implantation du calcul d'horloges de Signal dans les réseaux.

### 6.3 Réseaux de Petri, logique linéaire

**Mots clés** : réseau de Petri, logique linéaire, grammaires catégorielles.

**Participant** : Christian Retoré.

**Résumé** : *La logique linéaire, et en particulier ses variantes non commutatives ou mixtes, permet de décrire des grammaires formelles dites catégorielles et des systèmes concurrents. Dans cette dernière perspective, Ch. Retoré a modélisé l'exécution des réseaux de Petri dans le calcul mixte (commutatif / non commutatif de Ph. de Groote. Afin de rendre compte logiquement de grammaires utiles à la linguistique informatique, A. Lecomte (projet Calligramme et U. Grenoble II) et Ch. Retoré ont donné une définition logique des grammaires minimalistes de Stabler, qui formalisent les théories linguistiques récentes de Chomsky.*

#### 6.3.1 Réseaux de Petri et logique linéaire

**Participant** : Christian Retoré.

La logique linéaire a dès ses débuts été utilisée pour modéliser les réseaux de Petri. Si l'on souhaite traiter de questions telles que la synthèse de réseaux de Petri, ces codages souffrent

---

[BM97] R. BRUNI, U. MONTANARI, « Zero-Safe Nets, or Transition Synchronization Made Simple », *in* : *Proc. Express 97*, Electronic Notes in Theoretical Computer Science, 1997.

d'un gros défaut : les évènements sont absents du codage. En utilisant le calcul partiellement commutatif de [dG96], et en s'inspirant de la définition des grammaires de Lambek, Ch. Retoré a donné une description de l'exécution des réseaux de Petri [25]. Les places sont vues comme des variables propositionnelles, et un marquage (partiel) est un produit *commutatif* de places. Un évènement est décrit par l'implication *non commutative* d'un marquage partiel par un marquage partiel. Ce jeu entre connecteurs commutatifs et non commutatifs permet de rendre compte de l'exécution : un ordre série-parallèle  $O$  d'évènements  $e_i$  et de marquages partiels  $m_j$  entraîne, dans le calcul de [dG96], un marquage partiel  $M$  si et seulement si les évènements  $e_i$ , exécutés dans l'ordre  $O$  restreint aux  $e_i$  conduit du marquage initial produit (commutatif) des  $m_i$  au marquage  $M$ . On a donc une représentation assez fidèle de l'exécution des réseaux de Petri si ce n'est qu'on se limite à des exécutions séries-parallèles.

### 6.3.2 Grammaires minimalistes et logique linéaire

**Participant** : Christian Retoré.

Edward Stabler a proposé en 96 une formalisation du programme minimaliste de Chomsky sous la forme de grammaires d'arbres [Sta97]. Ce sont des grammaires totalement lexicalisées, où chaque entrée est une liste de traits, et où l'assemblage des constituants (*merge*, opération binaire) ainsi que leur déplacement à l'intérieur d'un arbre déjà constitué (*move*, opération unaire) sont gérés par l'annulation de traits de polarité opposée. Cela a permis à Ch. Retoré, en collaboration avec A. Lecomte (U. Grenoble II et projet Calligramme) de préciser et de formaliser le lien entre grammaires catégorielles et programme minimaliste, en faisant un autre usage grammatical du calcul de Lambek [Lam58] que celui proposé initialement. Les entrées lexicales à la Stabler sont traduites par des formules du calcul de Lambek, et on doit obtenir une démonstration dans le calcul de Lambek du symbole initial de la grammaire. Bien sûr, la difficulté est de rendre compte des déplacements de constituants; cela est fait en étiquetant la démonstration. Les étiquettes sont des chaînes qui comportent à la fois les informations phonologiques (les mots et les traits d'accord) et les informations sémantiques (à la Montague : variables, quantificateurs,...). Dans la règle d'élimination du produit, l'étiquette  $e : A \bullet B$  de la formule principale détermine quelles chaînes doivent remplacer les deux variables  $x : A$  et  $y : B$  que cette règle lie ; suivant la nature forte ou faible du trait  $A$  on a soit  $x := e$  et  $y := \epsilon$  soit  $x := (e)$  (la partie sémantique de  $e$ ) et  $y := /e/$  (la partie phonologique de  $e$ ). Ce genre de travaux intéresse les deux domaines rapprochés: d'une part les grammaires catégorielles peuvent ainsi profiter de la profondeur d'analyse de la théorie chomskienne, d'autre part les grammaires minimalistes acquièrent ainsi une interface plus aisée avec la sémantique, par exemple la sémantique de Montague. Le premier travail publié [17] présente les idées de base. Le second [16] étend la correspondance à des mouvements plus compliqués (dits de tête) en utilisant une extension conservatrice du calcul de Lambek : la logique linéaire intuitionniste

- 
- [dG96] P. DE GROOTE, « Partially commutative linear logic: sequent calculus and phase semantics », in : *Third Roma Workshop: Proofs and Linguistics Categories – Applications of Logic to the analysis and implementation of Natural Language*, CLUEB (Bologna), 1996.
- [Sta97] E. STABLER, « Derivational minimalism », in : *Logical Aspects of Computational Linguistics, LACL'96*, 1328, Springer Verlag LNCS/LNAI, 1997.
- [Lam58] J. LAMBEK, « The mathematics of sentence structure », *American mathematical monthly* 65, 1958.

partiellement commutative de Ph. de Grooté [dG96] ; le calcul de la sémantique de Montague est également abordé.

La connexion entre calcul de Lambek et grammaires minimalistes n'est qu'un aspect du rapprochement de l'approche générative (grammaires transformationnelles) et de l'approche logique (grammaires catégorielles). C'est pourquoi Ch. Retoré et E. Stabler (Ucla, Los Angeles) ont organisé un workshop sur « Resource logics and minimalist grammars » dans le cadre de l'école d'été « European Summer School in Logic, Language and Information » et ont écrit à ce propos un article de synthèse [11].

## 6.4 Graphes reconnaissables et dépliage

**Participants :** Didier Caucal, Christophe Morvan, Vincent Schmitt.

**Mots clés :** reconnaissabilité, décidabilité, dépliage.

**Résumé :** *Les graphes réguliers (de degré arbitraire) ont été caractérisés en tant que graphes reconnaissables. Une famille de graphes infinis plus étendue que celle des graphes reconnaissables a été étudiée, à savoir la famille des graphes rationnels, ou graphes étiquetés représentables par un ensemble fini de relations rationnelles sur les mots. Une caractérisation en a été donnée par images inverses de l'arbre binaire complet, étendant la caractérisation des graphes reconnaissables. L'étude des dépliages dans le cadre des enrichissements sur des bicatégories a par ailleurs abouti à la détermination d'une tricatégorie  $\text{caten}$  dont les objets sont les bicatégories.*

### 6.4.1 Graphes réguliers, reconnaissables, rationnels

**Participants :** Didier Caucal, Christophe Morvan.

D. Caucal et Ch. Morvan ont poursuivi l'étude des familles de graphes infinis de présentation finie présentée dans [Cau98]. Trois familles croissantes de graphes, les graphes réguliers, les graphes reconnaissables et les graphes rationnels, sont considérées. Pour chacune d'elles sont abordés les problèmes de représentation des graphes et les problèmes de décision de leurs propriétés [23].

Le coeur de l'étude est le passage effectif de la représentation externe d'un graphe, définie à un isomorphisme près, à sa représentation interne, dans laquelle chaque sommet est un mot. Ainsi, les graphes réguliers sont dans leur représentation externe les graphes engendrés par les grammaires déterministes de graphes (finis), alors que ces mêmes graphes, lorsque leur degré est fini, sont dans leur représentation interne les graphes de transitions des automates à pile [6].

---

[Cau98] D. CAUCAL, *Sur des graphes infinis réguliers*, thèse de doctorat, université de Rennes 1, février 1998, Document d'habilitation no.26 de l'Irisa.

De même, les graphes reconnaissables, vus de façon externe, sont les graphes dérivés de l'arbre binaire complet par substitutions rationnelles inverses sur les étiquettes des arcs et restrictions rationnelles sur l'ensemble des sommets, alors que ces graphes vus de façon interne sont les graphes de transitions préfixes des systèmes de réécriture reconnaissables sur les mots [8].

D. Caucal et T. Knapik (Université de La Réunion) ont caractérisé les graphes réguliers de degré arbitraire en tant que graphes reconnaissables [24].

Ch. Morvan a étudié la classe des graphes rationnels, dont la représentation interne est donnée en associant à chaque étiquette d'arc une relation rationnelle sur les mots. Il a donné de ces graphes une caractérisation analogue à celle des graphes reconnaissables : les graphes rationnels sont les graphes dérivés de l'arbre binaire complet par substitutions linéaires inverses sur les étiquettes des arcs et restrictions rationnelles sur l'ensemble des sommets. Ch. Morvan a aussi montré que la famille des langages de mots reconnus par des graphes rationnels (avec ensembles rationnels d'états initiaux/finaux) est une AFL (Famille Abstraite de Langages) contenant les langages algébriques.

#### 6.4.2 Une théorie du dépliage

**Participant** : Vincent Schmitt.

Le but est d'obtenir une théorie du dépliage qui généralise à la fois la théorie des espaces étales et la théorie des dépliages d'automates ou de graphes.

Le cadre algébrique choisi est celui des enrichissements sur des bicatégories. L'enrichissement sur des bicatégories permet de capturer la notion d'espace construit par assemblage de composants de base. Les enrichissements sur  $\mathcal{V}$  peuvent être interprétés comme des modèles de processus réalisant leur calcul dans  $\mathcal{V}$ . Ainsi, les automates sont décrits dans [BK82] comme des enrichissements sur la 2-catégorie ordonnée engendrée par leur alphabet de transitions. Les changements de base, modélisant alors des réétiquetages, raffinements ou abstractions sur les langages reconnus, correspondent toujours à la traduction correcte des calculs d'un langage vers un autre. Cet exemple a motivé une étude générale des changements de catégories de base, qui a donné lieu à un travail de V. Schmitt avec M. Kelly (Université de Sydney) et R. Street (Macquarie University, Sydney), présenté à la conférence Category 99 (Coimbra) par M. Kelly. Ce travail décrit une tricatégorie *caten* ayant pour objets les bicatégories.

---

[BK82] R. BETTI, S. KASANGIAN, « A quasi universal realization of automata », 1982, Rend. Ist. Mat. Univ. Trieste XIV.

## 7 Actions régionales, nationales et internationales

### 7.1 Actions régionales

#### 7.1.1 Application des réseaux de Petri aux services coopératifs

**Participants :** Eric Badouel, Philippe Darondeau, Danièle Quichaud, Andrew Tokmakoff.

Cette action s'inscrit dans le cadre d'une convention entre l'Inria et le Conseil Régional de Bretagne (no. 1 98 C 138 00 31323 06 1) (2 mars 98 au 31 mars 99), ayant pour objet la modélisation des systèmes de travail coopératif par des réseaux de Petri étendus, dont les jetons représentent les agents de ces systèmes (une description succincte en est donnée en 6.2). Les publications [15] et [20] complètent le rapport d'exécution produit à l'échéance de la convention.

### 7.2 Actions nationales

#### 7.2.1 Synthèse de commande des systèmes à événements discrets à l'aide des réseaux de Petri

**Participants :** Eric Badouel, Philippe Darondeau, Gilles Lesventes, Danièle Quichaud.

L'action coopérative Mars (1999-2000), dirigée par Xiaolan Xie (avant-projet MACSI), est menée par l'Inria avec l'ENS-Cachan et l'université de Bordeaux I. Elle porte sur la modélisation, la vérification et la synthèse de commande des systèmes à événements discrets, allant des systèmes de production aux systèmes de flux de tâches, avec des impératifs variables en matière de distribution. Elle a pour objectif de construire sur les réseaux de Petri une méthode de conception intégrée des systèmes à événements discrets contrôlés, et de contribuer ainsi au développement de logiciels sûrs et fiables. Notre contribution porte sur l'adaptation des diverses techniques de synthèse de réseaux dont nous disposons afin de permettre la construction de contrôleurs asynchrones en utilisant ces techniques (voir 6.1).

### 7.3 Actions européennes

Ch. Retoré est membre du réseau TMR « Linear Logic »

Ch. Retoré est le responsable français de l'action intégrée franco-italienne Galileo « Logiques linéaires non commutatives pour la syntaxe des langues romanes. » (Retoré ; Abrusci)

Ch. Retoré participe à l'action intégrée franco-néerlandaise van Gogh « Logique des ressources linguistiques et réseaux syntaxiques » (Lamarche ; Moortgat)

### 7.4 Actions internationales

#### 7.5 Visites, et invitations de chercheurs

D. Caucal a été invité deux mois à l'université de La Réunion.

J. Oliver (université de Valencia) a visité le projet du 2 février au 2 avril.

## 8 Diffusion de résultats

### 8.1 Animation de la Communauté scientifique

E. Badouel a participé au comité de programme du workshop Express (Eindhoven, août 99).

Ch. Retoré a édité un recueil d'articles intitulé *Recent Advances in Logical and Algebraic Approaches to Grammar*, paru chez Kluwer dans le volume 7 du journal *Journal of Logic Language and Information*.

Ch. Retoré a organisé avec E. Stabler (UCLA, Los Angeles) un workshop intitulé « Resource logics and minimalist grammars » dans le cadre de l'école d'été « European summer school of logic language and information ».

### 8.2 Enseignement

E. Badouel a donné un cours de 24 heures sur le thème *Outils de conception de protocoles, modélisation et validation à l'aide des réseaux de Petri* à l'ISAPT à Abidjan (Côte d'Ivoire) en décembre 1998.

D. Caucal a enseigné 15 heures en DEA d'informatique à Rennes (calculabilité et complexité).

### 8.3 Rapports de thèse

E. Badouel a été rapporteur de la thèse de Rémi Morin, intitulée « Catégorie de modèles du parallélisme » soutenue au LRI, université de Paris XI Orsay en janvier 1999.

Ph. Darondeau a été rapporteur de la thèse de Robert Riemann, intitulée « Modelling of Concurrent Systems: Structural and Semantical Methods in the High Level Petri Net Calculus », soutenue à l'université de Paris XI Orsay en février 1999.

Ch. Retoré a été rapporteur de la thèse d' Akim Demaille, intitulée « Logiques linéaires hybrides et leurs modalités », soutenue à l'ENST Paris en juin 99.

### 8.4 Séminaires

D. Caucal a présenté un exposé intitulé « Sur des familles de graphes infinis » aux journées « Systèmes Infinites » organisées par le GDR ALP et le LSV à l'ENS Cachan en novembre 1998.

D. Caucal a présenté un exposé intitulé « Sur des graphes réguliers » à l'université de La Réunion

Ph. Darondeau a présenté un exposé intitulé « Perspectives in Petri Net Synthesis » au meeting annuel du WG2.2 de l'Ifip à Udine en juillet 1999.

Ph. Darondeau a présenté un exposé intitulé « Synthèse de réseaux de Petri généraux » à la réunion du 24 septembre du groupe francophone sur les réseaux de Petri.

D. Harlet a présenté un exposé intitulé « Representation of Synchronous Programs by Nets » à l'université de Pise en décembre 1998.

Ch. Retoré a présenté un exposé intitulé « Towards a minimal logic for minimalist grammars » au congrès triennal de la Società Italiana di Logica e Filosofia delle Scienze (Cesena/Urbino), ainsi qu'à une réunion pour le lancement de la collection Research in Logic and Formal Linguistics à Bologne en février 99.

Ch. Retoré a participé à la rencontre de l'action intégrée van Gogh en mars 99 à Utrecht, où il a présenté ses travaux sur « Réécriture de cographes orientés et déduction en logique linéaire ».

Ch. Retoré a présenté ses travaux sur « A representation of Petri-net execution in partially commutative linear logic » au Logic Colloquium, LC'99, en août 99.

Ch. Retoré a été invité au workshop « Linear logic and applications » où il a présenté un exposé sur « Linear logic and Chomsky's minimalist program for linguistics » à Dagstuhl en août 99.

## 9 Bibliographie

### Ouvrages et articles de référence de l'équipe

- [1] E. BADOUEL, L. BERNARDINELLO, P. DARONDEAU, « Polynomial algorithms for the synthesis of bounded nets », *in: CAAP'95, 915*, Springer Verlag LNCS, p. 364–378, Aarhus, 1995.
- [2] E. BADOUEL, L. BERNARDINELLO, P. DARONDEAU, « The synthesis problem for elementary net systems is NP-complete », *TCS 186*, 1997, p. 107–134.
- [3] E. BADOUEL, P. DARONDEAU, J.-C. RAOULT, « Context-free event domains are recognizable », *Information and Computation*, 149, 1999, p. 134–172.
- [4] E. BADOUEL, P. DARONDEAU, « Dualities between nets and automata induced by shizophrenic objects », *in: CTCS'95, 953*, Springer Verlag LNCS, p. 24–43, Cambridge, 1995.
- [5] D. CAUCAL, « Monadic theory of term rewriting », *in: 7th IEEE Symp., LICS, IEEE*, p. 266–273, Santa Cruz, 1992.
- [6] D. CAUCAL, « On the regular structure of prefix rewriting », *TCS 106*, 1992, p. 61–86.
- [7] D. CAUCAL, « Bisimulation of context-free grammars and of pushdown automata », *in: Modal logic and process algebra, 53*, Stanford University CSLI Lecture Notes, 1995, p. 85–106.
- [8] D. CAUCAL, « On infinite transition graphs having a decidable monadic theory », *in: ICALP'96, 1099*, Springer Verlag LNCS, p. 194–205, Paderborn, 1996.

- [9] P. DARONDEAU, « Deriving Unbounded Petri Nets from Formal Languages », *in: Concur'98, 1466*, Springer-Verlag LNCS, p. 533–548, Nice, 1998.
- [10] V. SCHMITT, « CR-structures », *Fundamenta Informaticae 33(1)*, 1998, p. 71–83.

### Livres et monographies

- [11] C. RETORÉ, E. STABLER (éditeurs), *Resource Logics and Minimalist Grammars, European Summer School in Logic Language and Information*, Foundation of Logic, Language and Information, Utrecht, 1999. Featuring a twenty-page introduction by the editors.

### Thèses et habilitations à diriger des recherches

- [12] E. BADOUEL, *Automates réversibles et réseaux de Petri, dualité et représentation: le problème de synthèse*, thèse de doctorat, université de Rennes 1, avril 1999, Document d'habilitation no.31 de l'Irisa.

### Articles et chapitres de livre

- [13] E. BADOUEL, P. DARONDEAU, « Theory of regions », *in: Lectures on Petri Nets I: Basic Models, 1491*, Springer-Verlag LNCS, 1999, p. 529–586.
- [14] C. RETORÉ, « Handsome proof-nets: perfect matchings and cographs », *Theoretical Computer Science*, 1999, à paraître.

### Communications à des congrès, colloques, etc.

- [15] E. BADOUEL, P. DARONDEAU, A. TOKMAKOFF, « Modelling Dynamic Agents Systems with Cooperating Automata », *in: PDPTA'99*, CSREA Press (éditeur), I, p. 11–17, Las Vegas, 1999.
- [16] A. LECOMTE, C. RETORÉ, « A logical formulation of the minimalist program », *in: Third Tbilisi Symposium on Language, Logic and Computation*, CSLI, 1999.
- [17] A. LECOMTE, C. RETORÉ, « Towards a Minimal Logic for Minimalist Grammars: a Transformational Use of Lambek Calculus », *in: Formal Grammar, FG'99*, FoLLI, p. 83–92, 1999.
- [18] C. RETORÉ, « Pomset logic as a calculus of directed cographs », *in: Fourth Roma workshop, Dynamic Perspectives in Logic and Linguistics: Proof Theoretical Dimensions of Communication Processes*, Bulzoni (éditeur), 1999. à paraître.
- [19] A. TOKMAKOFF, J. BILLINGTON, « An approach to the Analysis of Interworking Traders », *in: ICATPN'99*, S. V. LNCS (éditeur), p. 127–146, Williamsburg, 1999.

### Rapports de recherche et publications internes

- [20] E. BADOUEL, P. DARONDEAU, D. QUICHAUD, A. TOKMAKOFF, « Modelling Dynamic Agents Systems with Cooperating Automata », *publication interne n° 1253*, Irisa, juin 1999.
- [21] E. BADOUEL, J. OLIVER, « Dynamic Changes in Concurrent Systems: Modelling and Verification », *rapport de recherche n° 3708*, Inria, juin 1999.
- [22] P. DARONDEAU, « On the Petri Net Realization of Context-Free Graphs », *Rapport de Recherche n° 3674*, INRIA, 1999, soumis à publication.

**Divers**

- [23] O. BURKART, D. CAUCAL, F. MÖLLER, B. STEFFEN, « Verification over infinite states », 1999, chapitre de livre en cours d'édition.
- [24] D. CAUCAL, T. KNAPIK, « On internal presentation of regular graphs », 1999, submitted.
- [25] C. RETORÉ, « A representation of Petri-net execution in partially commutative linear logic », 1999, texte présenté au Logic Colloquium LC'99.