

# *Projet PARAGRAPHÉ*

*Parallélisme et graphes*

*Rennes*

THÈME 1C



*R*apport  
*d'Activité*

2000



## 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	Synthèse des réseaux de Petri et théorie des régions . . . . .	4
3.1.1	Un aperçu des réseaux de Petri . . . . .	4
3.1.2	Théorie des régions . . . . .	5
3.1.3	Synthèse de réseaux à partir de langages . . . . .	7
3.2	Logique linéaire . . . . .	9
3.2.1	Grammaires catégorielles et calcul de Lambek . . . . .	9
3.2.2	Logique linéaire . . . . .	11
<b>4</b>	<b>Domaines d'applications</b>	<b>14</b>
4.1	Panorama . . . . .	14
4.1.1	Aide à la mise en œuvre de logiciels distribués . . . . .	15
4.1.2	Synthèse de contrôleurs asynchrones . . . . .	15
4.1.3	Inférence grammaticale . . . . .	16
4.1.4	Analyse syntaxique concurrente . . . . .	16
<b>5</b>	<b>Résultats nouveaux</b>	<b>17</b>
5.1	Graphes de marquages, théorie des régions, synthèse . . . . .	17
5.1.1	Distribution d'automates finis via la synthèse de réseaux de Petri bornés	17
5.1.2	Synthèse de réseaux à partir de HMSCs . . . . .	17
5.1.3	Contrôle des graphes marqués . . . . .	18
5.1.4	Automates généralisés et réseaux de Petri . . . . .	18
5.2	Réseaux de Petri, logique linéaire . . . . .	19
5.2.1	Réseaux de Petri et logique linéaire . . . . .	19
5.2.2	Grammaires catégorielles, grammaires algébriques et inférence grammaticale . . . . .	20
5.2.3	Grammaires minimalistes et logique linéaire . . . . .	21
5.3	Actions nationales . . . . .	22
5.3.1	Synthèse de commande des systèmes à événements discrets à l'aide des réseaux de Petri . . . . .	22
5.4	Actions européennes . . . . .	22
5.5	Actions internationales . . . . .	22
5.6	Visites et invitations de chercheurs . . . . .	22
<b>6</b>	<b>Diffusion de résultats</b>	<b>23</b>
6.1	Animation de la Communauté scientifique . . . . .	23
6.2	Enseignement . . . . .	23
6.3	Rapports de thèse . . . . .	23
6.4	Séminaires . . . . .	23

**7 Bibliographie****24**

## 1 Composition de l'équipe

### Responsable scientifique

Philippe Darondeau [DR Inria]

### Personnel Inria

Eric Badouel [CR - détaché depuis le 1/09/99]

Boubakar Gamatié [CR]

Christian Retoré [CR]

### Personnel INSA

Danielle Quichaud [maître de conférences]

### Personnel université de Rennes 1

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

### Chercheurs doctorants

Yannick Le Nir [allocataire MENRT depuis le 1/10/99]

## 2 Présentation et objectifs généraux

Paragraphe est un projet de l'UMR Irisa.

Nous étudions divers algorithmes de synthèse de réseaux de Petri, applicables à la parallélisation, à la distribution ou au contrôle de 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, au cœur des réseaux de Petri, et qui fournit souvent des solutions effectives échappant à l'intuition. Le projet étudie dans un but fondamental les représentations algébriques et logiques des réseaux de Petri et de leurs graphes de marquages, et accessoirement les propriétés 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. Le projet cherche aussi à consolider divers rapprochements entre les concepts issus du domaine de la concurrence et les concepts utiles au traitement des langues naturelles. L'objectif à long terme est le traitement concurrent des langues naturelles.

Les axes de recherche les plus actifs aujourd'hui sont:

- la réalisation de langages de scénarios par des réseaux de Petri distribuables,

- la synthèse de contrôleurs asynchrones via les réseaux de Petri,
- l'instrumentation de la synthèse de réseaux,
- la formalisation des réseaux de Petri par la logique linéaire,
- la formalisation du programme minimaliste de Chomsky par la logique linéaire.

Le projet est engagé en ce qui concerne la synthèse des réseaux dans le protocole de coopération franco-polonaise «Catalysis» (CNRS/IPIPAN); le projet participe quant aux autres thèmes au réseau TMR «Linear Logic» et à l'action intégrée franco-néerlandaise «Logique des ressources linguistiques et réseaux syntaxiques» (programme van Gogh).

### 3 Fondements scientifiques

#### 3.1 Synthèse des réseaux de Petri et théorie des régions

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

**Glossaire :**

**synthèse** 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.

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

**réseau distribuable** Réseau de Petri dans lequel les événements sont attachés à des sites tels que deux événements distants ne partagent pas de place d'entrée.

**Résumé :** *suivent une brève introduction aux réseaux de Petri, avec quelques renvois à la littérature, une présentation de la théorie des régions d'Ehrenfeucht et Rozenberg étendue à des types de réseaux arbitraires, avec ses applications à la synthèse de réseaux, et une adaptation de cette théorie à la synthèse de réseaux à partir de langages.*

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

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 œuvre 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 est définie par un jeu de jetons représentant les conditions valides: les places en entrée (resp. en sortie) d'une transition sont ses pré-conditions (resp. post-conditions).

---

[Mur89] T. MURATA, «Petri Nets: Properties, Analysis and Applications», *Proc. IEEE* 77(4), 1989.

Il s'est par la suite avéré utile de permettre l'accumulation des jetons dans les places, vues dès lors comme des 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 (ou P/T-nets), dans lesquels les places prennent des valeurs entières. Les réseaux distribuables de B. Caillaud [Cai99] sont une version enrichie des réseaux P/T.

Les travaux sur les réseaux P/T ont surtout été consacrés à l'étude de leurs graphes de marquages. Ces travaux reposent sur l'algèbre linéaire et les invariants [MR80] [STC99], sur la théorie des graphes et les symétries [HITZ95], ou sur le mélange de ces deux 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 P/T, 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 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 alors est de caractériser les familles de graphes ou de langages correspondant à une classe de réseaux donnée. La théorie des régions, présentée de façon détaillée dans [3] et adaptée aux langages dans [6], est l'un des éléments de réponse à ce problème. On pourra consulter [15] pour une présentation brève de l'ensemble du sujet.

### 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  $\tau \subseteq LS \times$

- 
- [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.
  - [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.
  - [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.

$LE \times LS$  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 dont les états sont tous accessibles à partir de  $s_0$  (l'état initial). Une *région* [ER90a] 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 [ER90b]:

- 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. 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 [1]. 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

[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.

des régions de  $A$  à un sous ensemble irredondant 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  $A$  [BBD95]. 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  $N$  irredondant 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  $\mathcal{Z}$  finiment engendrés, représentant toutes les composantes  $\eta$  des régions entières de  $A$ , ou morphismes de  $A$  dans  $\tau = (\mathbb{N}, \mathcal{Z}, \rightarrow)$ . La complexité de la synthèse est la même pour les réseaux P/T [BD96]. Un développement récent sur le sujet de la synthèse des réseaux P/T est la construction d'une procédure qui décide ce problème pour les graphes infinis réguliers [11].

### 3.1.3 Synthèse de réseaux à partir de langages

Le problème de synthèse de réseaux à partir de langages consiste à approximer au mieux un langage  $L \subseteq E^*$  fermé par préfixes et non vide (pris comme entrée) par le langage d'un réseau (à déterminer) dans une classe fixée de réseaux de Petri. Les approximations considérées sont des approximations supérieures pour l'inclusion des langages. L'étiquetage des événements des réseaux est injectif et non effaçant, et tous les marquages sont traités comme des états accepteurs. Sous ces hypothèses, le langage d'un réseau initialisé est l'intersection des langages de ses sous-réseaux atomiques (restrictions induites par élimination de toutes les places sauf une). L'intersection de tous les langages de réseaux plus grands que  $L$  est un langage de réseau et c'est la fermeture  $\bar{L}$  de  $L$  dans l'ensemble des langages de réseaux de Petri.

La fermeture  $\bar{L}$  de  $L$  n'est généralement pas le langage d'un réseau de Petri *fini* (c.à.d. ayant un ensemble fini de places). Un premier problème est de déterminer des conditions suffisantes sur  $L$  assurant que sa fermeture  $\bar{L}$  puisse être réalisée par un réseau fini. Un second problème est de calculer ce réseau fini (s'il existe) en partant d'une grammaire ou de tout autre générateur du langage  $L$ . Un troisième problème est de décider si l'approximation obtenue est exacte.

Supposons fixé dorénavant un type de réseaux  $\tau = (LS, LE, \tau)$ . Un langage  $L \subseteq E^*$  non vide et fermé par préfixes peut être identifié avec le système de transitions  $T(L) = (S, E, T)$  tel que  $S = L$  et  $T = \{u \xrightarrow{e} v \mid u, v \in L \wedge ue = v\}$ .

Mieux encore,  $L$  peut être identifié avec le système de transitions initialisé  $T(L)$  d'état initial  $\varepsilon$  (le mot vide). Appelons régions de  $L$  les régions de  $T(L)$ . On voit immédiatement que  $L$  coïncide avec le langage d'un réseau initialisé si et seulement si la condition de séparation relative aux événements non tirables est satisfaite dans  $T(L)$ . Ceci donne une caractérisation uniforme des langages de réseaux de Petri, mais ne mène généralement pas à une solution effective du problème de synthèse.

---

[BBD95] 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.

[BD96] E. BADOUEL, P. DARONDEAU, « On the Synthesis of General Petri Nets », *rapport de recherche n° 3025*, Inria, november 1996.

Afin d'obtenir une solution pratique de ce problème, on peut se concentrer sur les réseaux P/T. Une région de  $L$  ou de  $T(L)$  est alors une paire  $(\sigma, \eta)$  dans laquelle  $\sigma : L \rightarrow \mathbb{N}$  et  $\eta : E \rightarrow \mathbb{N} \times \mathbb{N}$  sont deux applications telles que, en notant  $\eta(e) = (\bullet\eta(e), \eta^\bullet(e))$ , on ait  $\sigma(u) \geq \bullet\eta(e)$  et  $\sigma(v) = \sigma(u) - \bullet\eta(e) + \eta^\bullet(e)$  dès que  $u, v \in L \wedge ue = v$ . Une région  $(\sigma, \eta)$  de  $L$  ou de  $T(L)$  peut maintenant être identifiée avec un vecteur d'entiers non négatifs

$$p = (M_0(p), p^\bullet e_1, e_1^\bullet p, \dots, p^\bullet e_n, e_n^\bullet p)$$

où  $M_0(p) = \sigma(\varepsilon)$ ,  $p^\bullet e_i = \bullet\eta(e_i)$ , et  $e_i^\bullet p = \eta^\bullet(e_i)$  pour tout  $i$ . Comme  $T(L)$  est un arbre, dont les branches sont les mots de  $L$ , un vecteur  $p$  comme ci-dessus représente une région de  $L$  si et seulement si, pour tout  $u \in L$  et pour tout  $e \in E$  tels que  $ue \in L$ ,

$$M_0(p) + \sum_{i=1}^n u(e_i) \times (e_i^\bullet p - p^\bullet e_i) - p^\bullet e \geq 0$$

où  $u(e_i)$  compte les occurrences  $e_i$  dans  $u$ . Cette condition engendre un système infini d'inégalités linéaires qui ne permet généralement pas le calcul pratique de l'ensemble des régions.

On rencontre une situation spéciale lorsque  $L$  est un langage régulier ou hors-contexte [6] et plus généralement lorsque les images de Parikh des résidus à droite  $L/e = \{u \mid ue \in L\}$  sont des ensembles semi-linéaires, c.à.d. lorsque pour chaque événement  $e \in E$ , l'ensemble des vecteurs de tir

$$\Psi(L/e) = \{(u(e_1), \dots, u(e_n)) \mid u \in L/e\}$$

est une union *finie* d'ensembles *linéaires*  $\Psi(vW^*)$  où  $v$  est un mot de  $E^*$  et  $W$  est un ensemble fini de mots de  $E^*$ . Dans ce cas, un vecteur comme ci-dessus représente une région de  $L$  si et seulement si pour chaque événement  $e$ , pour chaque sous-ensemble linéaire  $\Psi(vW^*)$  de  $L/e$ , et pour chaque mot  $w$  de l'ensemble fini  $W$ , les conditions suivantes sont satisfaites:

$$M_0(p) + \sum_{i=1}^n v(e_i) \times (e_i^\bullet p - p^\bullet e_i) - p^\bullet e \geq 0$$

$$\sum_{i=1}^n w(e_i) \times (e_i^\bullet p - p^\bullet e_i) \geq 0$$

Au total, on obtient un ensemble fini d'inégalités linéaires homogènes, donc les régions de  $L$  sont tous les vecteurs entiers d'un cône polyédrique. Le réseau canonique  $\mathcal{N}(L)$  ayant pour places  $\{p_1, \dots, p_m\}$  les régions définies (à un coefficient multiplicatif près) par les rayons extrémaux de ce cône polyédral réalise la fermeture  $\overline{L}$  de  $L$  dans les langages de réseaux de Petri. Le réseau synthétisé  $\mathcal{N}(L)$  donne la meilleure approximation de  $L$  par un langage de réseau de Petri, mais on ne peut généralement pas décider si cette approximation est exacte, c.à.d. si l'ensemble  $\{p_1, \dots, p_m\}$  contient suffisamment de témoins de validité de la condition de séparation relative aux événements non tirables.

La décision de cette question est possible dans le cas particulier où  $L$  est un langage régulier ou un langage déterministe hors-contexte [6], et plus généralement lorsque les images de Parikh des ensembles de refus  $(L \ominus e) = \{u \mid u \in L \wedge ue \notin L\}$  sont semi-linéaires. Dans ce cas, tester la condition de séparation relative aux événements revient à vérifier, pour chaque

événement  $e$  et pour chaque sous-ensemble linéaire  $\Psi(vW^*)$  de  $(L \ominus e)$ , que les conditions suivantes sont satisfaites pour chacun des mots  $w$  de l'ensemble fini  $W$  et pour l'une des places  $p \in \{p_1, \dots, p_m\}$  (dépendant de  $w$ ):

$$M_0(p) + \sum_{i=1}^n v(e_i) \times (e_i \bullet p - p \bullet e_i) - p \bullet e < 0$$

$$\sum_{i=1}^n w(e_i) \times (e_i \bullet p - p \bullet e_i) = 0$$

Lorsque  $L$  est un langage régulier, on peut aussi décider directement l'inclusion  $\bar{L} \subseteq L$  en utilisant la procédure proposée par Jancar et Moeller [JM95]. Cette procédure permet également de répondre au problème de synthèse de réseaux à partir d'encadrements ou de tolérances: étant donné deux langages réguliers  $L$  et  $L'$  tels que  $L \subseteq L'$ , décider s'il existe et construire un réseau P/T fini  $N$  dont le langage  $\mathcal{L}(N)$  contienne  $L$  et soit contenu dans  $L'$ . De fait, il existe une solution si et seulement si le réseau canonique  $\mathcal{N}(L)$  en est une, c.à.d. si et seulement si  $\bar{L} \subseteq L'$ .

## 3.2 Logique linéaire

**Mots clés :** grammaires catégorielles, logique linéaire, réseaux de démonstration, calcul mixte commutatif.

### Glossaire :

**grammaires catégorielles** grammaires formelles entièrement définies par un lexique qui associe à chaque terminal un ou plusieurs types, le calcul de types étant fixé par des règles de déduction logique indépendantes de la grammaire décrite

**logique linéaire** logique dans laquelle on ne peut ni affaiblir ni contracter les hypothèses qui sont traitées comme des ressources

**réseaux de démonstration** représentations de classes de démonstrations équivalentes par des graphes, reconnus comme tels par l'absence d'un certain type de cycles

**calcul mixte commutatif** calcul intuitionniste combinant connecteurs non commutatifs à la Lambek ( $\backslash, /, \bullet$ ) et connecteurs commutatifs ( $-\circ, \otimes$ ) de la logique linéaire

**Résumé :** *suivent une présentation des grammaires catégorielles et du calcul de Lambek, puis une introduction à la logique linéaire qui le généralise, en particulier dans le calcul mixte commutatif / non-commutatif de Ph. de Groote.*

### 3.2.1 Grammaires catégorielles et calcul de Lambek

Les grammaires catégorielles sont des grammaires formelles fondées sur la notion de type. Elles ont été introduites par Bar-Hillel [BH53] en 1953 (à peu près en même temps que les grammaires de la hiérarchie de Chomsky). À la différence de grammaires usuelles comme les

---

[JM95] P. JANCAR, F. MOELLER, « Checking Regular Properties of Petri Nets », *in: Proc. Concur*, 962, Springer Verlag LNCS, 1995.

[BH53] Y. BAR-HILLEL, « A quasi arithmetical notation for syntactic description », *Language* 29, 1953, p. 47-58.

grammaires algébriques, elles sont lexicalisées, c'est-à-dire que seul le lexique varie d'un langage à un autre: les règles de grammaire sont fixées une fois pour toutes. Cela présente de nombreux avantages algorithmiques, par exemple pour l'apprentissage (ou inférence grammaticale), et en même temps cela rejoint les théories et formalismes linguistiques actuels (les théories Chomskyennes depuis Principes et Paramètres en 86, ou les formalismes tels TAGs et HPSGs).

Le lexique associe à chaque mot un ou plusieurs types (ou formules) qui modélisent le comportement syntaxique de l'item lexical. Ces types sont constitués à partir de types de base, généralement  $n$  (nom commun),  $sn$  (groupes nominaux),  $S$  (phrases complètes, l'analogue du symbole initial d'une grammaire usuelle). Les constructeurs de types sont  $/$  et  $\backslash$ . L'intuition est simple: un objet  $x$  est de type  $A\backslash B$  (resp.  $B/A$ ) si chaque fois qu'il est précédé (resp. suivi) d'un objet  $a$  de type  $A$ , l'objet obtenu  $ax$  (resp.  $xa$ ) est de type  $B$ .

Notons  $\text{Lex}(m_i)$  l'ensemble fini de types associé par le lexique au mot (ou terminal)  $m_i$ . Une suite de mots  $s = m_1 \dots m_n$  est dans le langage lorsqu'il est possible de trouver pour chaque mot  $m_i$  de la suite  $s$  un type  $T_i^j \in \text{Lex}(m_i)$  tel que la suite des types  $T_1^j \dots T_n^j$  se réduise en  $S$ . La réduction notée  $\vdash$  est simplement définie par les deux schémas  $A(A\backslash B) \vdash B$  et  $(B/A)A \vdash B$ . Comme annoncé ci-dessus, les règles sont indépendantes de la langue (ou du langage formel): la grammaire est définie par son lexique, et lui seul varie d'une langue à l'autre.

Rien qu'avec ces règles de simplification de fractions, on peut représenter tous les langages algébriques, mais c'est peu pratique, et pour des raisons de modélisation linguistique, on a cherché à ajouter toutes sortes de principes, par exemple la transitivité  $(A\backslash B)(B\backslash C) \vdash (A\backslash C)$ , la montée de type (très utile en sémantique)  $A \vdash B \backslash (A/B) \dots$

En 1958, notant la similitude entre  $A\backslash B$  et  $A \Rightarrow B$  et entre  $B/A$  et  $B \Leftarrow A$ , Lambek [Lam58a] a eu l'idée de présenter ces grammaires comme un calcul logique, dont les règles pour  $\backslash$  et  $/$  sont les règles usuelles de l'implication.

$$\frac{\Gamma, B, \Gamma' \vdash C \quad \Delta \vdash A}{\Gamma, \Delta, A\backslash B, \Gamma' \vdash C} \backslash_h \quad \frac{A, \Gamma \vdash C}{\Gamma \vdash A\backslash C} \backslash_i \quad \Gamma \neq \emptyset$$

$$\frac{\Gamma, B, \Gamma' \vdash C \quad \Delta \vdash A}{\Gamma, B/A, \Delta, \Gamma' \vdash C} /_h \quad \frac{\Gamma, A \vdash C}{\Gamma \vdash C/A} /_i \quad \Gamma \neq \emptyset$$

On peut aussi ajouter la conjonction:

$$\frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, A \bullet B, \Gamma' \vdash C} \bullet_h \quad \frac{\Delta \vdash A \quad \Gamma \vdash B}{\Delta, \Gamma \vdash A \bullet B} \bullet_i$$

Les grammaires de Lambek sont définies comme les grammaires de Bar-Hillel. Le lexique associe à chaque mot (ou terminal)  $m_i$  un ensemble fini de types  $\text{Lex}(m_i)$  (comportant éventuellement la conjonction  $\bullet$ ); une suite de mots  $s = m_1 \dots m_n$  est dans le langage lorsque

$$\forall i \in [1, n] \exists T_i \in \text{Lex}(m_i) \quad T_1 \dots T_n \vdash S$$

---

[Lam58a] J. LAMBEK, «The Mathematics of Sentence Structure», *American Mathematical Monthly* 65, 1958, p. 154–170.

On remarquera que ce système ne permet pas d'ajouter des hypothèses, ni de considérer plusieurs occurrences de la même hypothèse comme une seule hypothèse: il s'agit de logique linéaire avant l'heure. Il n'y a, en effet, ni règle d'affaiblissement, ni règle de contraction:

$$\frac{\Gamma, \Gamma' \vdash C}{\Gamma, A, \Gamma' \vdash C} \text{ aff.} \qquad \frac{\Gamma, A, A, \Gamma' \vdash C}{\Gamma, A, \Gamma' \vdash C} \text{ contr.}$$

De plus, les hypothèses ne commutent pas car il n'y a pas de règle d'échange:

$$\frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, B, A, \Gamma' \vdash C} \text{ éch.}$$

C'est la linéarité du calcul qui permet là encore d'obtenir un bon calcul logique non commutatif où les deux implications ne sont pas équivalentes: l'une s'applique à gauche et l'autre à droite.

Côté grammairales, on reconnaît alors tous les langages algébriques, mais on a beaucoup plus d'arbres d'analyse qu'avec les grammairales hors-contexte (les ensembles d'arbres d'analyse passent des langages réguliers d'arbres à des langages hors-contexte [Tie99a]); la correspondance entre types syntaxiques et types sémantiques est immédiate:

$$\begin{aligned} S^s &= t \quad (\text{valeurs de vrits}) \\ sn^s &= e \quad (\text{individus}) \\ n^s &= (e \rightarrow t) \quad (\text{prédicats à un argument}) \\ (A \setminus B)^s &= (B/A)^s = A \rightarrow B \quad (\text{pour la traduction des types composés}) \end{aligned}$$

La structure produite par l'analyse syntaxique est transformée par cette correspondance en une structure sémantique, qui plus est compositionnelle vis à vis des  $\lambda$ -termes de Montague, car le calcul de Lambek est un sous-calcul de la logique intuitionniste (qui correspond au  $\lambda$ -calcul simplement typé utilisé en sémantique).

On pourra consulter [13] et [16] pour un complément d'information sur les grammairales catégorielles et le calcul de Lambek.

### 3.2.2 Logique linéaire

Le calcul de Lambek est longtemps resté isolé des logiques traditionnelles (logiques classique, intuitionniste, modale), car son lien exact avec ces logiques est demeuré obscur jusqu'aux années 80. C'est en particulier grâce à l'invention de la logique linéaire en 86 par J.-Y. Girard [Gir87] que ce lien a été mieux compris. La logique linéaire [Gir95] [GLR95] est commutative dans

- 
- [Tie99a] H.-J. TIEDE, *Deductive Systems and Grammars: Proofs as Grammatical Structures*, thèse de doctorat, Illinois Wesleyan University, 1999.
- [Gir87] J.-Y. GIRARD, «Linear Logic», *TCS* 50, 1, 1987, p. 1–102.
- [Gir95] J.-Y. GIRARD, «Linear logic: its syntax and semantics», in: Girard *et al.* [?], p. 1–42.
- [GLR95] J.-Y. GIRARD, Y. LAFONT, L. REGNIER (éditeurs), *London Mathematical Society Lecture Notes*, 222, Cambridge University Press, 1995.

sa version d'origine, contrairement au calcul de Lambek (elle contient la règle d'échange), mais elle ne contient pas non plus les règles usuelles de contraction et d'affaiblissement (voir ci-dessus), et permet seulement d'en faire usage dans des formules gouvernées par les modalités ? et ! (à rapprocher de  $\diamond$  et  $\square$ ):

$$\frac{\Gamma, !A, !A, \Gamma' \vdash C}{\Gamma, !A, \Gamma' \vdash C} \text{contr.} \quad \frac{\Gamma, \Gamma' \vdash C}{\Gamma, !A, \Gamma' \vdash C} \text{aff.}$$

On peut alors retrouver la logique intuitionniste par la traduction  $(A \Rightarrow B)^\circ = !A \multimap B$  où  $\multimap$  est la notation pour l'implication linéaire: une formule est démontrable en logique intuitionniste si et seulement si sa traduction en logique linéaire est démontrable en logique linéaire. Le calcul de Lambek apparaît alors comme le fragment intuitionniste de la logique linéaire non-commutative restreinte aux connecteurs dits multiplicatifs [10]. En l'absence de règle d'échange  $\multimap$  représente deux implications à travers cette traduction, l'une trouvant son argument sur sa gauche et l'autre sur sa droite: ce sont respectivement les  $\backslash$  et  $/$  du calcul de Lambek.

Du point de vue du traitement des langues, le plongement du calcul de Lambek dans le fragment intuitionniste de la logique linéaire non commutative permet de passer des types syntaxiques de ce calcul aux types de la sémantique de Montague: les  $\lambda$ -termes simplement typés utilisés par Montague pour représenter les formules du calcul des prédicats peuvent en effet être vus, par l'isomorphisme de Curry-Howard, comme des démonstrations en logique intuitionniste.

Un autre aspect intéressant de la logique linéaire est la facilité qu'elle offre pour décomposer les connecteurs de la logique usuelle en des connecteurs plus élémentaires et pour définir des fragments et des variantes (non commutatives, par exemple) qui soient bien fondées logiquement (c.à.d. avec élimination des coupures, propriété de la sous-formule, complétude, sémantique dénotationnelle, etc.).

On peut ainsi étendre les grammaires de Lambek tout en préservant le caractère logique du modèle grammatical: une analyse syntaxique est une démonstration dans un calcul logique, ce qui constitue un premier pas vers la représentation sémantique des énoncés [8] [7].

Une variante particulièrement intéressante à cet égard de la logique linéaire a été introduite en 1996 par Ph. de Groote [dG96]: il s'agit d'un calcul intuitionniste qui combine les connecteurs non commutatifs à la Lambek ( $\backslash, /, \bullet$ ) et les connecteurs commutatifs ( $\multimap, \otimes$ ) de la logique linéaire. Les « virgules » entre hypothèses correspondant à la conjonction, on a deux types de « virgules » pour former des contextes,  $(\dots; \dots)$  et  $\{\dots, \dots\}$ , et ces contextes sont en fait des ordres série-parallèles de formules: dans  $(\Gamma; \Delta)$  le contexte  $\Gamma$  précède le contexte  $\Delta$ , tandis que le contexte  $\{\Gamma; \Delta\}$  est la somme disjointe des contextes  $\Gamma$  et  $\Delta$ . Les règles sont celles qu'on imagine: de  $(A; \Gamma) \vdash C$  on déduit par exemple  $\Gamma \vdash A \backslash C$ , tandis que de  $\{A, \Gamma\} \vdash C$  on déduit  $\Gamma \vdash A \multimap C$ . Pour avoir une relation entre les connecteurs non-commutatifs et les connecteurs commutatifs, il faut s'autoriser à remplacer un contexte par un autre construit sur le même ensemble de formules en relâchant ou au contraire en renforçant l'ordre partiel sur ces formules. En fait les deux calculs obtenus avec l'une ou l'autre règle de relâchement ou de renforcement

---

[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.

des contextes fonctionnent, et ils sont faciles à manier car l'inclusion des ordres série-parallèles s'axiomatise finiment par des réécritures modulo associativité et commutativité [5]. Ces deux calculs ont l'élimination des coupures et la propriété de la sous-formule, et admettent des modèles vis-à-vis desquels ils ont la propriété d'être complets, mais ils n'ont pas à présent de réseaux de démonstration.

Une autre possibilité de calcul mixte commutatif, par ajout à la logique linéaire commutative d'un connecteur non commutatif avait été étudiée par C. Retoré en 93 [Ret93] [Ret97]. Ce connecteur  $<$ , issu de la sémantique dénotationnelle de la logique linéaire, est autodual:  $A^\perp < B^\perp \equiv (A < B)^\perp$ . Le calcul obtenu étend fidèlement le calcul commutatif, mais il n'a pas actuellement de syntaxe autre que celle des réseaux de démonstration. L'impossibilité de définir inductivement les démonstrations ne facilite pas l'usage de ce calcul.

Bien que souvent, seule la logique linéaire intuitionniste (plusieurs hypothèses, une seule conclusion) soit effectivement utilisée en informatique, la logique linéaire est aussi une logique classique, c'est-à-dire qu'elle admet une négation  $(\dots)^\perp$  involutive  $((A^\perp)^\perp = A)$  qui échange conjonction et disjonction suivant les lois de de Morgan. Le plongement de la logique linéaire intuitionniste dans la logique linéaire classique, qui permet notamment de voir une implication  $A \multimap B$  comme une disjonction  $A^\perp$  ou  $B$ , est en fait implicite chaque fois qu'on utilise les réseaux de démonstration. Ceux-ci sont plus naturels pour la logique linéaire classique et on obtient la logique linéaire intuitionniste par une simple restriction de langage: il suffit d'imposer aux formules d'être soit des formules écrites avec la conjonction, l'implication et la modalité  $!$ , soit la négation de telles formules.

Un grand avantage de la logique linéaire est qu'elle permet de décrire par des graphes les démonstrations formelles (qui peuvent à leur tour représenter des analyses syntaxiques). Si la démonstration a une conclusion et plusieurs hypothèses (fragment intuitionniste), un tel graphe s'obtient comme suit:

- Construire l'arbre des sous-formules de la conclusion et les arbres des sous-formules de la négation des hypothèses.
- Relier les feuilles (les variables propositionnelles) qui sont la négation l'une de l'autre introduite par l'axiome  $a$  ou  $a^\perp$  — c'est-à-dire  $a \vdash a$ .

Seuls correspondent à des démonstrations les graphes satisfaisant un *critère de correction* global défini par une propriété combinatoire (l'absence d'un certain type de cycles et éventuellement d'autres conditions dépendant du calcul considéré). On pourra consulter [12] pour plus d'informations sur les propriétés combinatoires des réseaux de démonstration.

Qu'apportent ces réseaux de démonstration en regard de formalismes antérieurs comme le calcul des séquents ou la déduction naturelle?

*D'un point de vue logique:* les réseaux de démonstration identifient de nombreuses démonstrations formelles qui ne diffèrent que par l'ordre d'application des règles. Ils permettent

---

[Ret93] C. RETORÉ, *Réseaux et Séquents Ordonnés*, Thèse de Doctorat, spécialité Mathématiques, Université de Paris 7, février 1993.

[Ret97] C. RETORÉ, «Pomset logic: a non-commutative extension of classical linear logic», in : *Typed Lambda Calculus and Applications, TLCA'97*, P. de Groote, J. R. Hindley (éditeurs), Springer-Verlag LNCS, 1210, p. 300–318, 1997.

aussi d'implanter efficacement, en faisant le plus de partage possible, la normalisation des démonstrations. Cette opération, qui transforme une démonstration en une démonstration où toute formule est sous-formule d'une hypothèse ou de la conclusion, est le processus de réduction des  $\lambda$ -termes à la base de l'implantation des langages fonctionnels.

*D'un point de vue grammatical:*

- Les réseaux permettent de définir des grammaires d'inspiration catégorielle plus riches, où le type d'un mot n'est plus une simple formule mais une partie d'un réseau de démonstration; ceci permet de modéliser des constructions syntaxiques assez compliquées: dépendances non bornées, pronoms clitiques, extraction médiane, . . . [8].
- Plutôt que de fonder l'analyse syntaxique du langage sur la construction de démonstrations en calcul des séquents, on peut la fonder sur des algorithmes performants sur des graphes [7] [10].
- Le calcul de la sémantique de Montague, passant par la normalisation des démonstrations, est plus rapide lorsque elles sont représentées par des réseaux [7].
- Les analyses obtenues sous forme de réseaux permettent de rendre compte de propriétés linguistiques importantes du point de vue de l'analyse incrémentale d'une phrase, par exemple la complexité instantanée d'une phrase comportant des relatives imbriquées [Joh98] [Mor98].

## 4 Domaines d'applications

### 4.1 Panorama

**Résumé :** *Les recherches menées dans Paragraphe autour des réseaux de Petri et de leur synthèse ont deux types d'applications que nous étudions conjointement avec le projet Pampa. Le premier type d'applications est la mise en œuvre assistée de systèmes distribués, et en particulier de systèmes de télécommunications. La mise en œuvre de protocoles de communication par répartition semi-automatique d'automates est l'une de ces applications. Une application voisine est la mise en œuvre distribuée de langages de scénarios définis par des HMSCs. Le second type d'applications est la synthèse de contrôleurs asynchrones, étudiée actuellement pour les systèmes de production dans le cadre de l'action coopérative Mars, à laquelle participe aussi le projet Pampa. Dans les deux domaines considérés, des résultats probants ont été obtenus en appliquant l'outil SYNETH à des cas d'étude significatifs (l'outil de B. Caillaud traite actuellement des systèmes à  $\sim 10^4$  états). La distribution du contrôle, qui n'est pas un aspect crucial pour les systèmes de production, est indispensable au contrôle des systèmes de télécommunications. Nous envisageons de*

- 
- [Joh98] M. E. JOHNSON, «Proof Nets and the Complexity of Processing Center-Embedded Constructions», in: *Special Issue on Recent Advances in Logical and Algebraic Approaches to Grammar*, C. Retoré (éditeur), *Journal of Logic Language and Information*, 7(4), Kluwer, p. 433–447, 1998.
- [Mor98] G. MORRILL, «Incremental processing and acceptability», *Research Report n° LSI-98-48-R*, Universitat Politècnica de Catalunya, 1998, To appear in *Computational linguistics*.

*coupler à l'avenir techniques de réalisation distribuée de langages de scénarios et techniques de contrôle asynchrone de systèmes discrets. Le but est d'aller vers une plate-forme intégrée de conception de logiciels distribués, fondée sur la synthèse de réseaux de Petri et sur l'algèbre linéaire, comparable dans l'esprit aux bancs utilisés pour la conception des circuits.*

*Les recherches menées dans Paragraphe autour des grammaires catégorielles et des variantes de la logique linéaire ont des applications potentielles en linguistique théorique et en traitement automatique des langues. L'enjeu dans le premier cas est l'analyse syntaxique concurrente de la langue. L'enjeu dans le second cas est la construction automatique de grammaires à partir de corpus définis par les média.*

#### 4.1.1 Aide à la mise en œuvre de logiciels distribués

La conception de logiciels distribués repose aujourd'hui dans une large mesure sur l'expertise pour adapter et assembler selon les besoins quelques algorithmes bien connus. Allant dans une toute autre direction, nous cherchons à développer des techniques de synthèse permettant d'obtenir sans effort des implémentations distribuées correctes et efficaces de systèmes de communication donnés par des spécifications de service. Ces spécifications de service peuvent prendre la forme d'automates, d'expressions régulières, de langages de scénarios etc... La base commune aux diverses techniques de synthèse que nous proposons est la synthèse de réseaux de Petri, et plus particulièrement la synthèse de réseaux de Petri distribuables telle que la réalise l'outil Synet construit par B. Caillaud dans le projet Pampa. Étant donné un système de transitions étiquetées dans un alphabet distribué, cet outil permet à la fois de décider s'il peut être réalisé par un réseau distribuable et de transformer ce réseau en un ensemble équivalent d'automates 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 [Cai99] une description de la synthèse distribuée du protocole de connexion / déconnexion.

#### 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

---

[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.

[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.

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 prenant la forme de réseaux de Petri. Les places de ces réseaux sont calculées en appliquant le principe des régions [3] aux comportements admissibles de l'installation [RXG00]. Des cas d'études significatifs ont été traités en utilisant Synet, et des contrôleurs compacts ont pu être calculés. La théorie du contrôle asynchrone et distribué reste toutefois essentiellement à construire.

#### 4.1.3 Inférence grammaticale

L'une des applications possibles de nos travaux sur les grammaires catégorielles et le calcul de Lambek est l'inférence grammaticale suivant le paradigme de Gold : étant donné une classe de langages et une suite d'énoncés, trouver une suite de grammaires qui en engendrent les préfixes finis et convergent lorsque l'ensemble des énoncés forme un langage de cette classe vers une grammaire qui engendre exactement ce langage. Pour les grammaires catégorielles de Bar-Hillel, dans lesquelles l'analyse se fait à coups de règles d'élimination, on dispose de bons algorithmes d'inférence grammaticale, fondés sur l'unification des formules logiques (types associés aux mots). Ces algorithmes fondés sur l'unification ont été étendus par R. Bonato [Bon00] aux grammaires catégorielles de Lambek, dont l'analyse utilise aussi des règles d'introduction, mais dans ce contexte l'unification échoue parfois alors que le problème a pourtant une solution. On pourrait y remédier en utilisant non seulement l'unification mais aussi la déduction (dans le calcul de Lambek). On disposera alors d'algorithmes capables de construire automatiquement des grammaires à partir de corpus, par exemple les textes qu'on peut trouver sur le web ! Les mots qui manquent au lexique sont souvent des noms, des adjectifs, des verbes ou des adverbes, et non des mots grammaticaux : ils ont de ce fait un type facile à inférer. Ces algorithmes d'inférence grammaticale pour des classes incluant des langages algébriques sont aussi utilisables dans d'autres domaines comme la génomique.

#### 4.1.4 Analyse syntaxique concurrente

Une application indirecte des travaux que nous menons autour du calcul mixte commutatif / non commutatif de Ph. de Groote se dessine dans le domaine de la linguistique théorique au gré de nos échanges avec E. Stabler. Les grammaires minimalistes qui décrivent des langages contextuels (par exemple  $a^n b^n c^n d^n e^n$ ) ont aujourd'hui des analyseurs syntaxiques à la Earley avec des techniques inspirées des travaux de Pierre Boulier (Projet Atoll). La présentation des grammaires minimalistes et de leurs analyses dans le calcul mixte commutatif n'est pas plus compliquée que leur présentation originale (paragraphe 5.2.3), et le fait que ce même calcul permette de décrire les exécutions concurrentes des réseaux de Petri (paragraphe 5.2.1) permet d'envisager pour ces grammaires des techniques d'analyse concurrente. Bien que la nature

---

[RXG00] N. REZG, X. XIE, A. GHAFARI, «Supervisory Control in Discrete Event Systems Using the Theory of Regions», in : *Proc. Wodes*, Kluwer Academic, 2000.

[Bon00] R. BONATO, *A study on learnability for rigid Lambek grammars*, Tesi di laurea & mémoire de dea, Università di Verona & Université de Rennes 1, 2000.

parallèle des calculs linguistiques ait souvent été affirmée dans les sciences cognitives, ce serait une grande première que de formaliser un mécanisme parallèle pour l'analyse syntaxique.

## 5 Résultats nouveaux

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

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

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

**Résumé :** *Il a été montré que la synthèse de réseaux de Petri peut être appliquée à la mise en œuvre distribuée de langages de scénarios (HMSCs): bien qu'il soit impossible de décider si un HMSC peut être réalisé de façon exacte par un réseau de Petri, on sait calculer la meilleure approximation d'un HMSC par un réseau de Petri [18]. Un document faisant le point de nos acquis sur la synthèse de réseaux de Petri à partir de systèmes de transitions finis et sur ses applications à la distribution a été rédigé [18]. Une étude du contrôle des graphes marqués a été entreprise dans le cadre de l'ARC Mars, visant à comparer les résultats obtenus par l'algèbre et les résultats obtenus par la synthèse de réseaux. Il a par ailleurs été montré que les correspondances de Galois entre réseaux de Petri et automates peuvent être étendues à des automates généralisés [17].*

#### 5.1.1 Distribution d'automates finis via la synthèse de réseaux de Petri bornés

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

Un projet d'article a été rédigé avec B. Caillaud afin de faire le point sur la partie la mieux stabilisée de nos travaux sur la synthèse de réseaux, celle concernant les systèmes de transitions finis et leur distribution [18]. Nous donnons dans ce document deux algorithmes pour la synthèse de réseaux de Petri: un algorithme polynomial utilisant l'optimisation combinatoire, et un algorithme non polynomial utilisant le calcul des rayons extrémaux d'un cône polyédrique. Nous donnons de ces algorithmes deux autres versions adaptées à la synthèse de réseaux distribuables. Nous prouvons la correction de la transformation des réseaux distribuables en automates communicants vis à vis de la bisimulation de branchement, et détaillons plusieurs cas d'études. Ces cas d'études démontrent le succès de la méthode mais montrent aussi les points sur lesquels il reste à travailler pour le conforter. L'un de ces points est l'insertion d'actions silencieuses dans un système de transitions, pour y forcer les deux conditions de séparation et le rendre synthétisable.

#### 5.1.2 Synthèse de réseaux à partir de HMSCs

**Participants :** Philippe Darondeau, Gilles Lesventes.

Les HMSCs (High-level Message Sequence Charts) sont une forme moderne des diagrammes de temps, adaptés à la description de scénarios dans lesquels les sites d'un système distribué

communiquent par messages point à point. Cette forme de spécifications graphiques est définie par une norme de l'ITU (International Telecommunication Union). Un MSC est un diagramme fini qui décrit les relations de cause à effet entre émissions et réceptions de messages constitutives d'un scénario. Les MSCs sont munis d'une opération de concaténation par mise bout à bout de diagrammes, compatible avec la concaténation des séquences sur chaque site. Un HMSC est un automate fini sur un alphabet dont les lettres sont des MSCs. Le langage d'un HMSC est l'ensemble des extensions linéaires des MSCs obtenus en interprétant les chemins de cet automate par concaténation des lettres. Un travail a été mené cette année en coopération avec B. Caillaud et L. Hélouët (projet Pampa) afin d'étudier les langages des HMSCs et les relations de ces langages avec les réseaux de Petri [14]. Les résultats de l'étude sont les suivants. L'inclusion du langage d'un HMSC dans un langage rationnel, l'inclusion inverse, et leur égalité sont des relations indécidables. Le problème de la réalisation exacte du langage d'un HMSC par un réseau de Petri est aussi un problème indécidable. Ces résultats indiquent que la bonne façon de traiter les HMSCs comme des spécifications est de considérer leurs langages comme les comportements minimaux exigés de tout système réalisant ces spécifications. Nous avons montré par ailleurs que les images commutatives des langages de HMSCs sont semi-linéaires, d'où une procédure effective calculant leur fermeture dans les langages de réseaux de Petri distribuables. Nous sommes ainsi à même de construire la réalisation optimale d'un HMSC par un réseau de Petri distribuable, et de la transformer en un système équivalent d'automates communicants.

### 5.1.3 Contrôle des graphes marqués

**Participant** : Philippe Darondeau.

Dans le cadre de l'action coopérative Mars de l'Inria, et indépendamment de l'étude des applications de la théorie des régions au calcul de contrôleurs asynchrones, a été entreprise avec X. Xie l'étude du contrôle optimal des graphes marqués, une classe de réseaux de Petri qui couvre bien les systèmes de production (dans lesquels les problèmes liés à la distribution ne se posent pas vraiment). Des résultats ont été obtenus par des méthodes algébriques ne demandant pas de calculer l'ensemble des marquages. Ces résultats ne permettent pas de garantir la vivacité des systèmes contrôlés, un point sur lequel échoue également la méthode de calcul de contrôleurs (sous-optimaux) fondée sur la théorie des régions.

### 5.1.4 Automates généralisés et réseaux de Petri

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

Dans le cadre de l'action de coopération franco-polonaise «Catalysis» (CNRS/IPIPAN), a été menée avec M. Bednarczyk une étude visant à étendre à de nouvelles formes d'automates les correspondances de Galois entre automates et réseaux de Petri décrites dans [BD95] (correspondance dans le cadre catégorique) et dans [3] (correspondance dans le cadre ordonné). Ont été considérés d'une part des automates dont les transitions sont des fonctions partielles,

---

[BD95] E. BADOUEL, P. DARONDEAU, « Dualities between nets and automata induced by schizophrenic objects », in : *CTCS'95, 953*, Springer Verlag LNCS, p. 24–43, Cambridge, 1995.

dont l'argument peut servir à représenter le temps des systèmes continus ou hybrides, d'autre part des automates dont les transitions sont étiquetées dans une structure algébrique, dont les opérations servent par exemple à représenter la composition séquentielle et la composition parallèle des actions. On a réussi dans [17] à obtenir dans les deux cas une connexion de Galois ordonnée entre automates généralisés et réseaux P/T (avec des règles de mise à feu adaptées). Ceci montre que la théorie des régions, jusque ici développée pour les systèmes de transitions et les langages, est en fait susceptible de jouer un rôle dans un domaine beaucoup plus large.

## 5.2 Réseaux de Petri, logique linéaire

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

**Participants** : Christian Retoré, Yannick Le Nir.

**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. Y. Le Nir a montré que les analyses syntaxiques des grammaires catégorielles, c'est-à-dire les démonstrations à une conclusion et plusieurs hypothèses du calcul de Lambek, peuvent être mises sous forme canonique en opérant d'abord sur chacune des hypothèses de façon séparée dans le calcul de Lambek, et en n'utilisant plus ensuite que les règles d'élimination de ce calcul. Afin de rendre compte logiquement d'autres grammaires apparues plus récemment dans 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 dernières théories linguistiques récentes de Chomsky.*

### 5.2.1 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 d'un gros défaut : les événements sont absents du codage. En utilisant le calcul partiellement commutatif de Ph. de Groote [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 [19]. Les places sont vues comme des variables propositionnelles, et un marquage (partiel) est un produit *commutatif* de places. Un événement 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'événements  $e_i$  et de marquages partiels  $m_j$  entraîne, dans le calcul de [dG96], un marquage partiel  $M$  si et seulement si

---

[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.

les événements  $e_i$ , 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. Ce travail, dont les grandes lignes ont été présentées au Logic Colloquium LC'99 [19], a été approfondi et un projet d'article a été soumis pour publication dans un volume dédié à cette conférence.

### 5.2.2 Grammaires catégorielles, grammaires algébriques et inférence grammaticale

**Participants :** Yannick Le Nir, Christian Retoré.

Le travail effectué porte sur l'analyse des grammaires catégorielles de Lambek [Lam58a] et sur leur lien avec d'autres systèmes comme les grammaires catégorielles AB d'Ajdukiewicz et Bar-Hillel [Ajd35] [BH53] ou les grammaires hors-contexte. Ces grammaires formelles sont principalement utilisées pour le traitement des langues naturelles et de nombreux résultats tant sur leur complexité que sur leur expressivité ont été obtenus ces dernières années [Pen97] [Kan98] [Tie99b].

Une grammaire catégorielle est constituée d'un lexique (application de mots vers des ensembles finis de types) et d'un ensemble de règles fixé par le calcul utilisé (Lambek ou son sous-ensemble AB). Nous proposons une transformation des déductions du calcul de Lambek (L) dans le but d'obtenir des déductions transposables en déductions du calcul AB moyennant une modification des hypothèses (extension du lexique de la grammaire catégorielle). Plus formellement, on obtient une transformation des déductions dans  $L : t_1, \dots, t_n \vdash S$  en déductions dans AB:  $t_1^*, \dots, t_n^* \vdash S$ . Les hypothèses  $t_i$  dans L sont remplacées par des hypothèses  $t_i^*$  dans AB telles que  $t_i \vdash t_i^*$  soit dérivable dans L pour tout  $i \in [1, n]$ . Cette transformation permet d'appliquer certains résultats connus pour les grammaires AB aux grammaires de Lambek (apprentissage par exemple). Ce travail a fait l'objet d'une présentation au workshop "Trees in Logic, Computer Science and Linguistics", ESSLI 2000. La suite des travaux consiste à obtenir maintenant une borne sur la taille des dérivations  $t_i \vdash t_i^*$  indépendante des déductions: les grammaires AB correspondant facilement aux grammaires algébriques, on obtiendrait alors une correspondance simple entre grammaires de Lambek et grammaires algébriques.

- 
- [Lam58a] J. LAMBEK, «The Mathematics of Sentence Structure», *American Mathematical Monthly* 65, 1958, p. 154–170.
- [Ajd35] K. AJDUKIEWICZ, «Die syntaktische Konnexität», *Studia Philosophica* 1, 1935, p. 1–27.
- [BH53] Y. BAR-HILLEL, «A quasi arithmetical notation for syntactic description», *Language* 29, 1953, p. 47–58.
- [Pen97] M. PENTUS, «Product-free Lambek Calculus and Context-Free Grammars», *The Journal of Symbolic Logic* 62, 2, 1997, p. 648–660.
- [Kan98] M. KANAZAWA, *Learnable Classes of Categorical Grammars*, *Studies in Logic, Language and Information*, Center for the Study of Language and Information (CSLI) and The European association for Logic, Language and Information (FOLLI), Stanford, California, 1998.
- [Tie99b] H.-J. TIEDE, *Deductive Systems and Grammars: Proofs as Grammatical Structures*, Mémoire, Indiana University, Berkeley, July 1999.

### 5.2.3 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 [Lam58b] 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 de Chomsky, 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 [9] présente les idées de base. Le second [LR99] étend la correspondance à des mouvements plus compliqués (dits de tête) en utilisant une extension conservative du calcul de Lambek: la logique linéaire intuitionniste partiellement commutative de Ph. de Groote [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 ESSLLI'99. Un numéro spécial du journal «Language and Computation» faisant suite à cette rencontre est en préparation sous la direction de Ch. Retoré et E. Stabler, avec en préface le texte de ces deux auteurs paru comme RR-INRIA 3780 sous le titre «Resource logics and minimalist grammars».

- 
- [Sta97] E. STABLER, «Derivational minimalism», *in: Logical Aspects of Computational Linguistics, LA-CL'96*, 1328, Springer Verlag LNCS/LNAI, 1997.
- [Lam58b] J. LAMBEK, «The mathematics of sentence structure», *American mathematical monthly* 65, 1958.
- [LR99] A. LECOMTE, C. RETORÉ, «A logical formulation of the minimalist program», *in: Third Tbilisi Symposium on Language, Logic and Computation, CSLI*, 1999.
- [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.

### 5.3 Actions nationales

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

**Participants :** Philippe Darondeau, Gilles Lesventes.

L'action coopérative Mars (1999-2000), dirigée par Xiaolan Xie (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 5.1).

### 5.4 Actions européennes

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

Ch. Retoré est le responsable côté français du programme d'échange de doctorants entre les universités de Vérone et de Rennes I.

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

### 5.5 Actions internationales

Ph. Darondeau est le responsable côté français de l'action de coopération franco-polonaise « Catalysis » (CNRS/IPIPAN). Les premiers résultats de cette coopération montrent comment construire des correspondances de Galois entre réseaux de Petri et automates généralisés [17] (voir 5.1).

### 5.6 Visites et invitations de chercheurs

Ch. Retoré a accueilli R. Bonato (Université de Vérone) pendant cinq mois, dans le cadre d'un échange Erasmus, en vue de la préparation d'un mémoire de « Tesi di Laurea » sur l'apprentissage des grammaires catégorielles.

Y. Le Nir a séjourné pendant quinze jours en décembre 1999 dans l'équipe de M. Abrusci à l'université de Roma Tre (sur invitation dans le cadre de l'action intégrée Galileo "Logiques linéaires non commutatives pour la syntaxe des langues romanes").

Ph. Darondeau a reçu puis a rendu visite (en deux périodes de quinze jours) à M. Bednarczyk (division de Gdansk de l'académie des sciences de Pologne) dans le cadre de l'action de coopération franco-polonaise « Catalysis » (CNRS/IPIPAN).

## 6 Diffusion de résultats

### 6.1 Animation de la Communauté scientifique

Ch. Retoré a organisé avec S. Pinchinat un nouveau séminaire hebdomadaire à l'Irisa, intitulé 68NQRT, regroupant les thèmes Génie Logiciel, Informatique Théorique, Combinatoire et Intelligence Artificielle.

Ch. Retoré a organisé avec A. Foret une rencontre intitulée *Grammar and Logic: natural language analysis, generation and learning*, regroupant à l'Irisa les participants de trois actions intégrées (avec l'Italie, les Pays-Bas et l'Allemagne). L'édition des actes est en cours (avec A. Dikovsky).

Ch. Retoré est membre du comité de sélection de la meilleure thèse de la Foundation of Logic, Language and Information (sur ces thèmes).

Ch. Retoré a été membre du comité de programme de la conférence *Tree Adjoining Grammars and related formalisms* (TAG+5), tenue à Paris en mai 2000.

Ph. Darondeau a été membre du comité de programme de la conférence AMAST'2000, tenue à Iowa City en mai 2000.

Ph. Darondeau a été membre du comité de programme de la conférence CONCUR'2000, tenue à Penn State University en août 2000.

### 6.2 Enseignement

Ch. Retoré a donné à l'ESSLLI (European Summer School in Logic Language and Information) un cours de 8 heures intitulé "The logic of categorial grammars" (avec distribution d'un photocopié de 70 pages).

### 6.3 Rapports de thèse

Ph. Darondeau a été rapporteur de la thèse de P. Baldan, intitulée «Modelling Concurrent Computations: from Contextual Petri Nets to Graph Grammars», soutenue en mars 2000 à Pise.

Ph. Darondeau a été l'External Examiner de la thèse de M. Pietkiewicz-Koutny, intitulée «Relating Formal Models of Concurrency for the Design of Asynchronous Digital Hardware», soutenue en septembre 2000 à Newcastle upon Tyne.

### 6.4 Séminaires

Ch. Retoré a présenté les grammaires catégorielles et la logique linéaire au séminaire TALC à Nantes.

Y. Le Nir a présenté un exposé intitulé «From Lambek grammars to AB-Grammars» aux rencontres *Grammar and Logic: natural language analysis, generation and learning* organisées à l'Irisa en mai 2000.

Y. Le Nir a présenté un exposé intitulé «From Lambek deduction trees to Adjukiewicz Bar-Hillel residuation binary-trees» au *Workshop on Trees in Logic, Computer Science, and Linguistics* organisé à Birmingham dans le cadre d'ESSLLI en août 2000.

Ph. Darondeau a présenté un exposé intitulé «Realizing Partial Specifications: Perspectives in Petri Net Synthesis» au meeting annuel du WG2.2 de l'Ifip à Oldenburg en septembre 2000.

## 7 Bibliographie

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

- [1] E. BADOUEL, L. BERNARDINELLO, P. DARONDEAU, «The synthesis problem for elementary net systems is NP-complete», *TCS 186*, 1997, p. 107–134.
- [2] E. BADOUEL, P. DARONDEAU, J.-C. RAOULT, «Context-free event domains are recognizable», *Information and Computation*, 149, 1999, p. 134–172.
- [3] E. BADOUEL, P. DARONDEAU, «Theory of regions», in : *Lectures on Petri Nets I: Basic Models, 1491*, Springer-Verlag LNCS, 1999, p. 529–586.
- [4] 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.
- [5] D. BECHET, P. DE GROOTE, C. RETORÉ, «A complete axiomatisation of the inclusion of series-parallel partial orders», in : *Rewriting Techniques and Applications, RTA'97*, H. Comon (éditeur), 1232, Springer Verlag LNCS, p. 230–240, 1997.
- [6] P. DARONDEAU, «Deriving Unbounded Petri Nets from Formal Languages», in : *Concur'98, 1466*, Springer-Verlag LNCS, p. 533–548, Nice, 1998.
- [7] P. DE GROOTE, C. RETORÉ, «Semantic readings of proof nets», in : *Formal Grammar*, G.-J. Kruijff, G. Morrill, D. Oehrle (éditeurs), FoLLI, p. 57–70, Prague, août 1996.
- [8] A. LECOMTE, C. RETORÉ, «Words as Modules: a lexicalised grammar in the framework of linear logic proof nets», in : *Mathematical and Computational Analysis of Natural Language — selected papers from ICML'96*, C. Martin-Vide (éditeur), *Studies in Functional and Structural Linguistics*, 45, John Benjamins Publishing Company, p. 129–144, 1998.
- [9] 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.
- [10] C. RETORÉ, «Calcul de Lambek et logique linéaire», *Traitement Automatique des Langues 37*, 2, 1996, p. 39–70.

### Articles et chapitres de livre

- [11] P. DARONDEAU, «On the Petri Net Realization of Context-Free Graphs», *Theoretical Computer Science*, 1999, référence 3643, à paraître.
- [12] C. RETORÉ, «Handsome proof-nets: perfect matchings and cographs», *Theoretical Computer Science*, 1999, à paraître.
- [13] C. RETORÉ, «Systèmes déductifs et traitement des langues: un panorama des grammaires catégorielles», *Technique et science informatiques*, 2000, À paraître.

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

- [14] B. CAILLAUD, P. DARONDEAU, L. HÉLOUËT, G. LESVENTES, «HMSCs as partial specifications ... with PNs as completions», *in* : *MOVEP'2K*, p. 87–103, 2000. (texte du INRIA-RR 3970, version révisée à paraître chez Springer).
- [15] P. DARONDEAU, «Region Based Synthesis of P/T-Nets and its Potential Applications», *in* : *ICATPN'2000, 1825*, Springer-Verlag LNCS, p. 16–23, 2000.

**Rapports de recherche et publications internes**

- [16] C. RETORÉ, «The logic of categorial grammars», *Lecture notes*, ESSLLI 2000, 2000, disponible sur the web.

**Divers**

- [17] E. BADOUEL, M. BEDNARCZYK, P. DARONDEAU, «Generalized automata and their net representations», 2000, manuscrit.
- [18] E. BADOUEL, B. CAILLAUD, P. DARONDEAU, «Distributing finite automata through Petri net synthesis», 2000, manuscrit.
- [19] C. RETORÉ, «A representation of Petri net execution in partially commutative linear logic», 1999, texte présenté au Logic Colloquium LC'99.