Philippe Blache
Edward Stabler
Joan Busquets
Richard Moot (Eds.)

Logical Aspects of Computational Linguistics

LNAI 3492

5th International Conference, LACL 2005
Bordeaux, France, April 2005
Proceedings

Springer

Philippe Blache Edward Stabler
Joan Busquets Richard Moot (Eds.)

Logical Aspects of Computational Linguistics

5th International Conference, LACL 2005
Bordeaux, France, April 28-30, 2005
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Philippe Blache
Université de Provence
CNRS, Laboratoire Parole et Langage
29, Avenue Robert Schuman, 13621 Aix-en-Provence, France
E-mail: pb@lpl.univ-aix.fr

Edward Stabler
UCLA Department of Linguistics
3125 Campbell Hall, Box 951543, Los Angeles, CA 90095-1542, USA
E-mail: stabler@ucla.edu

Joan Busquets
Université Bordeaux 3
CNRS, ERSS
Domaine Universitaire, 33607 Pessac Cedex, France
E-mail: busquets@u-bordeaux3.fr

Richard Moot
Université Bordeaux 1
CNRS, LaBRI
351, Cours de la Libération, 33405 Talence Cedex, France
E-mail: Richard.Moot@labri.fr

Library of Congress Control Number: 2005924437

CR Subject Classification (1998): I.2, F.4.1

ISSN 0302-9743
ISBN-10 3-540-25783-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25783-7 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11422532 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3492

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

FoLLI Publications on Logic, Language and Information

Preface

This volume contains the proceedings of the 5th International Conference on Logical Aspects of Computational Linguistics held April 28–30, 2005 in Bordeaux, France. This proceedings contains papers on a wide range of logical and formal methods in computational linguistics, with studies of particular grammar formalisms (Categorial Grammars, TAG, Dependency Grammars or Minimalist Grammars) and their computational properties (complexity, determinism, unification), language engineering (grammar development, parsing, translation) and traditional questions about the syntax/semantics interface. Formal aspects are moreover assessed with actual linguistic data from different languages (English, French, Arabic), which is the sign of a maturing field.

This text, as well as the conference itself, is then the occasion to bring together people coming from different horizons: logicians, linguists, computational scientists. This fits perfectly well with the mission of FoLLI, the Association of Logic, Language and Information, and so this textbook inaugurates the new FoLLI/LNAI series.

The Program Committee faced a difficult task because we received many submissions (40% rejected), and the reviewing task had to be done quickly. We thank all our reviewers and especially those who had to be recruited at the last minute. We also thank very much the Organizing Committee. Bordeaux, known as a wine capital, is now becoming a research center in this field.

We would like to thank all the people who made this 5th LACL possible: the Program Committee, the external reviewers, the Organizing Committee, and the LACL sponsors. Last, but not least, very special thanks to Christian Retoré who has been, since the very beginning, the heart of LACL. This conference, and these books, simply would not exist without him.

April 2005 Philippe Blache and Edward Stabler

Organization

LACL 2005 was organized by INRIA, Université Bordeaux 1, and Université Bordeaux 3.

Program Committee

Conference Chairs	Philippe Blache (Université de Provence, France) Edward Stabler (University of California, USA) Joan Busquets (ERSS and INRIA) Richard Moot (LABRI and INRIA)
Organizing Chairs	Joan Busquets (ERSS and INRIA) Richard Moot (LABRI and INRIA)

Referees

D. Aguilar-Solis
J.-M. Andreoli
P. Blache
J. Bos
J. Busquets
H. Christiansen
V. Dahl
P. de Groote
M. de Rijke
D. Duchier
M. Dymetman
K. Gerdes
F. Hamm
G. Holst
G. Jaeger
S. Kepser
G. Kobele
M. Kracht
A. Lecomte
U. Moennich
M. Moortgat
R. Moot
G. Morrill
D. Oehrle
J.-P. Prost
C. Retoré
F. Richter
J. Rogers
V. Shanker
E. Stabler
M. Steedman

Organizing Committee

Maxime Amblard	(Université Bordeaux 1 and LaBRI-CNRS and INRIA)
Philippe Biais	(CNRS LaBRI)
Karine Cabandé	(CNRS DR Aquitaine et Poitou-Charentes)
Catherine Girard	(INRIA-Futurs)
Patrick Henry	(CNRS LABRI and INRIA-Futurs)
Brigitte Larue-Bourdon	(INRIA-Futurs)
Jean-Louis Lassartesses	(Université Bordeaux 1)
Annie Nadeau	(CNRS DR Aquitaine et Poitou-Charentes)
Christian Retoré	(Université Bordeaux 1 and LaBRI-CNRS and INRIA)

Sponsoring Institutions

CNRS
INRIA
Université Bordeaux 1
Université Bordeaux 3
Pôle Universitaire de Bordeaux
Conseil Régional d'Aquitaine
France Télécom
ERSS
CoLogNet

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3492: P. Blache, E. Stabler (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3419: B. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Constraint Satisfaction and Constraint Logic Programming. X, 217 pages. 2005.

Vol. 3416: M. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005.

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005.

Vol. 3398: D.-K. Baik (Ed.), Systems Modeling and Simulation: Theory and Applications. XIV, 733 pages. 2005.

Vol. 3397: T.G. Kim (Ed.), Artificial Intelligence and Simulation. XV, 711 pages. 2005.

Vol. 3396: R.M. van Eijk, M.-P. Huget, F. Dignum (Eds.), Agent Communication. X, 261 pages. 2005.

Vol. 3394: D. Kudenko, D. Kazakov, E. Alonso (Eds.), Adaptive Agents and Multi-Agent Systems II. VIII, 313 pages. 2005.

Vol. 3392: D. Seipel, M. Hanus, U. Geske, O. Bartenstein (Eds.), Applications of Declarative Programming and Knowledge Management. X, 309 pages. 2005.

Vol. 3374: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems. X, 279 pages. 2005.

Vol. 3371: M.W. Barley, N. Kasabov (Eds.), Intelligent Agents and Multi-Agent Systems. X, 329 pages. 2005.

Vol. 3369: V.R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.), Law and the Semantic Web. XII, 249 pages. 2005.

Vol. 3366: I. Rahwan, P. Moraitis, C. Reed (Eds.), Argumentation in Multi-Agent Systems. XII, 263 pages. 2005.

Vol. 3359: G. Grieser, Y. Tanaka (Eds.), Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets. XIV, 257 pages. 2005.

Vol. 3346: R.H. Bordini, M. Dastani, J. Dix, A.E.F. Seghrouchni (Eds.), Programming Multi-Agent Systems. XIV, 249 pages. 2005.

Vol. 3345: Y. Cai (Ed.), Ambient Intelligence for Scientific Discovery. XII, 311 pages. 2005.

Vol. 3343: C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, T. Barkowsky (Eds.), Spatial Cognition IV. XIII, 519 pages. 2005.

Vol. 3339: G.I. Webb, X. Yu (Eds.), AI 2004: Advances in Artificial Intelligence. XXII, 1272 pages. 2004.

Vol. 3336: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. X, 523 pages. 2004.

Vol. 3327: Y. Shi, W. Xu, Z. Chen (Eds.), Data Mining and Knowledge Management. XIII, 263 pages. 2005.

Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004. XX, 987 pages. 2004.

Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004.

Vol. 3276: D. Nardi, M. Riedmiller, C. Sammut, J. Santos-Victor (Eds.), RoboCup 2004: Robot Soccer World Cup VIII. XVIII, 678 pages. 2005.

Vol. 3275: P. Perner (Ed.), Advances in Data Mining. VIII, 173 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004.

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004.

Vol. 3259: J. Dix, J. Leite (Eds.), Computational Logic in Multi-Agent Systems. XII, 251 pages. 2004.

Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004.

Vol. 3248: K.-Y. Su, J. Tsujii, J.-H. Lee, O.Y. Kwong (Eds.), Natural Language Processing – IJCNLP 2004. XVIII, 817 pages. 2005.

Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004.

Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004.

Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), Advances in Natural Language Processing. XII, 488 pages. 2004.

Vol. 3229: J.J. Alferes, J. Leite (Eds.), Logics in Artificial Intelligence. XIV, 744 pages. 2004.

Vol. 3228: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff (Eds.), Formal Approaches to Agent-Based Systems. VIII, 290 pages. 2004.

Vol. 3215: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LVII, 906 pages. 2004.

Vol. 3214: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LVIII, 1302 pages. 2004.

Vol. 3213: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LVIII, 1280 pages. 2004.

Vol. 3209: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web. IX, 201 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004.

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004.

Vol. 3192: C. Bussler, D. Fensel (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 522 pages. 2004.

Vol. 3191: M. Klusch, S. Ossowski, V. Kashyap, R. Unland (Eds.), Cooperative Information Agents VIII. XI, 303 pages. 2004.

Vol. 3187: G. Lindemann, J. Denzinger, I.J. Timm, R. Unland (Eds.), Multiagent System Technologies. XIII, 341 pages. 2004.

Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning. IX, 241 pages. 2004.

Vol. 3171: A.L.C. Bazzan, S. Labidi (Eds.), Advances in Artificial Intelligence – SBIA 2004. XVII, 548 pages. 2004.

Vol. 3159: U. Visser, Intelligent Information Integration for the Semantic Web. XIV, 150 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advances in Case-Based Reasoning. XIII, 822 pages. 2004.

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004.

Vol. 3127: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.), Conceptual Structures at Work. XI, 403 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004.

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004.

Vol. 3097: D. Basin, M. Rusinowitch (Eds.), Automated Reasoning. XII, 493 pages. 2004.

Vol. 3071: A. Omicini, P. Petta, J. Pitt (Eds.), Engineering Societies in the Agents World. XIII, 409 pages. 2004.

Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004. XXV, 1208 pages. 2004.

Vol. 3068: E. André, L. Dybkjær, W. Minker, P. Heisterkamp (Eds.), Affective Dialogue Systems. XII, 324 pages. 2004.

Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), Programming Multi-Agent Systems. X, 221 pages. 2004.

Vol. 3066: S. Tsumoto, R. Słowiński, J. Komorowski, J.W. Grzymała-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004.

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic in Computer Science. X, 275 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004.

Vol. 3055: H. Christiansen, M.-S. Hacid, T. Andreasen, H.L. Larsen (Eds.), Flexible Query Answering Systems. X, 500 pages. 2004.

Vol. 3048: P. Faratin, D.C. Parkes, J.A. Rodríguez-Aguilar, W.E. Walsh (Eds.), Agent-Mediated Electronic Commerce V. XI, 155 pages. 2004.

Vol. 3040: R. Conejo, M. Urretavizcaya, J.-L. Pérez-de-la-Cruz (Eds.), Current Topics in Artificial Intelligence. XIV, 689 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004.

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004.

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004.

Vol. 3020: D. Polani, B. Browning, A. Bonarini, K. Yoshida (Eds.), RoboCup 2003: Robot Soccer World Cup VII. XVI, 767 pages. 2004.

Vol. 3012: K. Kurumatani, S.-H. Chen, A. Ohuchi (Eds.), Multi-Agents for Mass User Support. X, 217 pages. 2004.

Vol. 3010: K.R. Apt, F. Fages, F. Rossi, P. Szeredi, J. Váncza (Eds.), Recent Advances in Constraints. VIII, 285 pages. 2004.

Vol. 2990: J. Leite, A. Omicini, L. Sterling, P. Torroni (Eds.), Declarative Agent Languages and Technologies. XII, 281 pages. 2004.

Vol. 2980: A. Blackwell, K. Marriott, A. Shimojima (Eds.), Diagrammatic Representation and Inference. XV, 448 pages. 2004.

Vol. 2977: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (Eds.), Engineering Self-Organising Systems. X, 299 pages. 2004.

¥490.88元

Table of Contents

LACL

k-Valued Non-associative Lambek Grammars (Without Product) Form a Strict Hierarchy of Languages

Denis Béchet[1] and Annie Foret[2]

[1] LINA – FRE 2729, Université de Nantes & CNRS
2, rue de la Houssinière — BP 92208, 44322 Nantes Cedex 03, France
Denis.Bechet@univ-nantes.fr
[2] IRISA – Université de Rennes 1, Campus Universitaire de Beaulieu,
Avenue du Général Leclerc, 35042 Rennes Cedex, France
Annie.Foret@irisa.fr

Abstract. The notion of k-valued categorial grammars where a word is associated to at most k types is often used in the field of lexicalized grammars as a fruitful constraint for obtaining several properties like the existence of learning algorithms. This principle is relevant only when the classes of k-valued grammars correspond to a real hierarchy of languages. This paper establishes the relevance of this notion for two related grammatical systems. In the first part, the classes of k-valued non-associative Lambek (*NL*) grammars without product is proved to define a strict hierarchy of languages. The second part introduces the notion of generalized functor argument for non-associative Lambek ($NL_\emptyset$) calculus without product but allowing empty antecedent and establishes also that the classes of k-valued $NL_\emptyset$ grammars without product form a strict hierarchy of languages.

1 Introduction

The field of natural language processing includes lexicalized grammars such as classical categorial grammars [1], the different variants of Lambek calculus [2], lexicalized tree adjoining grammars [3], etc. In these lexicalized formalisms, a k-valued grammar associates at most k categories to each word of the lexicon. For a particular model of lexicalized grammars and their corresponding languages, this definition forms a (strict) hierarchy of classes of grammars when k increases. This hierarchy of grammars corresponds to a growing list of classes of languages that does not necessarily form a strict hierarchy.

In fact, in the field of lexicalized grammars, the concept of k-valued grammars is often used to define sub-classes of grammars and languages that satify some property when the whole class does not satify it. In particular, this notion is important for a lot of learnability results in Gold's model [4]. Usually, to find a positive result, some kind of restriction is necessary because very often the whole

P. Blache et al. (Eds.): LACL 2005, LNAI 3492, pp. 1–17, 2005.

class of grammars corresponds to (at least) context free languages and this class is known to be unlearnable no matter which grammatical system is used[1].

Usually, when the learnability can be established for a particular k-valued class of grammars, this property can also be proved for any $k \in \mathbb{N}$. When this assumption is true, for each $k \in \mathbb{N}$, there exists a learning algorithm that learns the grammars of this class from positive examples even if the whole class is not learnable (there does not exist a universal learning algorithm for all $k \in \mathbb{N}$): we hope that all the interesting grammars (the grammars for natural languages, for instance) are in a particular k-valued class.

In this context, we can wonder if, for a particular system of lexicalized grammars and languages, the class of classes of languages forms a strict hierarchy. If this assumption is not verified for a system, it usually means that for a certain k the class of k-valued languages is the same as the whole class: the hierarchy is truncated. The other possibility which is very rare consists in a hierarchy that has infinite steps, some steps corresponding to several contiguous integers. In this context, the proof that the hierarchy is not strict is usually a bad news for the concept of k-valued grammars corresponding to a system. For instance, the classes of k-valued classical categorial grammar form a strict hierarchy of languages [5] and for each $k \in \mathbb{N}$, the class of k-valued classical categorial grammar is learnable from positive examples.

In the paper, we are interested to prove that non-associative Lambek grammars without product (written *NL*) and non-associative Lambek grammars without product but with empty antecedent (written $NL_\emptyset$) form two strict hierarchies of classes of languages. The results give a direct justification of the notion of k-valued grammars for both systems. The paper also recalls a useful alternative presentation of *NL* using generalized AB deductions (written *GAB*) and generalized functor-argument structures (written *FA*) that were used in [6] for defining a learning algorithm but are a central notion in the paper for proving the strict hierarchy. For $NL_\emptyset$, the paper introduces similar notions: generalized AB deductions with empty applications (written $GAB_\emptyset$) and generalized functor-argument structures with empty arguments (written $FA_\emptyset$). This presentation is also proved, in the paper, to be equivalent to $NL_\emptyset$ and the hierarchy theorem is given.

The paper is organized as follows. Section 2 gives some background knowledge on non-associative Lambek categorial grammars (without product). Section 3 focuses on (recently defined) structures, with alternative deduction rules for NL-grammars (without product) as generalized FA-structures; in fact these rules are extensions of the cancellation rules of classical categorial grammars that lead to the generalization of FA-structures used here. Section 4 presents the proof that the class of k-valued non-associative Lambek categorial grammars without product form a strict hierarchy. Section 5 considers the $NL_\emptyset$ variant of NL ; we

---

[1] The class of context free language has a limit point: $\exists L, (L_i)_{i\in\mathbb{N}}$ such that $L = \bigcup_{i\in\mathbb{N}} L_i$ and $\forall i \in \mathbb{N}\ L_i \subsetneq L_{i+1}$. This property entails that any class of grammars that corresponds to (a superclass of) context free languages (of strings) is not learnable from positive examples.

define generalized functor-argument structures for this variant and a strict hierarchy theorem; we also revisit the case of AB (classical categorial grammars). Section 6 concludes.

2 Background

2.1 Categorial Grammars

The reader not familiar with Lambek Calculus and its non-associative version will find nice presentation in the first articles written by Lambek [2, 7] or more recently in [8, 9, 10, 11, 12, 13]. We use in the paper two variants of Lambek calculus: non-associative Lambek calculus without product with ($NL_\emptyset$) and without (NL) empty antecedent.

Definition 1 (Types). *The* types *Tp, or formulas, are generated from a set of* primitive types *Pr, or atomic formulas, by two binary connectives*[2] *"/" (over) and "\" (under):*

$$Tp ::= Pr \mid Tp\backslash Tp \mid Tp/Tp$$

Definition 2 (Rigid and k-valued categorial grammars). *A* categorial grammar *is a structure $G = (\Sigma, I, S)$ where:*

- *Σ is a finite alphabet (the words in the sentences);*
- *$I : \Sigma \mapsto \mathcal{P}^f(Tp)$ is a function (called a lexicon) that maps a finite set of types to each element of Σ (the possible categories of each word);*
- *$S \in Pr$ is the* main type *associated to correct sentences.*

If $X \in I(a)$, we say that G associates X to a and we write $G : a \mapsto X$.

A k-valued categorial grammar *is a categorial grammar where, for every word $a \in \Sigma$, $I(a)$ has at most k elements. A* rigid categorial grammar *is a 1-valued categorial grammar.*

2.2 Non-associative Lambek Calculi NL and $NL_\emptyset$

$NL/NL_\emptyset$ Derivation $\vdash_{NL}/\vdash_{NL_\emptyset}$. As a logical system, we use Gentzen-style sequent presentation. A sequent $\Gamma \vdash A$ is composed of a binary tree of formulas Γ (the set of such trees is noted $\mathcal{T}_{Tp}$) which is the antecedent configuration and a succedent formula A. A context $\Gamma[\cdot]$ is a binary tree of formulas with a hole. For X, a formula or a binary tree of formulas, $\Gamma[X]$ is the binary tree obtained from $\Gamma[\cdot]$ by filling the hole with X. Γ can be empty in $NL_\emptyset$: Γ belongs to $\mathcal{T}_{Tp} \cup \{\emptyset\}$. In fact, to completely define the rules of $NL_\emptyset$, we use two equivalence relations on binary trees of formulas and the empty set[3]:

$$\Gamma[(\emptyset, \Delta)] \equiv_{NL_\emptyset} \Gamma[\Delta] \equiv_{NL_\emptyset} \Gamma[(\Delta, \emptyset)]$$

---

[2] No product connective is used in the paper.

[3] One can define $NL_\emptyset$ without the two equivalence relations by specializing the rules when one of the antecedents is empty but this gives a much longer definition.

Definition 3 (*NL/NL*$_\emptyset$)**.** *A sequent is valid in NL/NL*$_\emptyset$ *and is noted* $\Gamma \vdash_{NL} A / \Gamma \vdash_{NL_\emptyset} A$ *iff* $\Gamma \vdash A$ *can be deduced from the following rules:*

$$\frac{}{A \vdash A}\,\mathbf{Ax} \qquad \frac{(\Gamma, B) \vdash A}{\Gamma \vdash A/B}\,/R \qquad \frac{(A, \Gamma) \vdash B}{\Gamma \vdash A\backslash B}\,\backslash R$$

$$\frac{\Gamma \vdash A \quad \Delta[A] \vdash B}{\Delta[\Gamma] \vdash B}\,\mathbf{Cut} \quad \frac{\Gamma \vdash A \quad \Delta[B] \vdash C}{\Delta[(B/A, \Gamma)] \vdash C}\,/L \quad \frac{\Gamma \vdash A \quad \Delta[B] \vdash C}{\Delta[(\Gamma, A\backslash B)] \vdash C}\,\backslash L$$

Cut Elimination. We recall that the cut rule can be eliminated in $\vdash_{NL}$ and in $\vdash_{NL_\emptyset}$: every derivable sequent has a cut-free derivation.

NL/NL$_\emptyset$ **Languages.** $\mathcal{E}^+$ denotes the set of non-empty strings over $\mathcal{E}$. $\mathcal{T}_\mathcal{E}$ is the set of (non-empty) well-bracketed lists (binary trees) of elements of $\mathcal{E}$.

Definition 4 (Yield). *If* T *is a tree where the leaves are elements of a set* $\mathcal{E}$, $yield_\mathcal{E}(T) \in \mathcal{E}^+$ *is the list of leaves of* T.

This notation will be used for well-bracketed lists of words $yield_\Sigma$, for binary trees of formulas $yield_{Tp}$ and will be extended to *FA* structures (see further Definition 7).

Definition 5 (Language). *Let* $G = (\Sigma, I, S)$ *be a categorial grammar.*

- G generates a well-bracketed list of words $T \in \mathcal{T}_\Sigma$ *(in NL/NL*$_\emptyset$ *model) iff there exists* Γ *a binary tree of types,* $c_1, \ldots, c_n \in \Sigma$ *and* $A_1, \ldots, A_n \in Tp$ *such that:*

$$\begin{cases} G : c_i \mapsto A_i \ (1 \le i \le n) \\ \Gamma = T[c_1 \to A_1, \ldots, c_n \to A_n] \\ \Gamma \vdash_{NL} S / \Gamma \vdash_{NL_\emptyset} S \end{cases}$$

  where $T[c_1 \to A_1, \ldots, c_n \to A_n]$ *means the binary tree obtained from* T *by substituting the left to right occurrences of* $c_1, \ldots, c_n$ *by* $A_1, \ldots, A_n$.
- G generates a string $c_1 \cdots c_n \in \Sigma^+$ *iff there exists* $T \in \mathcal{T}_\Sigma$ *such that* $yield_\Sigma(T) = c_1 \cdots c_n$ *and* G *generates* T.
- *The* language of well-bracketed lists of words *of* G, *written* $\mathcal{BL}_{NL}(G) / \mathcal{BL}_{NL_\emptyset}(G)$, *is the set of well-bracketed lists of words generated by* G.
- *The* language of strings *corresponding to* G, *written* $\mathcal{L}_{NL}(G)/\mathcal{L}_{NL_\emptyset}(G)$, *is the set of strings generated by* G.

One interest of *NL* when compared to classical categorial grammars lies in its possibility to easily encode a restriction on the use of a basic category. For instance when we want to distinguish between a noun phrase and pronouns in subject position or object position, we can proceed as follows.

Example 1. Let $\Sigma_1 = \{John, Mary, likes, he, she, him, her\}$ and let $Pr_1 = \{S, N, X_1, X_2\}$. We define the following rigid grammar:

$$G_1 = \begin{cases} John, Mary \mapsto N \ ; he, she \mapsto N_1; \\ him, her \quad \mapsto N_2 \ ; likes \quad \mapsto N_1\backslash(S/N_2). \end{cases}$$

where $N_1 = X_1/(N\backslash X_1)$ and $N_2 = X_2/(N\backslash X_2)$.

We get: $((He\ likes)\ Mary) \in \mathcal{BL}_{NL}(G_1)$ but: $John\ likes\ she \notin \mathcal{L}_{NL}(G_1)$.

With $NL_\emptyset$, we can introduce a restrictive form of optional arguments; "little", in the following example, is optionally associated to proper nouns. This is not possible directly with *NL*.

Example 2. Let $\Sigma_2 = \{John, Mary, likes, little\}$ and let $Pr_2 = \{S, N, L\}$.

We define: $G_2 = \begin{cases} John \mapsto (L\backslash L)\backslash N \ ; Mary \mapsto (L\backslash L)\backslash N \\ little \mapsto L\backslash L \quad ; likes \ \mapsto N\backslash(S/N) \end{cases}$

G_2 is a rigid (or 1-valued) grammar. We can prove that $(((L\backslash L)\backslash N,\ N\backslash(S/N)),\ (L\backslash L)\backslash N) \vdash_{NL_\emptyset} S$ and $(((L\backslash L)\backslash N,\ N\backslash(S/N)),\ (L\backslash L, (L\backslash L)\backslash N)) \vdash_{NL_\emptyset} S$. Thus, we get:

$$John\ likes\ Mary \ \in \mathcal{L}_{NL_\emptyset}(G_2) \ ; \quad John\ likes\ little\ Mary \ \in \mathcal{L}_{NL_\emptyset}(G_2)$$
$$((John\ likes)\ Mary) \in \mathcal{BL}_{NL_\emptyset}(G_2) \ ; \ ((John\ likes)\ (little\ Mary)) \in \mathcal{BL}_{NL_\emptyset}(G_2)$$

2.3 Some Useful Models

Powerset Residuated Groupoids and Semi-groups. Let $(M, .)$ be a groupoid. Let $\mathcal{P}(M)$ denote the powerset of M. A *powerset residuated groupoid* over $(M, .)$ is the structure $(\mathcal{P}(M), \circ, \Rightarrow, \Leftarrow, \subseteq)$ such that for $X, Y \subseteq M$:

$$\begin{aligned} X \circ Y &= \{x.y : x \in X, y \in Y\} \\ X \Rightarrow Y &= \{y \in M : (\forall x \in X)\ x.y \in Y\} \\ Y \Leftarrow X &= \{y \in M : (\forall x \in X)\ y.x \in Y\} \end{aligned}$$

If $(M, .)$ has a unit I (that is : $\forall x \in M : I.x = x.I = x$), then the above structure is a *powerset residuated groupoid with unit* (it has $\{I\}$ as unit).

Interpretation. Given a powerset residuated groupoid $(\mathcal{P}(M), \circ, \Rightarrow, \Leftarrow, \subseteq)$, an *interpretation* is a map from primitive types p to elements $[\![p]\!]$ in $\mathcal{P}(M)$ that is extended to types and sequences in the natural way :

$$[\![C_1\backslash C_2]\!] = [\![C_1]\!] \Rightarrow [\![C_2]\!] \ ; [\![C_1/C_2]\!] = [\![C_1]\!] \Leftarrow [\![C_2]\!] \ ; [\![(C_1, C_2)]\!] = ([\![C_1]\!] \circ [\![C_2]\!])$$

By a model property for *NL* : If $\Gamma \vdash_{NL} C$ then $[\![\Gamma]\!] \subseteq [\![C]\!]$
If $(M, .)$ is a groupoid with a unit I, we add $[\![\emptyset]\!] = \{I\}$ for the empty sequence $\emptyset$ and get a model property for $NL_\emptyset$: if $\Gamma \vdash_{NL_\emptyset} C$ then $[\![\Gamma]\!] \subseteq [\![C]\!]$.

3 GAB Deductions and Generalized FA-Structures

In this section we focus on (recently defined) structures [6], with alternative deduction rules for NL-grammars (without product) as generalized FA-structures; in fact these rules are extensions of the cancellation rules of classical categorial grammars that lead to the generalization of FA-structures used here.

3.1 *FA* Structures over a Set $\mathcal{E}$

We give a general definition of *FA* structures over a set $\mathcal{E}$, whereas in practice $\mathcal{E}$ is either an alphabet Σ or a set of types such as Tp.

Definition 6 (*FA* structures). *Let $\mathcal{E}$ be a set, a FA* structure *over $\mathcal{E}$ is a binary tree where each leaf is labelled by an element of $\mathcal{E}$ and each internal node is labelled by FApp (forward application) or BApp (backward application):*

$$\mathcal{FA}_{\mathcal{E}} ::= \mathcal{E} \mid FApp(\mathcal{FA}_{\mathcal{E}}, \mathcal{FA}_{\mathcal{E}}) \mid BApp(\mathcal{FA}_{\mathcal{E}}, \mathcal{FA}_{\mathcal{E}})$$

Definition 7 (Tree yield). *The well-bracketed list of words obtained from a FA structure F over $\mathcal{E}$ by forgetting FApp and BApp labels is called the* tree yield *of F over $\mathcal{E}$ (notation $tree_{\mathcal{E}}(F)$).*

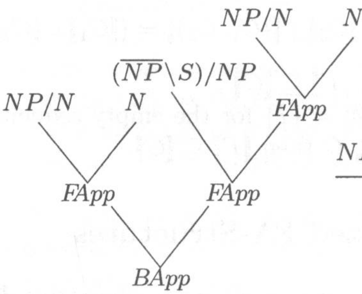
3.2 *GAB* Deductions

Definition 8 (*GAB* Deduction). Generalized *AB* deductions *(GAB deductions) over Tp are the deductions built from formulas on Tp (the base case) using the following conditional rules ($C \vdash_{NL} B$ must be valid in NL):*

$$\frac{A/B \quad C}{A}\,FApp \qquad \frac{C \quad B\backslash A}{A}\,BApp$$
$$C \vdash_{NL} B \text{ valid in } NL$$

GAB deductions can be seen as a generalization of *AB* deductions in the following sense: for *AB* application rules C and B must be the same formula.

Definition 9 (*FA* structure of a *GAB* deduction). *To each GAB deduction $\mathcal{P}$, we associate a FA structure, written $FA_{Tp}(\mathcal{P})$, such that each internal node corresponds to the application of a rule in $\mathcal{P}$ and is labelled by the name of this rule and where the leaves are the same as in $\mathcal{P}$.*



$$\dfrac{\dfrac{NP/N \quad N}{NP}\,FApp \qquad \dfrac{(\overline{NP}\backslash S)/NP \qquad \dfrac{NP/N \quad N}{NP}\,FApp}{\overline{NP}\backslash S}\,FApp}{S}\,BApp$$

Here, $\overline{NP} = X/(NP\backslash X)$ and thus $NP \vdash_{NL} \overline{NP}$