Henning Christiansen
Peter Rossen Skadhauge
Jørgen Villadsen (Eds.)

LNAI 3438

Constraint Solving and Language Processing

First International Workshop, CSLP 2004
Roskilde, Denmark, September 2004
Revised Selected and Invited Papers

Springer

Henning Christiansen Peter Rossen Skadhauge
Jørgen Villadsen (Eds.)

Constraint Solving and Language Processing

First International Workshop, CSLP 2004
Roskilde, Denmark, September 1-3, 2004
Revised Selected and Invited Papers

Springer

Series Editors

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

Volume Editors

Henning Christiansen
Roskilde University, Computer Science
Building 42.1, DK-4000 Roskilde, Denmark
E-mail: henning@ruc.dk

Peter Rossen Skadhauge
Copenhagen Business School, Department of Computational Linguistics
Center for Computational Modeling of Language
Bernhard Bangs Alle 17 B, DK-2000 Frederiksberg, Denmark
E-mail: prs@id.cbs.dk

Jørgen Villadsen
Roskilde University, Computer Science
Building 42.1, DK-4000 Roskilde, Denmark
E-mail: jv@ruc.dk

Library of Congress Control Number: 2005926888

CR Subject Classification (1998): I.2.3, I.2.6-7, I.2, F.4.2-3, D.1.6, D.3.2

ISSN	0302-9743
ISBN-10	3-540-26165-6 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-26165-0 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: 11424574 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3438

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Preface

This volume contains selected and thoroughly revised papers plus contributions from invited speakers presented at the *1st International Workshop on Constraint Solving and Language Processing*, held in Roskilde, Denmark, September 1–3, 2004.

Constraint programming and constraint solving, in particular constraint logic programming, appears to be a very promising platform, perhaps the most promising present platform, for bringing forward the state of the art in natural language processing, this due to the naturalness in specification and the direct relation to efficient implementation. Language, in the present context, may refer to written or spoken language, formal or semiformal language, and even general input data to multimodal and pervasive systems, which can be handled in very much the same ways using constraint programming.

The notion of constraints, with slightly differing meanings, applies in the characterization of linguistic and cognitive phenomena, in formalized linguistic models as well as in implementation-oriented frameworks. Programming techniques for constraint solving have been, and still are, in a period with rapid development of new efficient methods and paradigms from which language processing can profit. A common metaphor for human language processing is one big constraint solving process in which the different(-ly specified) linguistic and cognitive phases take place in parallel and with mutual cooperation, which fits quite well with current constraint programming paradigms.

It would be too much to claim that this volume gives an extensive overview of the area of constraint-based language processing, but one of our main goals with the publication is to give more attention to this important research direction and to provide an opportunity for researchers from different traditions and research environments to see themselves in this common trend. We present a selection of interesting contributions, showing the spectrum from models grounded on cognitive studies to implementation-oriented formal models involving constraint programming. The often enthusiastic discussions at the workshop indicated a mutual interest and relevance across borders, and we are convinced that much more exciting research exists that can benefit from being brought into this forum. A second workshop on the topic is under preparation and we hope that this volume can stimulate interest.

We would like to thank the participants, the Program Committee, whose members are listed below, the invited speakers Philippe Blache, Veronica Dahl, Denys Duchier, and Gerald Penn, all researchers who submitted papers to the workshop, and those who made it practically possible.

Roskilde, March 2005

Henning Christiansen, Peter Rossen Skadhauge, Jørgen Villadsen

Organization

Organizing Committee, Editors

Henning Christiansen, Roskilde University (Chair)
Peter Rossen Skadhauge, Copenhagen Business School
Jørgen Villadsen, Roskilde University

Program Committee

Troels Andreasen, Roskilde, Denmark
Philippe Blache, Aix-en-Provence, France
Henning Christiansen, Roskilde, Denmark (Chair)
Veronica Dahl, Simon Fraser University, Canada
Denys Duchier, LORIA, France
John Gallagher, Roskilde, Denmark
Claire Gardent, LORIA, France
Daniel Hardt, Copenhagen Business School, Denmark
Jørgen Fischer Nilsson, Technical University of Denmark
Kiril Simov, Bulgarian Academy of Science
Peter Rossen Skadhauge, Copenhagen Business School, Denmark
Jørgen Villadsen, Roskilde, Denmark

Sponsors

The 1st International Workshop on Constraint Solving and Language Processing was supported by

- the CONTROL project, Constraint-Based Tools for Robust Language Processing, funded by the Danish Natural Science Research Council,
- CMOL, Center for Computational Modelling of Language at Copenhagen Business School, and
- Computer Science Section at Roskilde University, which also hosted the workshop.

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3518: T.B. Ho, D. Cheung, H. Li (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V. A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

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

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3464: S.A. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (Eds.), Engineering Self-Organising Systems. XIII, 299 pages. 2005.

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

Vol. 3438: H. Christiansen, P.R. Skadhauge, J. Villadsen (Eds.), Constraint Solving and Language Processing. VIII, 205 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. 3301: G. Kern-Isberner, W. Rödder, F. Kulmann (Eds.), Conditionals, Information, and Inference. XII, 219 pages. 2005.

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.

Table of Contents

Invited Papers

Property Grammars: A Fully Constraint-Based Theory
Philippe Blache 1

An Abductive Treatment of Long Distance Dependencies in CHR
Veronica Dahl 17

Metagrammar Redux
Benoît Crabbé, Denys Duchier 32

The Other Syntax: Approaching Natural Language Semantics Through Logical Form Composition
Gerald Penn, Frank Richter 48

Contributed Papers

Gradience, Constructions and Constraint Systems
Philippe Blache, Jean-Philippe Prost 74

Problems of Inducing Large Coverage Constraint-Based Dependency Grammar for Czech
Ondřej Bojar 90

Multi-dimensional Graph Configuration for Natural Language Processing
Ralph Debusmann, Denys Duchier, Marco Kuhlmann 104

An Intuitive Tool for Constraint Based Grammars
Mathieu Estratat, Laurent Henocque 121

Parsing Unrestricted German Text with Defeasible Constraints
Kilian A. Foth, Michael Daum, Wolfgang Menzel 140

Animacy Information in Human Sentence Processing: An Incremental Optimization of Interpretation Approach
Monique Lamers, Helen de Hoop 158

An Exploratory Application of Constraint Optimization in Mozart to Probabilistic Natural Language Processing
Irene Langkilde-Geary 172

A Constraint-Based Model for Lexical and Syntactic Choice in Natural Language Generation
Véronique Moriceau, Patrick Saint-Dizier 184

Author Index 205

Property Grammars: A Fully Constraint-Based Theory

Philippe Blache

LPL-CNRS, Université de Provence,
29 Avenue Robert Schuman, 13621 Aix-en-Provence, France
pb@lpl.univ-aix.fr

Abstract. This paper presents the basis of *Property Grammars*, a fully constraint-based theory. In this approach, all kinds of linguistic information is represented by means of constraints. The *constraint system* constitutes then the core of the theory: it is the grammar, but it also constitutes, after evaluation for a given input, its description. *Property Grammars* is then a non-generative theory in the sense that no structure has to be build, only constraints are used both to represent linguistic information and to describe inputs. This paper describes the basic notions used in *PG* and proposes an account of long-distance dependencies, illustrating the expressive power of the formalism.

1 Introduction: Constraints in Property Grammars

Constraints are usually used in linguistics as a filtering process. They represent fine-level information in order to rule-out some unwanted structures. According to the theory, constraints can more or less play an intensive role, as it is the case with *HPSG* (see [18]), *Optimality Theory* (see [17]), Constraint *Dependency Grammars* (see [16]), or in other kinds of formalisms such as *Dynamic Syntax* (see [13]). We propose to extend the use of constraints to the representation and the treatment different of linguistic information.

This approach presents several advantages, in particular the possibility of representing (and treating) separately different linguistic information. This idea was initially implemented in GPSG (see [8]) which proposes, by means of the ID/LP formalism, to distinguish hierarchy from linear order whereas this information is merged in classical phrase-structure grammars. The introduction of constraints completed this tendency in offering the possibility of representing directly some specific information such as cooccurrence restriction. We propose, in *Property Grammars* (cf. [3]) to represent separately each type of linguistic information by means of a specific constraint. In this way, constraints can represent all kinds of linguistic information.

One consequence of this characteristic is that constraints can be evaluated independently from each other. In generative theories, syntactic information is represented within a system in which all information has to be interpreted with respect to the entire system. In other words, a phrase-structure rule, which is

H. Christiansen et al. (Eds.): CSLP 2004, LNAI 3438, pp. 1–16, 2005.

the basic representation level in generative approaches, is not evaluable in itself. It takes its meaning in the global derivational process, and has no specific value outside of it. This is what Pullum in his lectures (cf. also [10]) call the *holistic* aspect of generative theories. And this is one of the main drawbacks of such theories. For example, taking into account the different uses of a language (for example spoken language) usually leads to deal with partial information. This means the necessity of interpreting sub-parts of the description system, which is incompatible with the generative model. An alternative approach consists in representing linguistic information in a decentralized way. In this case, each information is autonomous in the sense that it can be evaluated independently from the entire system. Constraints can then constitute an interesting alternative to classical approaches both for theoretical and computational reasons. Because of its flexibility (representation of partial information, no need of entire structures and non-holistic representation), a fully constraint-based approach, as the one presented here, makes it possible to describe any kind of input. Moreover, it becomes possible to take into account the contextual aspect of the linguistic phenomena: the same object or the same set of constituents can have different uses, different interpretations according to their environment. In the same perspective, interpreting a specific construction usually involve information coming from different sources (morphology, syntax, pragmatics, etc.). This way of thinking linguistic information is adopted in some recent works, in particular *Construction Grammars* (see [7]).

This paper proposes a new presentation of the Property Grammars approach (hereafter noted *PG*) that integrates the notion of construction. The first section describes the basic units in PG. These objects, called constructions, represent both local and global information, the global one being represented in terms of constraints. The second section presents the notion of property in syntax. This section shows in what sense any kind of linguistic information can be represented by means of constraints. The last section addresses the question of long distance dependencies, illustrating the fact that using constraints and constructions constitutes an efficient and simple way of representing linguistic information.

2 Constructions in PG

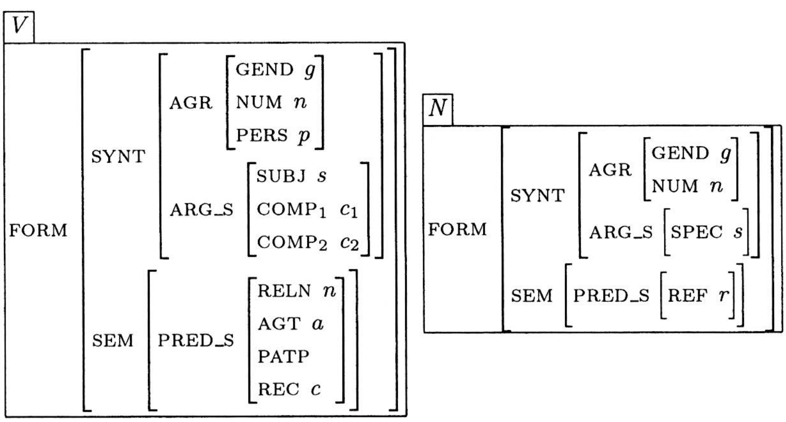
The notion of construction, as presented in the Construction Grammars paradigm (see [7], [11]), consists in bringing together different parts of linguistic information, making it possible to describe specific or marked phenomena in terms of interacting properties. More precisely, a construction specifies a set of constituents plus different values or relations that these constituents bear. Any kind of information (in particular coming from syntax and semantics) can be used in such structures. All objects can be described in terms of constructions: lexical categories as well as phrases or specific syntactic phenomena. One of the interests of this notion is the possibility of representing directly contextual phenomena. We propose in the following a PG version of the notion of construction.

Generally speaking, it is necessary to distinguish between two different types of information: the one specific to the object (also called local information, or form) and the relations between the different components of the object, expressed in terms of properties or constraints (see next section). A construction in PG is then a triple <L, F, P> in which L is the label of the construction, F its local information and P the properties representing relations between its components:

Label
FORM (*feature structure*)
PROPERTIES (*constraints*)

The form of the construction is represented by a feature structure. This structure is not fixed; the ones presented in this paper are a proposal among other possibilities. The important point is that this structure does not contain directly information about the constituents. Its role consists in instantiating different feature values, eventually by propagating some information from the relation side, as it will be shown later. In this version, we propose a basic repartition into syntactic and semantic features that can be completed by features coming from other domains. The structures are typed, which makes it possible to specify different forms according to the construction.

The following figures present the basic structures of two lexical constructions: the noun and the verb (we only show in these examples the Form part of the constructions). We make use in these structures of a very basic representation, making it possible to specify on the one hand the argument structure (a syntactic information) and a predicative structure (a semantic one). These values are not appropriated for all lexical constructions. For example, the adverb construction doesnt bear such structures. Moreover, when the feature is appropriated to a type, its form can vary (as it is the case between nouns and verbs).



The argument structure is indicated in order to specify for a given construction its subcategorization schema. We will see that this information is also implemented by means of a property (requirement). At this stage, it is used in order to implement, if necessary, some semantic restrictions (for example lexical selection). In the case of the noun, this feature implements the subcategorization of the determiner as a specifier. As for the verb, we consider the subject and two complements as the basic structure. The predicative structure bears the basic roles. We consider for the verb the following ones: agent, patient and recipient, plus the relation name.

A construction is then a set of information describing a category, a phrase, a specific phenomenon. In other words, a construction is the general description of a particular phenomenon. The description of a given input is done by means of a set of constructions that are instantiated according to the input. An instantiated construction is called an object and can be used as a constituent of other constructions. More precisely, activating a construction results in inferring that the described object (referred by the label) is considered as realized. This means, from an operational point of view, that the object corresponding to the label is instantiated and added to the set of objects to be taken into consideration (this aspect will be detailed later). In the following, objects (or instantiated constructions) are indicated by means of an index.

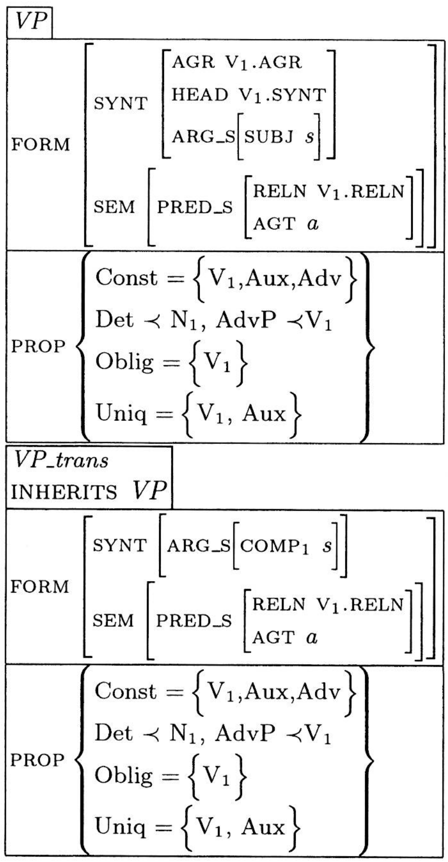
The following example illustrates the NP construction. The form of the construction is specified by means of a feature structure indicating some syntactic and semantic properties (again, this structure can be modified in function of specific requirements). We will present in the next section more precisely the notion of property. One can already note in this representation the relations between the constituents and feature values, in terms of structure sharing. For example, the semantic referential feature takes as value the semantic value of the noun N1, one of the constituents of the construction.

NP

$$\begin{array}{|ll|}\hline \text{FORM} & \begin{bmatrix} \text{SYNT} & \begin{bmatrix} \text{AGR } N_1.\text{FORM.SYNT.AGR} \\ \text{HEAD } N_1.\text{FORM.SYNT} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{PRED\_S} & \begin{bmatrix} \text{REF } N_1.\text{FORM.SEM} \\ \text{QUANT Det}_1.\text{FORM.SEM.QUANT} \end{bmatrix} \\ \text{MOD } AP_1.\text{FORM.SEM} & \end{bmatrix} \end{bmatrix} \\ \hline \text{PROPERTIES} & \left\{ \begin{array}{l} \text{Const} = \{\text{Det}_1, N_1, AP_1, PP, \text{Pro}\} \\ \text{Det} \prec N_1,\ N_1 \prec PP \\ \text{Oblig} = \{N_1, \text{Pro}\} \end{array} \right\} \\ \hline \end{array}$$

Properties will be presented in detail in the next section. Roughly speaking, the one presented here describes the list of constituents, the linearity and the possible heads of the construction. References to feature values are indicated by means of a path in the object. In order to simplify notations, paths are indicated in reduced forms when there is no ambiguity (for example N.AGR instead of N.FORM.SYNT.AGR). In the same way, for simplicity reasons, we note an object by its label instead of the entire path towards the constituency property (for example, we note N_1 instead of NP.PROPERTIES.CONST.N_1). We can also notice the use of a MOD feature in order to represent modification which comes from the adjective (or, for the VP, from different modalities such as modal verbs, negation, etc.).

As it is usually the case, constructions are typed; the types being structured into a hierarchy. A type specifies for a given construction some characteristics in terms of features (the appropriate features in HPSG) and relations. The inheritance hierarchy is explicit: a construction specifies from what other construction it inherits. In the example of the figure (3), the subject-auxiliary construction is a specification of the verb phrase in the sense that all properties of the VP also apply to this construction. Some of them are overridden, which means that a property can be replaced by another one which is more specific (as it is the case in an object-oriented paradigm).



3 The Properties

The basic syntactic information corresponds to some regularities that can be observed and described separately. The kind of information presented in the following list illustrates this point:

- words follow a certain (partial) order
- some words are mutually exclusive in a close context
- some words systematically coocur in a close context
- some words cannot be repeated in a close context
- the form of some words co-varies in a close context
- the realization of some words is more facultative than some others

In PG, the idea consists in specifying each kind of such information by means of a distinct constraint. The remaining of the section proposes a presentation of the different types of constraints (see also [vanRullen03] for a slightly different presentation of the notion of property). We use for the property description the following definitions:

- let $\mathcal{K}$ be a set of categories, let $\mathcal{A}$ be a subset of categories corresponding to a given input,
- let $pos(\mathcal{C}, \mathcal{A})$ be a function that returns the position of $\mathcal{C}$ in $\mathcal{A}$,
- let $card(\mathcal{C}, \mathcal{A})$ be a function that returns the number of elements of type $\mathcal{C}$ in $\mathcal{A}$,
- let $\{\mathcal{C}_1, \mathcal{C}_2\} \in \mathcal{K}$,
- let $comp(\mathcal{C}_1, \mathcal{C}_2)$ be a function that verifies the semantic compatibility of $\mathcal{C}_1$ and $\mathcal{C}_2$ and complete the semantic structure of $\mathcal{C}_2$ with that of $\mathcal{C}_1$.

3.1 Precedence

The linear precedence relation stipulates explicitly the order relations between the objects. These relations are valid inside a context.

Def: $\mathcal{C}_1 \prec \mathcal{C}_2$ is satisfied for $\mathcal{A}$ iff $pos(\mathcal{C}_1, \mathcal{A}) < pos(\mathcal{C}_2, \mathcal{A})$

This property makes it possible to represent explicitly what is often done implicitly (as for example in HPSG).

3.2 Constituency

This relation indicates for a given construction the set of categories (or constructions) that are used in its description. This is then a classical description of constituency. However, in PG, this notion plays a very secondary role: it is theoretically possible to use any kind of constituent for any construction. The only necessity is to distinguish between a construction made only with licit constituents and another that could also contain other constituents, violating then this constraint. The important point is that this constraint does not have any priority over other constraints as in classical phrase structure grammars. It is even possible not to use it without any deep consequence on the parsing process itself.

3.3 Obligation

Among the set of constituents, some of them are required. This notion of obligation recovers partially that of head. However, it is always possible to have different categories in this set.

Def: $\mathit{Oblig}(\mathcal{C}_1)$ is satisfied for $\mathcal{A}$ iff $card(\mathcal{C}_{1},_{A}) = 1$

This notion of obligation makes it possible to identify specific elements that constitutes in a certain sense the core of the construction. However, at the difference with the notion of head, we do not consider that the status of obligatory element has consequences on the government and semantic relations. It is possible for some non-head elements within a construction to have direct relations. For example, within a NP, determiners can have some restrictions on the adjuncts. At the semantic level, it can also be the case that a dependency exists between two elements that are not heads. This is another argument in favor of a separate representation of the various types of information.

3.4 Uniqueness

This property indicates the categories that can be realized only once into a construction.

Def: $\mathit{Uniq}(\mathcal{C}_1)$ is satisfied for $\mathcal{A}$ iff $card(\mathcal{C}_1, \mathcal{A}) \leq 1$

This property has to be explicit for every elements, including the obligatory ones.

3.5 Agreement

This property is used to implement different kinds of agreement. The classical agreement constraint consists in stipulating that two constituents of a construction must share the same agreement features. This is represented by a relation specifying the kind of features involved in the agreement and the objects concerned by the agreement. For example, the number agreement between the determiner and the noun is implemented as follows:

NP	
PROP	$\left\{ \text{Det} \leftrightarrow_{[num]} \text{N} \right\}$

In some cases, the agreement value has to be propagated from a lexical construction to a phrasal one, as for the agreement between adjective and nouns in French which is stipulated between the N and the AP of a NP construction. In this case, the propagation has to be implemented between the Adj and the AP constructions, which is directly implemented as follows:

AP	
FORM	AGR Adj_1.AGR
PROP	$\left\{ \text{Oblig} = \left\{ \text{Adj}_1 \right\} \right\}$