Compilation de grammaire de propriétés pour l'analyse syntaxique par optimisation de contraintes

Jean-Philippe Prost[1] Rémi Coletta[1] Christophe Lecoutre[2]
(1) LIRMM, CNRS – Université de Montpellier, Montpellier, France
(2) CRIL, CNRS – Université d'Artois, Lens, France
{prost,coletta}@lirmm.fr, lecoutre@cril.fr

RÉSUMÉ

Cet article présente un processus de compilation d'une grammaire de propriétés en une contrainte en extension. Le processus s'insère dans le cadre d'un analyseur syntaxique robuste par résolution d'un problème d'optimisation de contraintes. La grammaire compilée est une énumération de tous les constituants immédiats uniques de l'espace de recherche. L'intérêt de ce travail encore préliminaire tient principalement dans l'exploration d'une modélisation computationnelle de la langue à base de Syntaxe par Modèles (MTS, *Model-Theoretic Syntax*), qui intègre la représentation indifférenciée des énoncés canoniques et non-canoniques. L'objectif plus particulier du travail présenté ici est d'explorer la possibilité de construire l'ensemble des structures candidat-modèles à partir de l'ensemble des structures syntagmatiques observées sur corpus. Cet article discute notamment le potentiel en matière d'intégration de prédictions probabilistes dans un raisonnement exact pour contribuer à la discrimination entre analyses grammaticales et agrammaticales.

ABSTRACT

Compilation of a Property Grammar for Syntactic Parsing through Constraint Optimisation

This paper introduces the compilation process of a property grammar into a constraint in extenso. The process is part of a robust syntactic parser implemented as the resolution of a Constraint Optimisation Problem. The compiled grammar enumerates all the unique immediate constituents in the search space. The interest of this preliminary work stands in the exploration of a Model-Theoretic computational modelling of language, which integrates the representation of both canonical and non-canonical utterances. The objective of this work is more particularly to explore the possibility to build the set of all candidate models from a set of phrasal structures observed on corpus. The paper also discusses the potential integration of probabilistic predictions within an exact reasoning process, in order to discriminate the grammatical parses from the ungrammatical ones.

MOTS-CLÉS : syntaxe par modèles, jugement de grammaticalité, ingénierie de grammaire.

KEYWORDS: Model-Theoretic Syntax, Grammaticality Judgement, grammar engineering.

Le domaine du TAL exprime depuis peu le souhait de voir un regain d'intérêt pour les problèmes, approches, et architectures, et appelle à mettre à nouveau l'accent sur l'investigation scientifique et cognitive des langues plutôt que s'appuyer presque exclusivement sur un modèle de recherche à base d'ingénierie (Manning, 2015). Les travaux présentés ici s'inscrivent dans cette lignée. Plus généralement, ils s'insèrent dans un projet de dévelopement d'un analyseur syntaxique qui offre un environement expérimental pour une représentation de la syntaxe par modèle (MTS, *Model-Theoretic Syntax*). Nous présentons ici le processus de compilation de la grammaire sur lequel repose cet

analyseur.

Le §1 introduit le contexte et la problèmatique générale ; le §2 présente le processus de compilation, tandis que le §3 conclue.

1 Problématique et objet d'étude

Le problème général auquel nous nous intéressons, et dont les travaux présentés ici constituent une des étapes, est l'analyse syntaxique robuste de langage tout-venant. Par *langage tout-venant* nous entendons couvrir à la fois le langage *canonique*, qui satisfait parfaitement aux règles de grammaire établies, et le langage *non-canonique*, qui viole certaines règles de grammaire mais reste néanmoins partiellement grammatical à travers le respect d'un sous-ensemble non-nul de règles. Comme le discutent notamment Pullum & Scholz (2001), sur un plan linguistique certaines de ces violations ne constituent pas nécessairement des erreurs à proprement parler, mais peuvent par exemple concerner différents phénomènes observables comme des énoncés incomplets ("*Tu m'avais dit que ...*"), des néologismes, des emprunts aux langues étrangères, etc. Plus largement encore, les phénomènes qui nous intéressent incluent les erreurs de génération automatique, de transcription de l'oral, le style télégraphique, etc. Notons que par abus de langage il nous arrivera, par référence à l'usage commun, de parler d'*énoncé agrammatical* en lieu et place d'un énoncé non-canonique, en dépit du fait qu'il puisse être partiellement grammatical.

Ce faisant, nous souhaitons également pouvoir distinguer sans ambiguïté les énoncés canoniques de ceux non-canoniques. Nous souhaitons donc pouvoir décrire la structure syntaxique d'un énoncé quelconque, soit par une structure exacte dans le cas d'un énoncé canonique, soit par une structure approchée dans le cas d'un énoncé non-canonique. Ce point revêt une importance particulière pour ce qu'il exclut les approches strictement probabilistes.

Représentation par modèles de la syntaxe La famille des cadres formels relevant de la *syntaxe par modèles* (MTS [1]) se prête naturellement à une modélisation de l'analyse syntaxique sous la forme d'un problème de satisfaction de contraintes (CSP), ainsi qu'à la représentation de la gradience syntaxique. Ces deux propriétés justifient notre choix. Plus particulièrement, nous adoptons le cadre des Grammaires de Propriétés (GP) (Blache, 2001) pour lequel une formalisation par modèles a été proposée par Duchier *et al.* (2009).

La MTS s'appuie sur la théorie logique des modèles pour formaliser la représentation de la structure syntaxique d'une phrase. Par analogie à la théorie des modèles, le principe général est que les règles de la grammaire G qui s'appliquent à une phrase s et à l'arbre syntaxique associé $\tau : s$ constituent une formule conjonctive $\Phi_{G,\tau} = \bigwedge_i \varphi_i$ dont chaque formule atomique φ_i peut être évaluée indépendamment des autres sur τ. La structure syntaxique $\tau : s$ pour la phrase s est alors un modèle pour la théorie $\Phi_{G,\tau}$ (noté $\tau : s \models \Phi_{G,\tau}$) ssi $\tau : s$ rend vraie chaque φ_i de $\Phi_{G,\tau}$.

En GP, chaque φ_i est une formule relationnelle sur les fils d'un nœud π de $\tau : s$ (et leurs étiquettes respectives), appelée *propriété*. La formalisation de Duchier *et al.* définit la sémantique pour six des relations proposées par Blache (2001). Il n'est pas nécessaire, pour ce qui nous intéresse ici, de maîtriser dans le détail le formalisme utilisé en GP. Précisons simplement qu'il permet de spécifier une grammaire sous la forme d'un ensemble de propriétés, indépendantes les unes des autres (i.e. les

1. pour *Model-Theoretic Syntax*.

φ_i). Par exemple, la propriété de *Linéarité* permet d'exprimer que "*Dans un Syntagme Nominal (SN) en français, le Déterminant précède le Nom (dans le sens de lecture)*", et la propriété d'*Obligation* permet d'exprimer que "*Dans un SN en Français la présence d'un Nom est obligatoire*".

Un problème central aux approches par modèles et à leur implantation est la surgénération des structures candidates, qui constituent l'espace de recherche d'une structure modèle $\tau : s$. Dans notre cas ce problème est amplifié par la notion de langage que nous adoptons, étendue aux énoncés non-canoniques. Cet article se concentre sur un élément particulier de la stratégie d'analyse que nous implantons, la compilation de la grammaire, qui s'attaque au problème de surgénération tout en permettant un changement d'échelle par rapport à l'implantation de Duchier *et al.* (2010). Bien que cette compilation ne règle pas entièrement le problème, elle permet néanmoins d'envisager d'autres pistes d'amélioration que nous discutons plus loin, notamment en matière de choix heuristique de filtrage des solutions sur la base de prédictions probabilistes.

Analyse syntaxique et résolution par contraintes Les travaux de Maruyama (1990) montrent que l'analyse syntaxique peut être modélisée comme un problème de résolution de contraintes (CSP), et implantent un analyseur pour un domaine d'arbres en dépendances. Ces travaux définissent les Grammaires de Dépendance par Contraintes (CDG, *Constraint Dependency Grammar*). L'implantation des CDG au moyen de CSP a été largement développée pour le traitement de la parole (Harper & Wang, 2010). Les CDG ont également été étendues notamment par Schröder (2002) à une modélisation en WCSP (Weighted CSP) pour l'analyse robuste, et par Debusmann *et al.* (2004) pour une analyse multi-niveaux en dépendances, et un traitement de l'analyse et de la génération de langage naturel par Programmation Par Contraintes (PPC). Ces extensions se font néanmoins au détriment de la discrimination entre langage canonique et non-canonique.

L'analyse en structure syntagmatique a également fait l'objet de modélisations et implantations en CSP, parmi lesquelles certaines ne permettent cependant pas une analyse robuste (Morawietz & Blache, 2002; Dahl & Blache, 2004; Estratat, 2006). Les travaux de Duchier *et al.* (2010) (ci-après DDPL10), en revanche, offre en théorie cette propriété de robustesse, mais l'implantation connue à ce jour n'offre pas les performances suffisantes pour envisager des expérimentations à échelle réaliste. Ils constituent cependant une approche originale dont s'inspire largement l'implantation à laquelle il est fait référence ici. La compilation de la grammaire que nous proposons doit permettre d'envisager un changement d'échelle par rapport aux travaux de DDPL10.

Contrairement aux grammaires génératives, une grammaire MTS ne spécifie aucune règle de dérivation. Par conséquent, lors de la modélisation du processus d'analyse en CSP il convient :

- d'une part d'établir une stratégie de dérivation permettant le parcours de l'espace de recherche et la génération d'une structure en constituants, et
- d'autre part d'évaluer une structure pour une grammaire GP donnée.

Pour ce faire, deux types de contraintes sont nécessaires et cohabitent au sein d'un même CSP :

- des contraintes *de grammaire*, qui pour les GP reposent sur les relations mentionnées plus haut (i.e. les φ_i, appelées *propriétés*) ;
- des contraintes *structurelles*, qui permettent de construire la structure dérivationnelle, c'est-à-dire l'arbre syntaxique en constituants.

Les modélisations CSP existantes, notamment DDPL10, combinent toutes ces deux types de contraintes au sein d'un seul et même système, résolu dynamiquement. Nous proposons de reléguer la vérification des contraintes de grammaire lors d'une phase statique préliminaire de compilation. Le

processus d'analyse se ramène alors à la seule résolution dynamique des contraintes structurelles [2]. Cette compilation de la grammaire résulte en une contrainte en extension, qui explicite l'ensemble des combinaisons de valeurs possibles autorisées sur un ensemble de variables et associe un coût à chaque combinaison. En l'occurrence, chaque combinaison peut être vue comme une règle de grammaire hors-contexte (CFG) valuée par un score de grammaticalité ≥ 0. Néanmoins, comme nous l'avons déjà évoqué plus haut ces règles CFG ne servent pas de règles de dérivation mais simplement à spécifier les sous-structures élémentaires qui peuvent être combinées ensemble pour constituer une structure candidat-modèle. La combinaison de ces sous-structures en un arbre syntaxique complet résulte de la résolution du système de contraintes structurelles, et non d'un processus de dérivation. Nous parlerons de *constituant immédiat* pour faire référence à la sous-structure élémentaire spécifiée par une règle CFG, c'est-à-dire à l'arbre de hauteur 1 constitué d'un nœud (i.e. la partie gauche de la règle CFG) et de ses fils (i.e. la partie droite de la règle CFG).

Le score de chaque constituant résulte de l'évaluation de la grammaire GP pour ce constituant immédiat. Plus précisément, il correspond à la proportion d'instances de propriétés violées $I^{-}_{G,\tau}$ parmi la somme des instances satisfaites et violées $I^{0}_{G,\tau} = I^{+}_{G,\tau} \cup I^{-}_{G,\tau}$. Ce score est le dual du score *d'adéquation* (*fitness*) $f(\Phi_{G,\tau})$ défini par Duchier *et al.* (2009) comme la proportion d'instances satisfaites. La phase dynamique du processus d'analyse cherche alors une configuration de constituants immédiats (i.e. de règles CFG) qui optimise le score global en s'appuyant sur la sémantique non-classique des GP proposée par Duchier *et al.* (2009), selon laquelle $\tau : s \mid\!\approx \Phi_{G,\tau} \quad \equiv \quad \tau \in \text{argmax}(f(\Phi_{G,\tau}))$.

L'intérêt prinicipal d'une telle compilation est d'éviter bon nombre de contraintes réifiées dans le CSP final (la réification étant nécessaire pour implanter la sémantique des types de propriétés utilisés en GP). Ce faisant, elle permet également d'exclure du CSP final l'évaluation de la grammaire GP pour chaque constituant, et donc de gagner en temps d'exécution.

Taille de l'espace de recherche et surgénération des structures candidat-modèles La surgénération des structures candidates est un problème crucial des approches à base de résolution de contraintes. Nous proposons ici de contrôler en partie cette surgénération en distinguant plusieurs types de constituants immédiats :

- ceux observés sur corpus, toujours considérés comme grammaticaux, et donc de coût 0,
- ceux théoriques, non-observés sur corpus, parmi lesquelles on distingue
  - les constituants grammaticaux, de coût théorique 0, et
  - les constituants agrammaticaux, de coût théorique > 0.

Par *coût théorique* nous faisons référence au coût obtenu par évaluation de la grammaire GP pour ce constituant immédiat (cf. §2). Afin de réduire le champ des possibilités d'analyses grammaticales nous attribuons en pratique aux constituants grammaticaux théoriques (i.e. non observés) un coût arbitraire non-nul, dont il conviendra de calibrer la valeur afin de ne pas confondre ces derniers avec les constituants agrammaticaux.

La classe d'arbres à laquelle nous nous intéressons est la même que celle de Duchier *et al.* (2009). On note $\mathbb{N}_0$ pour $\mathbb{N} \setminus \{0\}$; un *domaine d'arbres* D est un sous-ensemble fini de $\mathbb{N}_0^*$, fermé pour les préfixes et les fils gauches. Soient également $\mathcal{S}^*$ un lexique, vu comme un ensemble de mots de la langue, et $\mathcal{L}$ un ensemble fini d'étiquettes représentant des catégories syntagmatiques. Un arbre syntaxique $\tau = (D_\tau, L_\tau, R_\tau)$ consiste en un domaine d'arbres, une fonction d'étiquetage $L_\tau : D_\tau \to \mathcal{L}$ qui

2. Remarquons que les modélisations CSP pour les CDG ne permettent pas une telle distinction entre contraintes de grammaire et contraintes structurelles du fait même du formalisme.

associe une étiquette à chaque nœud, et une fonction $R_\tau : D_\tau \rightarrow \mathcal{S}^*$ qui à chaque nœud associe sa réalisation de surface. Par abus de langage on appelle *empan* d'un constituant immédiat le nombre de ses fils, c'est-à-dire la cardinalité de l'ensemble des catégories présentes en partie droite de la règle correspondante — l'abus tenant au fait que le terme fait habituellement référence à la réalisation de surface d'un constituant, ce qui n'est pas le cas ici. Par extension, on parlera d'empan du nœud en partie gauche de la règle. Étant donné k une valeur d'empan maximum, l'ensemble théorique de toutes les règles CFG possibles est l'ensemble $R_k = \mathcal{L} \cup \mathcal{L} \times \mathcal{L} \cup \mathcal{L} \times \mathcal{L} \times \mathcal{L} \cup \ldots \cup \mathcal{L}^k$. La table de grammaire est alors l'ensemble $T \subseteq R_k$ tel que $\forall \tau \in T, I^0_{G,\tau} \neq \emptyset$. On peut poser que $R_k \simeq \bigcup_{i=1}^{k+1} \mathcal{L}^i$, où $\mathcal{L}^i$ est la puissance cartésienne i-ème de $\mathcal{L}$. Avec $l = \sharp\mathcal{L}$ la cardinalité de $\mathcal{L}$, on a alors un nombre de règles théoriques $\sharp R_k = \sum_{i=1}^{k+1} l^i = (1 - l^{k+2})/(1 - l) \simeq l^{k+1}$.

À noter que strictement parlant la taille d'empan maximum ne peut pas être connue a priori, puisqu'elle dépend de la phrase analysée. Notre modélisation nécessitant que cette valeur soit fixée, elle constitue un paramètre qu'il conviendra de calibrer expérimentalement.

Bien que la distinction entre constituants grammaticaux observés et constituants grammaticaux théoriques permette de limiter la surgénération, elle n'est, en pratique, pas suffisante, les grammaires de corpus étant généralement beaucoup trop permissives (au sens où une même catégorie de syntagme peut correspondre à un très grand nombre de constituants immédiats différents, ce qui réduit considérablement les possibilités de généralisation). Une possibilité de filtrage supplémentaire sera discutée ultérieurement, qui intègre dans le coût d'un constituant un facteur significatif de la fréquence d'occurrence de ce constituant observée sur corpus.

2 Compilation de la grammaire

L'objet de la compilation est de transformer les contraintes de grammaire en contraintes définies en extension (appelées également contraintes tables). Le résultat est un ensemble de n-uplets, chacun représentant une règle CFG et son coût associé (le coût correspondant au score de grammaticalité). Le score d'une règle (ou son *coût*, dans une optique CSP) résulte de sa *caractérisation* selon une grammaire GP. La caractérisation d'une règle consiste à vérifier l'ensemble des propriétés de la grammaire pour ce constituant immédiat. Il en résulte trois ensembles de propriétés évaluées : P^+, P^- et P^0, respectivement des propriétés satisfaites, violées et non-pertinentes (i.e. trivialement satisfaites). Comme vu précédement, le score de chaque constituant représente $\sharp P^- / \sharp(P^+ \cup P^-)$. Les constituants pour lesquels $P^+ = \emptyset = P^-$ sont ignorés.

Afin de réduire l'espace de recherche, nous avons vu que seules les règles observées ont un score nul. Les règles non-observées dont le score théorique serait nul se voient affecter un score arbitraire > 0 qui permet, lors de la résolution du problème (sous forme d'un problème d'optimisation sous contraintes, appelé COP, cette fois-ci), de toujours donner la priorité aux règles observées. L'attribution d'une valeur à ce score arbitraire fait l'object d'un calibrage expérimental.

Par souci de cohérence les constituants théoriques sont caractérisés à l'aide d'une grammaire GP dérivée du même corpus que celui des constituants observés. Le processus de dérivation, décrit dans (Prost, 2014), s'appuie lui-même sur la grammaire CFG extraite du corpus. Dans le cas du corpus Sequoia (Candito & Seddah, 2012), la CFG extraite de la partition *développement* est composée d'environ 1400 règles, qui sont dérivées en environ 1400 propriétés de GP, pour un alphabet de 39 catégories morpho-syntaxiques. Les règles extraites ont des tailles d'empan allant jusqu'à 14. À titre de comparaison, DDPL10 utilise une grammaire-jouet de 19 propriétés pour 6 catégories.

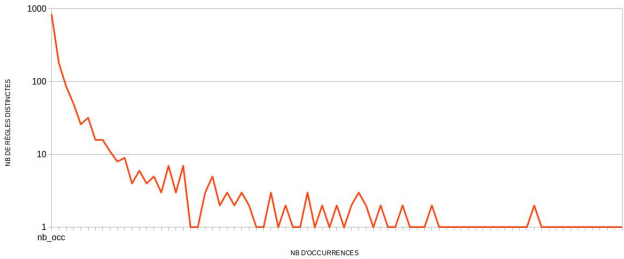

FIGURE 1 – Distribution des règles CFG extraites du corpus sequoia

Calibrage des coûts des constituants immédiats Une stratégie de filtrage lors de l'analyse syntaxique consiste à adapter la fonction d'adéquation de façon à prendre en compte, dans les coûts des constituants immédiats, leur fréquence d'occurrence observée sur corpus. Cette stratégie devrait permettre de contribuer à réduire le nombre de nœuds solutions dans l'espace de recherche. Une analyse du corpus sequoia, illustrée en figure 1, montre ainsi que 60 % des règles utilisées (848 sur 1410) n'ont qu'une seule occurrence dans l'ensemble du corpus de développement. Ces règles peuvent être identifiées comme non-prioritaires, tout en conservant un jugement exact de grammaticalité. Pour cela il convient de fixer un seuil de grammaticalité > 0 arbitraire, calibré empiriquement, de façon à pouvoir assigner aux règles grammaticales des coûts différents en fonction de la préférence à leur accorder lors de l'analyse. De cette façon il devient possible de réserver certaines règles, grammaticales mais exceptionnelles, à la résolution de constructions syntaxiques très particulières.

3 Conclusion

Le travail que nous avons préenté s'intègre dans un projet plus large de dévelopement d'un analyseur syntaxique robuste pour le langage tout-venant par résolution d'un problème d'optimisation de contraintes. Plus précisément, nous avons présenté un processus de compilation d'une grammaire par modèles en constituants qui résulte en une contrainte de table. Cette table explicite la liste exhaustive des constituants immédiats (ou règles CFG) valués pouvant être combinés lors du processus d'analyse syntaxique, dont l'objet est la construction d'une structure d'arbre syntaxique pour une phrase donnée. Cette grammaire compilée est donc destinée à être utilisée en conjonction avec une modélisation en problème d'optimisation de contraintes (COP) de l'analyse syntaxique en constituants. Lors de la résolution du COP l'utilisation d'une grammaire compilée doit permettre un changement d'échelle par rapport aux travaux comparables (Duchier *et al.*, 2010), en ceci qu'elle permet un meilleur contrôle du nombre de contraintes impliquées, et une suppression des contraintes réifiées nécessaires chez Duchier *et al.* à la représentation de la grammaire. La compilation fait appel à une grammaire de propriétés pour évaluer un score exact de grammaticalité pour chaque règle d'une grammaire hors contexte de corpus. Chaque score est alors transformé en un coût de satisfaction pris en compte lors du processus d'optimisation.

Ces travaux, bien que préliminaires, doivent permettre d'explorer de nouvelles pistes en matière de représentation des connaissances pour les langues naturelles, qui rendent compte à la fois des propriétés syntaxiques des énoncés canoniques et non-canoniques. Une telle représentation doit permettre également d'explorer les possibilités computationnelles en matière de jugement exact

de grammaticalité pour une analyse, et en matière de discrimination entre solutions exactes, donc grammaticales, et solutions optimales, donc agrammaticales.

Références

BLACHE P. (2001). *Les Grammaires de Propriétés : des contraintes pour le traitement automatique des langues naturelles*. Hermès Sciences.

CANDITO M.-H. & SEDDAH D. (2012). Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Actes de TALN'2012*, Grenoble, France.

DAHL V. & BLACHE P. (2004). Directly Executable Constraint Based Grammars. In *Journées Francophones de Programmation en Logique avec Contraintes (JPFLC'2004)*, p. 149–166, Angers, France.

DEBUSMANN R., DUCHIER D. & NIEHREN J. (2004). The XDG Grammar Development Kit. In *Proceedings of the 2nd International Mozart/Oz Conference, MOZ04*.

DUCHIER D., DAO T.-B.-H., PARMENTIER Y. & LESAINT W. (2010). Property Grammar Parsing Seen as a Constraint Optimization Problem. In *FG*, p. 82–96.

DUCHIER D., PROST J.-P. & DAO T.-B.-H. (2009). A Model-Theoretic Framework for Grammaticality Judgements. In *Proceedings of Formal Grammar (FG'09)*, volume 5591 of *LNCS* : FOLLI Springer.

ESTRATAT M. (2006). Vers les grammaires de configuration.

HARPER M. P. & WANG W. (2010). Constraint Dependency Grammars : SuperARVs, Language Modeling, and Parsing. In S. BANGALORE & A. K. JOSHI, Eds., *Language*, p. 207–239. MIT Press.

MANNING C. (2015). Computational Linguistics and Deep Learning. *Computational Linguistics*, **41**(4), 701—-707.

MARUYAMA H. (1990). Structural Disambiguation with Constraint Propagation. In *Proceedings 28th Annual Meeting of the ACL*, p. 31–38, Pittburgh, PA.

MORAWIETZ F. & BLACHE P. (2002). Parsing Natural Languages with CHR.

PROST J.-P. (2014). Jugement exact de grammaticalité d'arbre syntaxique probable. In *Proceedings of TALN 2014 (Volume 1 : Long Papers)*, p. 352–362, Marseille, France : Association pour le Traitement Automatique des Langues.

PULLUM G. K. & SCHOLZ B. (2001). On the Distinction Between Model-Theoretic and Generative-Enumerative Syntactic Frameworks. In *Logical Aspects of Computational Linguistics : 4th International Conference (LACL)*, Lecture Notes in Artificial Intelligence, p. 17–43, Berlin.

SCHRÖDER I. (2002). Natural Language Parsing with Graded Constraints.