From Intuitionistic Proof Nets to Interaction Grammars

Guy Perrier

LORIA - Université Nancy2
Campus Scientifique - BP 239
54506 Vandœuvre-lès-Nancy Cedex -France
e-mail: perrier@loria.fr

Abstract

We show that the construction of proof nets in the implicative fragment of intuitionistic linear logic reduces to the generation of models in the shape of completely specified and neutral trees from polarised tree descriptions. This provides us with a new framework for revisiting grammatical formalisms and leads us to introduce Interaction Grammars which aim to take advantage of two main characteristics of this framework: under-specification and polarities.

Introduction

Apparently, Categorial Grammars (CG) and Tree Adjoining Grammars (TAG) are two very different approaches to the syntax of natural languages. CG are characterised as calculi of polarised syntactic types based on the idea that grammatical categories are consumable resources: some constituents behave as resource consumers whereas others behave as resource providers so that syntactic composition is viewed as a process in which consumers and providers try to cancel each other out; most often, CG are expressed in a logical framework that takes the Lambek Calculus as its nucleus, which combines resource sensitivity with order sensitivity. This intimate combination, which explains the central role of this logic as a framework for CG, is at the same time a cause of rigidity which limits its expressive power greatly. The search for an appropriate way of relaxing this framework constitutes an important research area of CG (Moo96).

TAG do not manipulate syntactic types but syntactic trees with the adjunction operation as their cornerstone. In this way, their expressivity goes beyond that of CG but their rigidity is also their weak point: like CG, they are lexicalised and all syntactic configurations in which a word is used are stored in the lexicon in the form of elementary trees. As soon as a word is used in a new syntactic configuration, a new elementary tree must be added to the lexicon directly or via a lexical rule. In this way, lexicons quickly become colossal and very awkward to manage.

Recent works have contributed to establish links between CG and TAG with the common aim to embed TAG in a logical setting (AFV97; JK97). Our proposal aims to provide a common framework for comparing CG and TAG and for overcoming some of their specific limitations in a new formalism which we call *Interaction Grammars* (IG). The common framework that we choose is that of *tree descriptions*. This notion is not new in the TAG community since it was introduced by (VS92) for making adjunction monotone and embedding TAG in a unification framework. The key idea behind this notion is to replace reasoning about syntactic trees as completely defined objects with reasoning about properties which are used for characterising

these trees; in this way, syntactic trees are viewed as models of descriptions. This allows one to use the notion of *under-specification* in a fruitful manner for structuring TAG lexicons (Can99) or for dealing with semantic ambiguity (MK; ENRX98) for instance. This also allows a new and promising constraint-based style of computing within linguistics (?; DT99; Bla99).
We propose to show that CG can be revisited in this framework with new developments which lead us to IG. The starting point of this proposal is purely theoretic since it concerns proof theory in Intuitionistic Linear Logic (ILL).

1. Intuitionistic proof nets as polarised tree descriptions

Resource sensitivity of linear logic entails a specific form of proof: *proof nets* (Gir87). In the general framework of classical linear logic, these proof nets are not directed so that each extremity of a proof net can be viewed as either an input or an output; in other words, each formula that is attached to an extremity of a proof net can be considered either as an assumption (input) or as a conclusion (output) of the proof.
In ILL, this symmetry is broken and things freeze in a configuration where all formulas become *polarised*, one as the output (denoted +) and the others as the inputs (denoted –). F. Lamarche has devised a correctness criterion for these proof nets which takes their specificity into account (Lam96). Hence, he has sketched a more abstract representation of proof nets which is inspired by the games semantics for PCF introduced by (HO93) and which only takes the induced order between atomic formulas into account.
By using the notion of tree description, we propose to perfect this representation for Implicative Intuitionistic Linear Logic (IILL), which is the implicative fragment of ILL, built only from the linear implication ($\multimap$); we choose this fragment because of its linguistic interest but our proposal can be easily extended to the whole multiplicative fragment.

1.1. Syntactic descriptions of IILL formulas

Let $\mathcal{P}$ be a set of propositions. The set of IILL formulas built from $\mathcal{P}$ is defined by the grammar $\mathcal{F} ::= \mathcal{P} \mid \mathcal{F} \multimap \mathcal{F}$. By adding a polarity + or – to every IILL formula, we obtain the set $\mathcal{F}(\mathcal{P})$ of polarised IILL formulas. From the syntax of these formulas, we abstract particular tree descriptions, called *IILL syntactic descriptions.*

Definition 1.1 *An IILL syntactic description D is a set of polarised atomic formulas taken from $\mathcal{F}(\mathcal{P})$ that is equipped with two binary relations:* dominance *(denoted $>^*$) and* immediate dominance *(denoted $>$).*

For every polarised IILL formula F^p (p represents the polarity + or – and $-p$ its opposite), we build its syntactic description, denoted $D(F^p)$ from the root, denoted $Root(D(F^p))$, to the leaves recursively according to the following definition.

Definition 1.2 *$D(F^p)$ is an IILL syntactic description such that:*

- *if F^p is atomic, then $D(F^p)$ is reduced to the unique element F^p, the two relations $>^*$ and $>$ are empty and $Root(D(F^p)) = F^p$;*
- *if $F^p = (F_1 \multimap F_2)^p$, then $D(F^p)$ is the disjoint union of $D(F_1^{-p})$ and $D(F_2^p)$ where the relations $>^*$ and $>$ are completed with a relation between $Root(D(F_1^{-p}))$ and $Root(D(F_2^p))$ according to the following rule:*
  - *if p=+, then $Root(D(F_2^+)) >^* Root(D(F_1^-))$ and $Root(D(F^p)) = Root(D(F_2^+))$;*

– *if* p=–, *then* $Root(D(F_2^-)) > Root(D(F_1^+))$ *and* $Root(D(F^p)) = Root(D(F_2^-))$.

According to the previous definition, an IILL syntactic description has a very particular shape: it appears as a hierarchy of levels which alternate positive and negative formulas and, at the same time, dominance links and immediate dominance links between them.

1.2. Provability in IILL as validity of syntactic descriptions

Syntactic descriptions are interpreted on trees according to the following definition:

Definition 1.3 *A tree T is a model of a syntactic description D if there is an interpretation I from D to T such that:*

- *For every node N of T, $I^{-1}(N)$ is composed of exactly two elements of D: F^+ and F^-.*
- *For every pair $(F_1^{p_1}, F_2^{p_2})$ of D , $F_1^{p_1} > F_2^{p_2}$ $(F_1^{p_1} >^* F_2^{p_2})$ entails that $I(F_1^{p_1})$ is the parent (an ancestor) of $I(F_2^{p_2})$ in T.*

If a description D accepts a model, D is said to be valid.

In others terms, a syntactic description is valid if one can merge its nodes by dual pairs while respecting its dominance constraints. Equivalence between provability of IILL sequents in linear logic and validity of the corresponding syntactic descriptions is established by the following theorem.

Theorem 1.1 *An IILL sequent $F_1, \ldots, F_n \vdash G$ is provable in linear logic if and only if the syntactic description $D((F_1 \multimap \cdots \multimap F_n \multimap G)^+)$ is valid.*

Sketch of proof 1.1 *To show that provability entails validity, we proceed by induction on proofs of IILL sequents in the linear sequent calculus. We consider the last inference I of any proof of such a sequent. By induction hypothesis, we get models of the syntactic descriptions of the I-premises and it is not very difficult to combine these models to build a model of the syntactic description of the I-conclusion.*
To show that validity entails provability, we proceed by induction on the number of nodes of syntactic descriptions. We consider any valid description of an IILL formula F. We drop the root R^+ of the description and its dual node R^- which match in a model T; all partial descriptions $D(F_i^-)$ which become unconnected in this way are linked to the children of R^- that dominate them in the model T. In this way, we obtain a set of valid syntactic descriptions to which we can apply the induction hypothesis; as a consequence, we obtain a set of provable sequents from which we deduce $\vdash F$.

Example 1.1 *The transitivity of linear implication is expressed by the provability of the IILL sequent $a \multimap b,\ b \multimap c\ \vdash\ a \multimap c$, which amounts to the provability of the one-sided sequent $\vdash (a \multimap b) \multimap (b \multimap c) \multimap (a \multimap c)$. From the left to the right, Figure 1 successively presents the proof net which establishes this provability, the corresponding syntactic description and the model* [1] *which guarantees the validity of this description. In the proof net, positive formulas are represented by down arrays and negative formulas by up arrays; axioms links are represented by dotted edges.*

Proof search, which, in IILL, takes the form of proof net construction, now reduces to the generation of models from syntactic descriptions; some details are forgotten while essentials

[1] The model is unique up to an isomorphism.

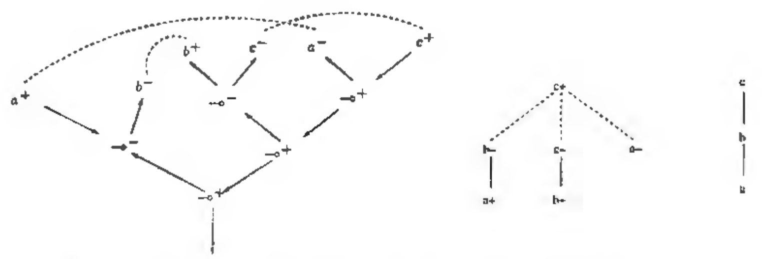

Figure 1: IILL proof of $\vdash (a \multimap b) \multimap (b \multimap c) \multimap (a \multimap c)$

are preserved: identification of dual nodes of a syntactic description corresponds to putting an axiom link in the corresponding proof net and the correctness criterion of the proof net is constantly guaranteed by the tree-like structure of the description. In order to extend these results to the whole multiplicative fragment of ILL, we have to relax the tree structure of descriptions to a DAG structure.

1.3. Planarity of Lambek proof nets and precedence order in syntactic descriptions

In the implicative fragment of the Lambek Calculus, linear implication is replaced by two implications, left and right, respectively denoted \ and / , which results from the non-commutativity of the calculus. Lambek proof nets differ from IILL proof nets by the fact that the premises of inference links are ordered and axiom links must not cross each other (Roo91).

This enrichment of IILL proof nets by a precedence order can be translated in the corresponding syntactic descriptions without difficulties: besides the two relations of dominance and immediate dominance, we add a *precedence* relation between atomic formulas. A difficulty comes then when we want to express the axiom links of a proof net with the merging of dual nodes in the corresponding syntactic description. This operation requires movement of nodes, which generally entails a violation of the precedence order. As a consequence, the monotonicity of the process of generating models from syntactic descriptions collapses. If we try to relax the precedence order, we obtain valid descriptions that correspond to non correct Lambek proof nets where some axiom links cross each other.

The fundamental reason of this difficulty lies in the intimate interweaving between the precedence and dominance orders in Lambek proof nets. The construction of a Lambek proof net can be viewed as the construction of an ordered tree from a syntactic description under the control of both dominance and precedence order. Whereas the initial dominance order is preserved in the final tree, this is not the case for the precedence order: it is only preserved between the children of negative nodes; for the rest, this order is used for bounding the movement of dual nodes in terms of good parenthesising, which corresponds to the planarity of Lambek proof nets.

2. Polarised tree descriptions: a framework for developing grammatical formalisms

2.1. Outline of Interaction Grammars

Even if Lambek Grammars (LG) do not fit in exactly with the framework of polarised tree descriptions, as we have just pointed out, their application to linguistics shows that this framework captures the essentials; the generation of syntactic trees driven by a mechanism of polarities from descriptions which use three kinds of relations: dominance, immediate dominance and precedence.

Concerning TAG, Vijay-Shanker (VS92) proposes their translation in terms of tree descriptions which can be completed with polarities for exactly recovering the shape defined in the previous section. This common shape highlights the main difference between TAG and LG: both definition and realisation of dominance relations are more constrained in TAG than in LG. Two nodes which participate in such a relation must have the same grammatical category in TAG and the relation can only be realized by insertion of another syntactic description between the two nodes, whereas, in LG, the only constraints are polarity and good parenthesising constraints.
By exploiting tree descriptions, some works aim to relax the TAG adjunction operation in order to capture linguistic phenomena which are beyond TAG (RVSW95; Kal99). Unfortunately, the counterpart of a more flexible framework is often over-generation and a loss of computational efficiency in the absence of control on the process of syntactic composition. IG are an attempt of exploiting the flexibility of tree descriptions as far as possible while keeping the notion of polarity as central for controlling syntactic composition.
A particular interaction grammar G, which is associated with a vocabulary $\mathcal{V}$, is defined from a finite set of labels $\mathcal{C}$, which can be in a first approach a set of atomic categories. The basic objects of G are IG syntactic descriptions which are a variant of IILL syntactic descriptions.

Definition 2.1 *An IG syntactic description is a finite set of nodes structured by dominance, immediate dominance and precedence relations. Immediate dominance is defined in two ways: either classically with a binary relation between two nodes or with a parent-children relation which enumerate all children of a node. Every node is equipped with a label from* $\mathcal{C}$ *and a polarity −1, 0 or +1.*

IG are lexicalised so that G is completely defined by its lexicon which associates a set of syntactic descriptions to every word of $\mathcal{V}$.
With respect to the abstract IILL syntactic descriptions, IG descriptions present three differences: the use of precedence order in addition to dominance orders, the presence of neutral nodes and the possibility of closing the set of children for a node. These differences are reflected in the definition of a model.

Definition 2.2 *A model of an IG syntactic description D is an ordered tree T such that there exists an interpretation I which respects the following conditions:*

- *every node of the syntactic tree interprets a set of node variables labelled with the same labels; all these variables are neutral, otherwise, there is exactly one positive and one negative variables in this set;*
- *the interpretation respects dominance and precedence relations of D and the tree structure of T is totally realised by means of parent-children relations initially present in the description.*

IG differ from LG on two main points: precedence order between syntactic constituents is dissociated from dominance order and neutral nodes are used for pattern matching between syntactic structures. In this way, parsing amounts to generating models from syntactic descriptions and a parsing process can be viewed as an electrostatic process in which opposite charges attract themselves while charges with the same polarity repel each other, whence the name of Interaction Grammars.

Example 2.1 *Parsing the phrase* Marie que Jean voit *starts with extracting appropriate syntactic descriptions from a lexicon for all its words and gathering them in a unique syntactic description as Figure 2 shows it. The root of the description represents the request whereas* *each of its four children corresponds to a word of the phrase. Every syntactic node is labelled with its grammatical category and its polarity (polarity 0 is omitted). Contrary to ILL descriptions, we choose the opposite convention for polarities, which is better suited to linguistic reality: positive nodes represent actual constituents and negative nodes virtual constituents which are expected. Precedence order between syntactic nodes is denoted with dotted arrays and dominance order with dotted edges. Parsing succeeds in finding a model for this syntactic*

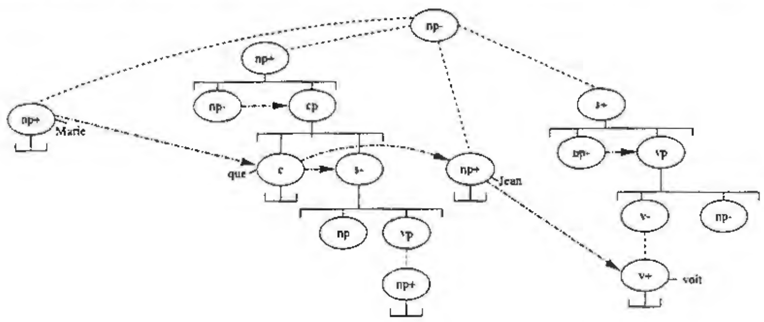

Figure 2: syntactic description of the phrase *Marie que Jean voit*

description: this model is the syntactic tree given by Figure 3.

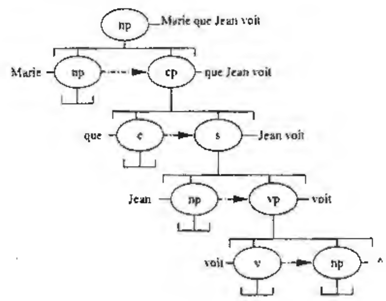

Figure 3: syntactic tree of the phrase *Marie que Jean voit*

In this first version, IG go beyond the expressivity of LG (for instance, middle extraction from relative clauses is representable in such a framework) but they are still too rigid.

2.2. *Polarised features and non-determinism in descriptions*

The outline of IG that was just presented encounters similar limitations to TAG for expressing the flexibility of word order in natural languages. For instance, the SVO order is sometimes relaxed like in the phrase *Marie que voit Jean*: the object of *voit* is provided by the relative pronoun *que* the form of which indicates the accusative case. As a consequence, there is no more ambiguity on the assignment of the subject and the object of *voit* and word order can be relaxed. Nevertheless, the phrase *Marie que voit il* is not acceptable because the position of the clitic *il* after the verb *voit* generates an interrogative type for the clause *que voit il*. Such a complex interaction between word order and grammatical features is not captured by the previous version of IG.

Another difficulty comes from the fact that a word can be used in several syntactic contexts which often differ only partially. For instance, the verb *voit* can be used without any object like

in the interrogative sentence *Jean voit-il ?*.
A answer to these problems consists in a refinement of our formalism on two main points:

- polarities are transferred from syntactic constituents to elementary features that are used for describing their properties, which gives a finer granularity to this notion;
- syntactic descriptions can be composed in two ways: in a product of two descriptions, the resources of both components are used whereas in a sum, either the resources of the first component or the resources of the second are used, but not both.

These refinements make the notion of model more complex: the neutrality condition refers now to features and not to nodes and for every choice point in a description exactly one alternative is used in a model.
Because of lack of space, we do not present the IG formal system in its complete shape; we prefer to give an example for illustrating the last refinements of IG. This formal system uses the framework of multiplicative and additive linear logic (MALL): "electrostatic" interaction is expressed by the resource sensitivity of MALL and non-determinism in descriptions by the additive part of this logic.

Example 2.2 *Figure 4 present the possible lexical entries for* Jean, voit *and* il. *Every entry is a combination of partial descriptions organized in a hierarchy according to a decision tree. Every node of this tree is a choice point between two partial descriptions. For instance, the lexical entry of* voit *includes two choice points: the left corresponds to the presence or not of an order subject-verb and the right to the presence or not of an object for the verb. All possible complete descriptions are built by making a choice at each choice point and by superposing all remaining descriptions. In this way, a single entry expresses four syntactic contexts for the verb* voit. *The entry for* il *also includes two choice points corresponding to the presence or not of an explicit subject in the sentence and to the order clitic-verb.*
Positive, negative and neutral features are respectively denoted $\rightarrow$, $\leftarrow$ *and* $=$. *A polarity which is not followed by a value means that this value is non determined. To remain readable, the figure includes only the most significant features of every node. With these entries, we succeed in parsing* il voit Jean, voit-il Jean ? *and* Jean voit-il ? *at once.*

The price for having a flexible model is a loss of computational efficiency but the monotonicity of the model generation process allows us to use the powerful tool of constraint solving for computing models from syntactic descriptions. Such an approach was inspired from the proposals of (DT99) and it gave rise to the implementation of a prototype in the constraint programming language *Oz* (Smo95). The first experiments show that polarities play a decisive role for computational efficiency and further validate our direction of research: exploiting in a same linguistic model the advantages of both under-specification and polarities.

References

V. M. Abrusci, C. Fouqueré, and J. Vauzeilles. Tree adjoining grammars in noncommutative linear logic. In C. Retoré, editor, *LACL'96. Nancy, France, September 1996*, volume 1328 of *LNCS*, pages 96–117. Springer Verlag, 1997.

P. Blache. *"Contraintes et théories linguistiques : des Grammaires d'Unification aux Grammaires de Propriétés"*. Thèse d'habilitation, Université Paris 7, 1999.

M.-H Candito. *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées. Application au français et à l'italien,*. Thèse de doctorat, Université Paris 7, 1999.

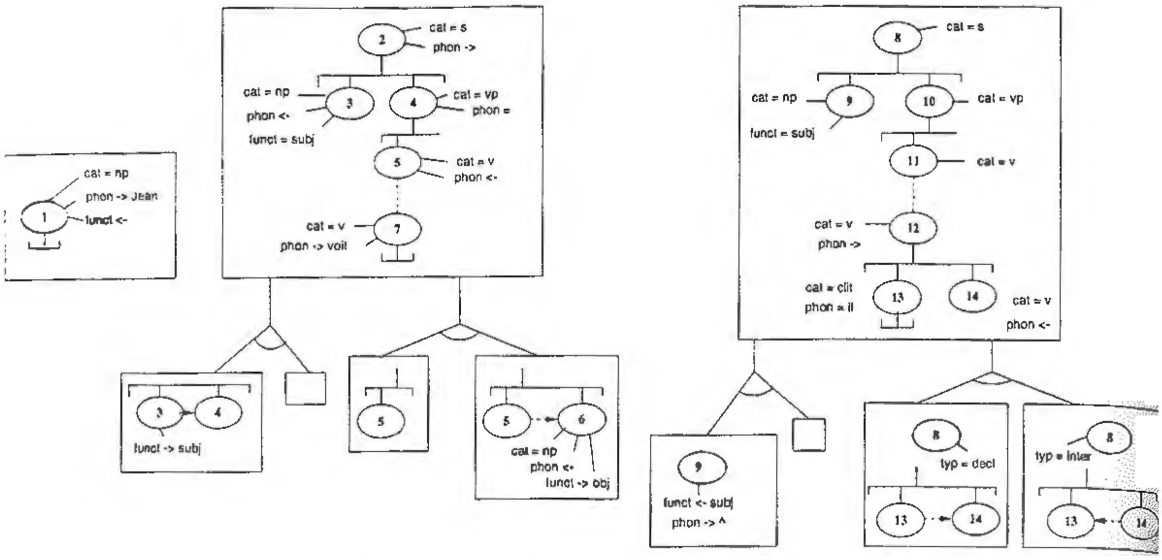

Figure 4: Lexical entries for *Jean*, *voit* and *il*

D. Duchier and S. Thater. Parsing with tree descriptions: a constraint based approach. In *NLULP'99,Dec 1999, Las Cruces, New Mexico*, 1999.

M. Egg, J. Niehren, P. Ruhrberg, and F. Xu. Constraints over lambda structures in semantic underspecification. In *COLING/ACL'98, Montreal,Quebec, Canada. August 1998*, 1998.

J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

J. M. E. Hyland and L. Ong. Dialogue games and innocent strategies: an approach to intensional full abstraction for PCF. Manuscript, July 1993.

A. Joshi and S. Kulick. Partial proof trees as building blocks for a categorial grammar. *Linguistics and Philosophy*, 20(6):637–667, 1997.

L. Kallmeyer. *Tree Description Grammars and Underspecified Representations*. PhD thesis, Universität Tübingen, 1999.

F. Lamarche. From proof nets to games. *Electronic Notes in Theoretical Computer Science*, 3, 1996. Special Issue of Linear Logic'96, Tokyo Meeting, march 1996.

R. Muskens and E. Krahmer. Talking about trees and truth-conditions. In *LACL'98. Grenoble, France,December 1998*.

M. Moortgart. Categorial Type Logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, chapter 2. Elsevier, 1996.

D. Roorda. Resource Logics. Proof-Theoritical Investigations. Phd Thesis, Universiteit van Amsterdam, 1991.

O. Rambow, K. Vijay-Shanker, and D. Weir. D-tree grammars. In *ACL'95*, pages 151–158, 1995.

Gert Smolka. The Oz programming model. In Jan van Leeuwen, editor, *Computer Science Today*, Lecture Notes in Computer Science, vol. 1000, pages 324–343. Springer-Verlag, Berlin, 1995.

K. Vijay-Shanker. Using description of trees in a tree adjoining grammar. *Computational Linguistics*, 18(4):481–517, 1992.