

Towards efficient parsing with proof-nets

Alain Lecomte
GRIL
Université Blaise Pascal
63037-Clermont-Ferrand
France
email: lecomte@shm.grenet.fr

Abstract

This paper presents a method for parsing associative Lambek grammars based on graph-theoretic properties. Connection graphs, which are a simplified version of proof-nets, are actually a mere conservative extension of the earlier method of syntactic connexion, discovered by Ajduckiewicz [1935]. The method amounts to find alternating spanning trees in graphs. A sketch of an algorithm for finding such a tree is provided. Interesting properties of time-complexity for this method are expected. It has some similarities with chart-parsing ([König, 1991, 1992], [Hepple, 1992]) but is different at least in the fact that intervals are here edges and words are vertices (or trees) instead of the contrary in classical chart-parsing.

1 Introduction

In this paper, we present a method for parsing Lambek grammars based on graph-theoretic properties. We expect that it may be done efficiently by an algorithm (time-polynomial even in the worst case) which aims at finding an alternating spanning tree in a graph. We do not give the explicit formulation of such an algorithm in this paper: we will only give an idea and an illustration of it. This paper is thus mostly devoted to the properties on which the method is based. We call *connection graphs* the special kind of proof-nets we explore, just in order to make explicit some difference with the usual method of proof-nets, as it can be found in [Roorda, 1991; 1992] and [Moortgat 1992], but the two concepts are very similar. In many respects, connection graphs are a mere conservative extension of

the earlier method of syntactic connection, discovered by Ajduckiewicz [1935]. The method amounts to link the nodes of an ordered sequence of trees in such a way that properties of connexion, "non overlapping", acyclicity and "strong connectivity" are verified. Connection graphs are simpler than proof-nets in that they loose some information. As they are here formulated, they are only convenient for the *associative* version of the product-free Lambek calculus. One of their advantages lies in the geometrical viewpoint they provide on the proofs of a sequent. By means of this viewpoint, questions of provability may be reduced to graph-theoretical problems. For instance, every *reading* of a sentence is provided by an *alternating spanning tree*.

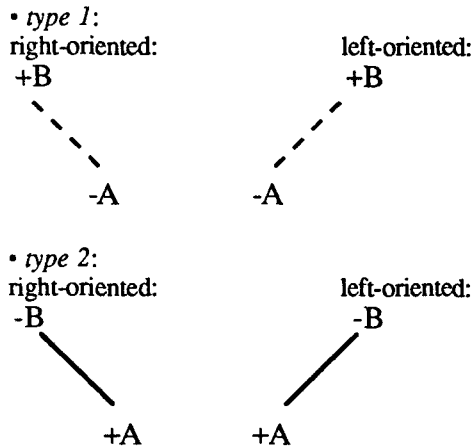
In many aspects, the method resembles the well known method of *chart-parsing*. König [1991, 1992] was the first to apply chart-parsing to Lambek calculus. Hepple [1992] has recently improved this application. An obvious difference with the method proposed here lies in the fact that, in ours, words are points and intervals between them are edges instead of the contrary in chart-parsing. In both cases, computational advantages are found by keeping correct partial analyses after they have been obtained. A chart is actually used in both methods.

2 Connection Graphs

2.1 Links and Nodes

Definition 1: Let S be a set of signed vertices (i-e: labelled with letters belonging to an alphabet A and with a + or - sign). We define three types of links:

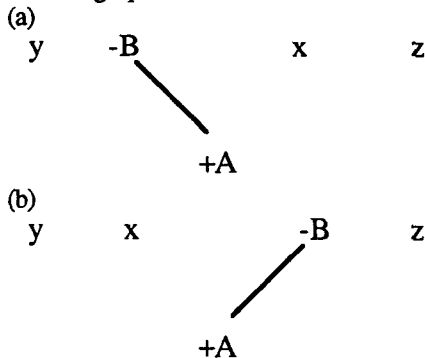
- *external links:*
+a ----- -a or -a ----- +a, between complementary vertices (same letter and opposite signs)
- *internal links:*



2.2 Connection Graphs (inductive definition)

We define by induction the class of *connection graphs* and the associated notions of *unit* and of *linking a sequence of units*.

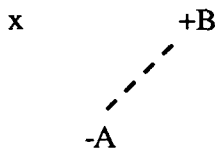
- every external link is a connection graph, which links the units $+a$ and $-a$,
- (I) if α is a connection graph which links a sequence of units x and a unit $+A$ and if β is a connection graph which links $y, -B$ and z (where $z = z' + C$)¹, then the following schemes define new connection graphs:



they will be noted respectively: $\alpha \oplus^r \beta$ and $\alpha \oplus^l \beta$, and they link respectively: $y, -(B/A), x, z$ and $y, x, -(A/B), z$ where $-(B/A)$ and $-(A/B)$ are new units.

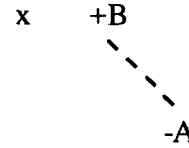
- (II) if α is a connection graph which links $-A, x$ and $+B$, then:

(a) if $-A$ is the left end of the linking, we get a new connection graph which links x and $+(A/B)$ by the scheme:



¹ We use small latin letters for sequences of units and capital latin letters with a sign for units.

(b) if $-A$ is the right end just before $+B$, then we get a new connection graph which links x and $+(B/A)$ by the scheme:

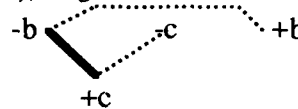


they are respectively noted: $t^l(\alpha)$ and $t^r(\alpha)$.

Example:

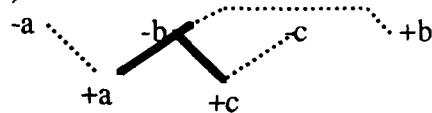
$-b \text{-----} +b$ is a connection graph which links $-b$ and $+b$, idem for $-c \text{-----} +c$.

By (Ib), we get:



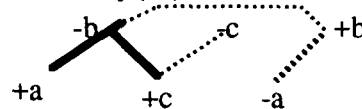
which is a connection graph which links $-(b/c), -c$ and $+b$

$-a \text{-----} +a$ is also a connection graph and we obtain by (Ia):



which is a connection graph which links $-a, -(a(b/c)), -c$ and $+b$

and we obtain by (IIa):



which links $-(a(b/c)), -c$ and $+(ab)$.

Proposition 1: for any sign \pm , we have for all A, B and C :

$$\pm (A/B)/C = \pm A/(B/C)$$

Proof: assume we have a connection graph β which links $y, -(A/B), z$ and a connection graph α which links x and $+C$, then, by (Ia) we obtain a connection graph which links $y, -(A/B)/C, x$ and z . But since β links units having already a type 2 link, it necessarily comes from a $[y', -B, z]$ and a $[x', +A]$ such that $y' x' = y$. From $[y', -B, z]$ and $[x, +C]$ we obtain a connection graph which links $y', -(B/C), x$ and z and from this graph and the graph which links $[x', +A]$, we get a graph which links $y', x', -(A/B/C), x$ and z , that means a graph which links $y, -(A/B/C), x$ and z , which is identical to the graph which links $y, -(A/B)/C, x$ and z . \diamond

2.3 Alternating trees

Definition 2: Let $L1$ and $L2$ respectively the sets of type 1 links and type 2 links. An alternating tree on $L1 \cup L2$ is a tree in which all the nodes at a same level have the same sign, all the edges are type 1 or type 2 links and the sign of a node is alternatively $+$ and $-$ along any path from the root to any leaf.

Proposition 2: Let G be a connection graph and E be the set of its external links. The set $G-E$ is an ordered set of units, each of them consisting in an alternating tree, noted $-A$ if the root is negative and $+A$ if the root is positive.

Proof: by induction on building a connection graph.

1) Let G consist in a single external link, if we remove the external link, we get two distinct vertices: $+a$ and $-a$, which are alternating trees.

2) Let us assume the property true for β which links $y, -B, z$ and for α which links x and $+A, -B$, as a unit, is an alternating negative tree and $+A$ is an alternating positive tree. By (1a) and (1b), a type 2 link is added from the root of $-B$. We thus keep a tree the root of which is negative and the type of the added link is the same as that of the links at the same level. Moreover, no cycle is added because before this operation, the two graphs were not already connected. $-(B/A)$ and $-(A \setminus B)$ are thus alternating trees.

3) Let us assume now the property true for α which links $-A, x$ and $+B$, then it is also true for $t^1(\alpha)$ and $t^f(\alpha)$ because a type 1 link is added from the positive root of $+B$. Obviously, no cycle is added when we exclude the external links. \diamond

2.4 Numbering the nodes of a connection graph

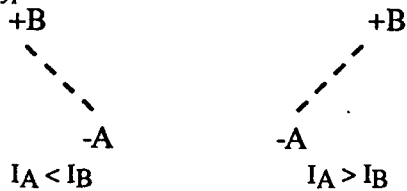
Let Γ be the ordered set of alternating trees in a connection graph G .

Proposition 3: Γ contains one and only one positive tree. It is the last tree of the set. Its root will be called the positive root of G .

Proof: very easy, by induction on building a connection graph. \diamond

Proposition 4: Let us assume that G contains $2n$ vertices. There is one and only one way of numbering these vertices in order that the following conditions are fulfilled:

- if X and Y are alternating trees and $X < Y$ (X before Y in the order defined on Γ) the set I_X of numbers associated to X and the set I_Y are such that: $I_X < I_Y$ (where $I < I'$ means: $\forall i \forall i' i \in I \text{ and } i' \in I' \Rightarrow i < i'$)
- type 1 links:



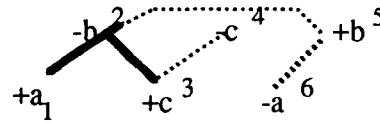
- type 2 links:



$$I_B < I_A \qquad I_A < I_B$$

Proof: easy. (cf one of the several ways of enumerating the nodes of a tree). \diamond

Example:



Definition 3: a connection graph G is said to be *well numbered* if and only if its nodes are numbered according to Proposition 4.

2.5 Completeness of Connection Graphs with respect to the Associative Product-free Lambek Calculus

We show that every deduction d in the calculus A (for Associative Product-free Lambek calculus) may be translated into a connection graph $\chi(d)$.

axiom: $a \rightarrow a$ is translated into:

$$-a \text{-----} +a \text{ or } +a \text{-----} -a$$

rules:

[L/]: if $x \rightarrow A$ translates into β and $y B z \rightarrow C$

translates into γ , $y B/A x z$ translates into $\beta \oplus^f \gamma$

[L]: $y x A \setminus B z$ translates into $\beta \oplus^1 \gamma$

[R/]: translates into $t^f(\alpha)$ where α is the translation of $A x \rightarrow B$

[R\]: translates into $t^1(\alpha)$. \diamond

Remark: this translation is not a one-to-one mapping, because several deductions can be translated into the same connection graph. We assume here that connection graphs provide a semantics for derivations. It is possible to show that this semantics is isomorphic to the associative directed lambda calculus (see Wansing 1990).

2.6 Soundness of Connection Graphs with respect to A

This paragraph is very similar to Roorda 1991, chap III, § 4.

Lemma 1: If we remove a type 1 link from a connection graph G , we keep a connection graph.

Proof: we may assume that this link has been added at the last stage of the construction. \diamond

Definition 4: a type 2 link is called *separable* if it could have been added in the last stage of the construction.

Lemma 2: If a connection graph, consisting of more than one link, has no terminal type 1 link, it has a separable type 2 link.

Proof: obvious.

Proposition 5: To every connection graph G the units of which are: $-A_1, -A_2, \dots, -A_n, +B$, there corresponds a deduction in A of the sequent:

$$A_1^*, A_2^*, \dots, A_n^* \rightarrow B^*$$

(where X^* is the formula associated with the alternating tree X)

Proof: by induction on the structure of G . G has necessarily a last link, in the order of the construction. As seen in the previous lemma, it is necessarily either a type 1 link or a type 2 link. In the first case, when removing it, we still have a connection graph. In the second case, when removing it, we get two connection graphs α and β which correspond, by induction hypothesis, respectively to $x \rightarrow A$ and $y B z \rightarrow C$.

2.7 Non-Overlapping property

Definition 5: given a connection graph G , we call *interval* every set of integers $[i, j]$ (ie: $\{x; i \leq x \leq j\}$) such that i and j are indices associated with ending points of an external link (and $i < j$).

Two intervals $[i, j]$ and $[i', j']$ *do not overlap* if and only if:

- $[i, j] \cap [i', j'] = \emptyset$
- or • $[i, j] \supset [i', j']$ and $i \neq i'$ and $j \neq j'$
- or • $[i', j'] \supset [i, j]$ and $i \neq i'$ and $j \neq j'$

Given a family I of intervals, we say that it *satisfies the Non Overlapping Condition (NOC)* if it does not contain any pair of intervals which overlap.

Theorem 1: in a well numbered connection graph G , the family of intervals associated with all the external links satisfies NOC.

Proof: easy, by induction. \diamond

2.8 Linking the positive root

Theorem 2: in a connection graph G , the positive root is connected by an external link either to a negative vertex in the same tree (just below it) or to a negative root.

2.9 Connectivity and acyclicity by external links and type 2 links

Theorem 3: Let G be a connection graph. Let $L1$ be the set of its type 1 links. $G-L1$ is connected and acyclic (it is a tree).

Proof: a type 2 link connects two connection graphs for the first time they meet and a type 1 link does neither connect two graphs, nor modify the topology of type 2 links and external links. \diamond

2.10 One-to-one mapping between nodes

Theorem 4: for every i in a connection graph G , let $\phi(i)$ be the node linked to i by an external link, ϕ is a one-to-one mapping from S onto S .

Proof: trivial by induction. \diamond

2.11 Strong connectivity

Definition 6: given a graph G , a *spanning tree* of G is defined as a tree on the complete set of nodes of G . A tree is said to be *alternating* on $L2 \cup E$, if each of its

paths from the root to a leaf is alternatively composed by $L2$ -edges and E -edges.

Theorem 5: every connection graph G admits an alternating spanning tree with the positive root of G as the root.

Proof:

- true for any axiom,
- Let us assume it is true for α and γ . Then by (Ia) and (Ib):

• *by induction hypothesis*, there is a path from the root of $+C$ to the root of $-B$ which is alternating. Since it arrives at a negative vertex, its last link cannot be of type 2, then it is an external link.

• There is also a path from the root of $+A$ to any leaf of the spanning tree of α , which is alternating. Since it comes from a positive vertex, it cannot begin with a type 2 link, hence it begins with an external link.

Thus, by inserting a type 2 link between the external link arriving at $-B$ and the external link starting from $+A$, we get a path starting from the positive root of $+C$ and arriving at any leaf of α , inserted into γ , which is alternating.

Therefore, there is an alternating path from the positive root of $+C$ to any leaf of $\alpha \oplus \gamma$.

Let us assume now it is true for α which links $-A$, x and $+B$. The transformation t^r or t^l does not modify the set of paths starting from the positive root of $+B$. \diamond

Definition 7: a node in a connection graph G will be said *strongly connected* to another node in the same graph if they are connected by an alternating path.

Definition 8: a link will be said to be *strong* if its two ends are strongly connected.

Theorem 6: in a connection graph G , every type 1 link is strong.

Proof: this is shown when installing a new type 1 link. Such an installation does not modify the topology of $G-L1$. The previous graph (before applying t^l or t^r) was necessarily a connection graph. Thus by Theorem 5, it was scanned by an alternating spanning tree with as root the positive root of the graph. This tree is preserved by t^l or t^r , it contained an alternating path connecting the two vertices which are now linked by a type 1 link. \diamond

As a matter of recapitulation, we enumerate now the following properties, satisfied by any connection graph.

- one-to-one mapping by external links (CG0)
- positive root property + uniqueness of the positive root (CG1)
- non-overlapping condition (CG2)
- strong connectivity (CG3)
- connectivity and acyclicity on $L2 \cup E$ (CG4)
- alternating spanning tree (CG5)
- $G-E$ is a set of well numbered alternating trees (CG6)

Proposition 6: CG5 is a consequence of CG0, CG1, CG3, CG4.

Proof: By CG4, G-L1 is a tree on S, it is therefore a spanning tree of G. Let us consider a path σ from the positive root +b (which is the root of the positive tree +B, and which is unique according to CG1) to a leaf a. We must notice that a cannot be positive, because if it was, it would necessarily be an end of a type 2 link and this type 2 link would be the last edge on the path σ , but by CG0, it would be linked by an external link to another node and thus it would not be a leaf. Thus, a is necessarily negative, and we can write -a instead of a. If -a is isolated (as a negative root of a negative tree), we can remove the last external link and the type 2 link before the last, we are led to the same problem: a path σ' arriving at a negative leaf, but σ' is shorter than σ .

If -a is not isolated, it is necessarily the end of a type 1 link, but by CG3, there is an alternating path joining -a and the positive node +c which is the other end. Removing this path and the type 2 link arriving at +c, we still get the same problem of a path σ' arriving at a negative node, but again σ' is shorter than σ . We can proceed like that until we have a mere external link between the positive root +b and a vertex -b. In this case, the path is obviously alternate.

3 Well Linked Graphs (WLG)

Definition 9: Let $-A_1, -A_2, \dots, -A_n, +B$ a sequence of alternating trees on the set S of signed vertices. We call Well Linked Graph on $[-A_1, -A_2, \dots, -A_n, +B]$ the result of adding external links in order that CG0, CG1, CG2, CG3, CG4 are satisfied.

Proposition 7: every connection graph is a well linked graph.

Proof: obvious according to the previous §.0

Theorem 7: every well linked graph is a connection graph.

(ie: every well linked graph could be obtained by the inductive construction of a connection graph, with the sequence of alternating trees as G-E).

Proof: given a WLG on $[-A_1, -A_2, \dots, -A_n, +B]$, it has a unique positive root +b (the root of +B). Thus it satisfies the property of uniqueness of the positive root. Let us assume there is a type 1 link from +b, then let us remove:

- if it is left-oriented: the leftmost one
- if it is right-oriented: the rightmost one.

Let us assume for instance that it is left-oriented:

• The tree below this link may be moved towards the left end of the sequence of trees by the inverse of the construction rule (IIa). This move preserves the topological structure of EUL2, therefore, CG1, CG3 and CG4 are preserved. This move implies a re-numbering but it does not destroy the non-overlapping property. Thus CG2 is preserved. CG0 is trivially preserved. The argument is similar for a right-oriented link. Thus after this removal, we keep a WLG.

Let us assume now there is no type 1 link from +b. Then there is an external link which links +b to a vertex -b situated among the negative trees. If -b is not related to another node, we get an elementary WLG: $-b \text{ ----} +b$, which is obviously a connection graph. If -b is related to another node, then by CG5, either -b is a leaf, or it is the starting point of a type 2 link. Let us assume -b is a leaf (of a non atomic tree), then -b is linked by a type 1 link to a vertex +a (and not to +b since we have assumed there is no longer type 1 link from +b). Because of CG3, -b and +a are connected by an alternating path on EUL2, thus -b is necessarily the starting point of a type 2 link, but in this case, -b is not a leaf. Therefore -b is not a leaf and it is the starting point of a type 2 link. Let +c the other end of this link.

• Let us assume that this link is left-oriented: we remove the leftmost one if many. In this case, the scanning tree is broken into two parts and the connection graph is also separated into two pieces. One contains +b, the other contains +c.

Let us consider the first one:

• it keeps CG3 and CG4:

for example CG3:

- let us consider a type 1 link situated in this part. It does not come from +b since we have assumed there is no longer type 1 link from +b.

- its ends are linked by an alternating path. Let us assume that the removed type 2 link belonged to this path. By removing it, we get either a single external link: $-b \text{ ----} +b$, but such a piece does not contain any type 1 link, or another kind of graph. If we want this graph has a type 1 link, it necessarily must contain another type 2 link starting from -b, and arriving, say, at +d. But an alternating path between two ends of a type 1 link can neither arrive by an external link at -b since -b is already connected by such a link to the positive root +b (and we have assumed there is no type 1 link attached to +b), nor pass through +d since, in this case, the path would have two consecutive type 2 links, which contradicts the definition of an alternating path. Therefore, the removed type 2 link cannot be on the alternating path linking the ends of a type 1 link in this part of the graph. Finally, no alternating path in the first component is destroyed by this removal, among all the alternating paths connecting ends of type 1 links.

Let us consider the second one:

- let us consider a type 1 link situated in this part and let us assume that its ends are linked by an alternating path passing through the removed type 2 link. The proof is the same as previously: the path can neither arrive at -b by an external link nor by a type 2 link. Moreover, it has one and only one positive root +c, because it does not contain +b, and +c is necessarily linked by an external link to either a negative root or a negative vertex just below it (if not, there would be a type 1 link $+x \text{ ---} -c$, with -c externally linked to +c, the alternating path from -c to +x would thus necessarily pass through +c and -b, which is impossible according to the first part of the proof).

When all the type 2 links attached to $-b$ are removed, there remains only the external link $-b \text{ ---- } +b$ which is a WLG, and we can perform this decomposition for each part resulting from a previous step.

It would then be possible to reconstruct the graph according to the induction schemes (I) and (II), starting only from axioms.

Corollary: well linked graphs are sound and complete with respect to the calculus A.

4 Method of construction of a well linked graph

An alternating tree was defined by a set of signed vertices and a set of typed links which link them. We are now adding two new kinds of entity in order to facilitate tree-encoding.

4.1 Colours and anti-colours

4.1.1 Colours

Let us assign to each vertex in a sequence of trees $[-A1, -A2, \dots, -An, +B]$ a colour (originally unknown and represented by a free variable X) in order that:

- two nodes linked by a type 2 link have same colours
- two nodes which are not linked or which are linked by a type-1 link have not the same colours ($X \neq Y$).

Proposition 8: for every connection graph G with set of type 1 links L1, the connectivity and acyclicity of $G - L1$ translates into: every external link links two nodes having different colours. After linking by an external link, the two colours are equalized ($X = Y$).

4.1.2 Anticolours

Anticolours are assigned to nodes in an alternating tree in order that:

- two nodes linked by a type 1 link have same anticoulour,
- if a positive node receives an anticoulour α , (by (a) or by an external link), the negated anticoulour $\neg\alpha$ is transmitted to all other positive nodes having same colour.

Rule:

- When joining two nodes by an external link, which are associated with different (positive) anticoulours α and β , α and β are said to be equalized, that means: put in a same multi-set.
- When joining a node having a negated anticoulour $\neg\beta$ to a node having a colour X by an external link, the anticoulour $\neg\beta$ is transmitted to the colour X as a label.
- When linking two ends of a type 1 link by external links, the two occurrences of the same (positive) anticoulour α must meet only one colour, or two colours which have been already equalized and such that one of the two is not labelled by a negated anticoulour $\neg\beta$ if β is an anticoulour already equalized to α .

Proposition 9: in a connection graph G, the strong connectivity translates into: the anticoulour proper to a

type 1 link meets only one colour (or colours which have been equalized).

Corollary: Every connection graph verifies: CG0, CG1, CG2, CG3', CG4', CG5', CG6 where: CG3' is the condition on unifying anti-colours, CG4' the conditions on colours, CG5' the fact that any connection graph is monocoloured.

4.2 Method

4.2.1 Categories

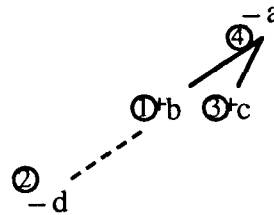
Definition 9: We call a category any set of 6-tuples each consisting in:

- a label taken from an alphabet A,
- a sign (+ or -)
- an index (integer),
- a colour (free variable)
- an anticoulour (free variable of a second sort)
- the indication of being a root if it is the case.

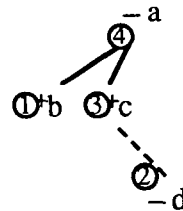
Definition 10: We call an ordered category a category where 6-tuples are ordered according to their index.

Proposition 10: each alternating tree has one and only one encoding into an ordered category.

Examples:



translates into: $\{ \langle +, b, 1, X, U, _ \rangle, \langle -, d, 2, Y, U, _ \rangle, \langle +, c, 3, X, \neg U, _ \rangle, \langle -, a, 4, X, _ , r \rangle \}$

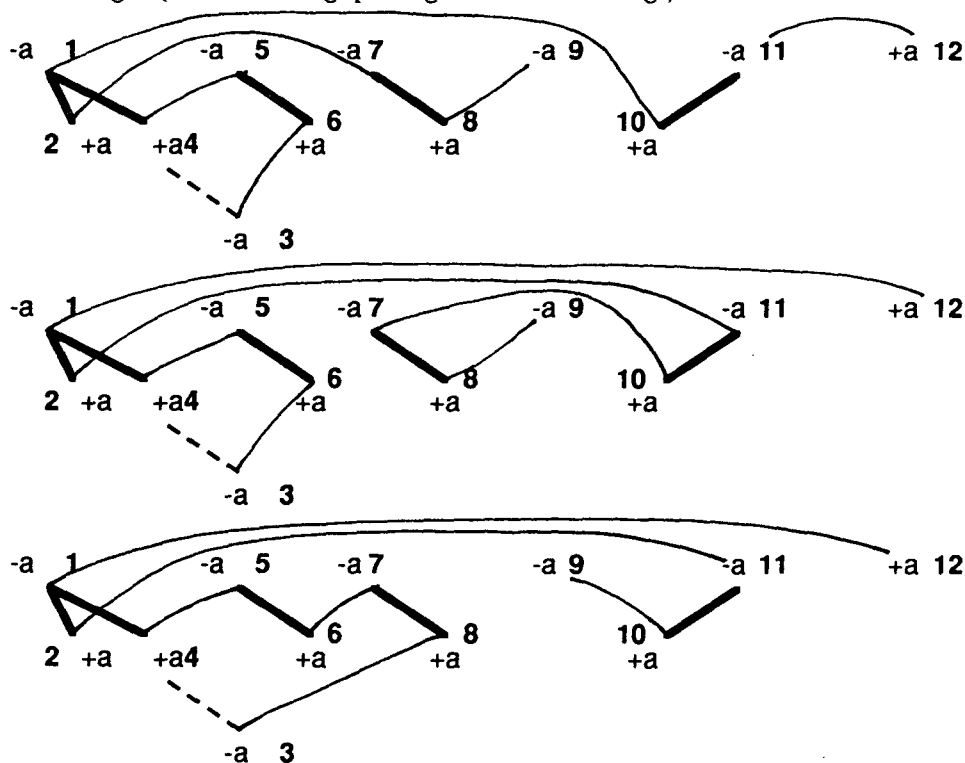


translates into: $\{ \langle +, b, 1, X, \neg U, _ \rangle, \langle -, d, 2, Y, U, _ \rangle, \langle +, c, 3, X, U, _ \rangle, \langle -, a, 4, X, _ , r \rangle \}$

Definition 11: two 6-tuples are said to be mergeable if:

- they have same literal label,
- they have opposite signs,
- they have different colours,
- if one of them has an anticoulour α , the other must not have a colour which has been labelled by a negated anticoulour $\neg\beta$ such that α and β have already been equalized, in a same multiset.
- if one node is the positive root, the other is a negative root or a negative node just below it in the same tree (same anticoulour).

fig 1: (three alternating spanning trees = three readings)



4-2-2 A sketch of an algorithm

We scan the ordered list of nodes from left to right, creating links at each step, between the current node and all the possible mergeable nodes on its left or just shifting. When nodes are shifted, they are pushed onto a stack. Links are recorded on the chart in the following way. Each link is a node of the chart (in consequence, the chart has no more than n^2 nodes, where n is the number of nodes on the reading tape R). A link l is joined by arcs in the chart to all links already recorded l_1, \dots, l_n such that l makes a correct partial linking by insertion into the linkings represented by the paths arriving at l_1, \dots, l_n , and l has a left extremity which coincides with either the rightmost right extremity of a link already recorded on such a path, or with a top of stack attached to such a previously recorded link. Thus, a link l may be an arriving point for several paths. In this case, we will consider l as a new starting point. That means that when joining a new link l' to links above l in the chart, we only test the correctness of a partial linking down to the link l . We consider here that if l is in the chart at this place, there is necessarily a correct path up to it, and all the partial paths from l to the current node are, by definition, correct. Thus, when adding a link above l (and before a possible new "crossroads"), even if there are many paths joining l to it, there is at least one correct path from the bottom of the chart to the current node. Each time a link is recorded and joined to other ones, we record for each arc arriving at it, the possible tops of stack, the possible

rightmost right extremities, the list of nodes through which the path has passed since the previous embranchment, the list of equalized colours (possibly labelled with negated anticoulores) and the list of equalized anticoulores (for the piece of path coming from the previous embranchment). When joining a new link, we have to retrieve a new top of the stack, if added by consuming a previous one or a previous rightmost right extremity, and to test the correctness of the path. This necessitates a descent along paths down to the bottom of the chart. This descent is made deterministic because of the informations stored on arcs. If n is the number of nodes in the original sequent, a maximum of n^2 links may be created, and there can be a maximum of n^4 arcs in the chart. At step i , there can be a maximum of i^4 arcs. We add new links on the basis of stack informations stored on arriving arcs at each previously recorded link. Each checking does not take more than i steps, and there are at most i^2 nodes to check at step i . For a given link to add, when looking for new tops of the stack and checking the correctness of the new linking, we explore the current state of the chart by scanning no more than twice (one in one direction, one in the other) each arc it contains. Thus joining a new link to previous ones entails a maximum of $2i^7$ steps. At step i , i new links can be added. Thus step i entails a maximum of $2i^8$ steps. Thus, when reaching step n , we have done a maximum of $2\sum_{i=1}^n i^8$ steps ($i=1$ to n), that is $O(n^9)$. This is obviously a too big order. Nevertheless,

the method is time-polynomial and more improvements can be expected in the future.

Example:

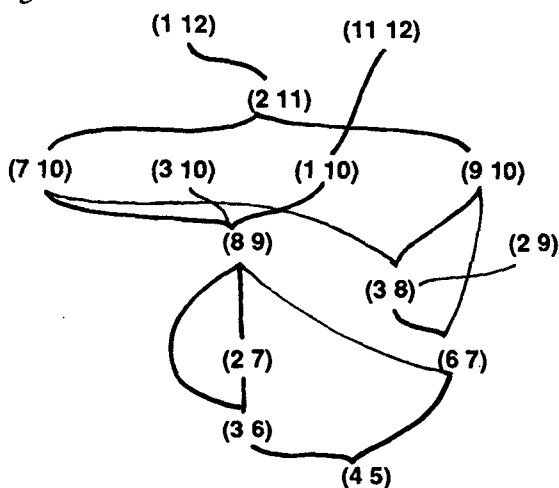
Suppose we have to demonstrate the sequent:

$$(a/a) / (a/a) \ a/a \ a/a \ a \ a \backslash a \rightarrow a$$

(cf fig 1)

At beginning steps 1, 2, 3, 4, nodes are pushed onto the top of the stack. At step 5, the link (4 5) is created and recorded in the chart. The new top of stack 3 is attached to it. At step 6, (3 6) is added (with new top of stack 2), on the top of the previous link. At step 7, (6 7) is created and joined to (4 5) (with top of stack 3) and not to (3 6) (because they have the node 6 in common). (2 7) is joined to (3 6) (with top of stack 1). At step 8, (3 8) is created and joined to (6 7) (with top of stack 2), but not to (2 7) because of anticolours (7 receives $-\beta$ and 8 receives β and they have same colour). (1 8) is not created because they have same colour. At step 9, (2 9) is created and joined to (3 8), and (8 9) is created too, but joined to (6 7) and (2 7). At step 10, (1 10) is joined to (8 9), (3 10) is joined to (8 9) and (6 7), (9 10) to (3 8). (7 10) is also joined to (8 9) and (3 8) because 7 is a rightmost right extremity in paths leading to these nodes. In such a circumstance, the node previously linked to the released right extremity, here 2 or 6 is pushed onto the stack. After that, (2 11) may be added to (7 10) and (9 10) but not to (2 11) because of anticolours. And finally, (1 12) may be joined to (2 11) and (11 12) to (1 10). By looking at the list of nodes attached to links installed in the chart, we see that these last moves lead to complete linkings. By going down to the bottom of the chart, we find the three solutions: [1 [2 [3 [4 5] 6] 7] [8 9] 10][11 12], [1 [2 [3 [4 5] 6] [7 [8 9] 10] 11] 12] and [1 [2 [3 [4 5] [6 7] 8] [9 10] 11] 12] (cf fig 2).

fig2: the final chart.



5- Conclusion

We have presented a conception of parsing essentially based on a geometrical viewpoint. It amounts to build a

correct linking of nodes in an ordered sequence of types represented as trees. Such a linking corresponds to an alternating spanning tree in a graph. We have shown that this method is sound and complete with respect to the associative product free Lambek calculus and we have given an idea of what an algorithm for finding such a spanning tree could be.

References

[Ajduckiewicz,1935] K.Ajduckiewicz. Die Syntaktische Konnexität, *Studia Philosophica* 1, 1-27, engl. transl. 'Syntactic Connexion', in S. McCall (ed)(1967), 207-231.

[Busacker and Saaty, 1965] R.Busacker and T.Saaty. *Finite Graphs and Networks, An Introduction with Applications*, McGraw-Hill, New-York.

[Hepple, 1992] Mark Hepple. Chart Parsing Lambek Grammars: Modal extensions and Incrementality, *Proceedings of COLING*, Nantes, 23-28 july.

[König, 1991] Esther König. Parsing Categorical Grammar, DYANA, deliverable 2.1.2.C., reproduced in [Lecomte, 1992a].

[König, 1992] Esther König. Chart Parsing and the Treatment of Word Order by Hypothetical Reasoning, in [Lecomte, 1992a].

[Lecomte, 1992a] Alain Lecomte (ed.). *Word Order in Categorical Grammar*, ADOSA, Clermont-Ferrand.

[Lecomte, 1992b] Alain Lecomte. Proof-Nets and Dependencies, *Proceedings of COLING*, Nantes, 23-28 july.

[Moortgat, 1992] Michael Moortgat. Labelled Deductive Systems for categorial theorem proving. *Proceedings of the 8th Amsterdam Colloquium*, Dekker & Stokhof (eds).

[Roorda, 1991] Dirk Roorda. *Resource Logics: Proof-theoretical Investigations*, PhD Thesis, Faculteit van Wiskunde en Informatica, Amsterdam.

[Roorda, 1992] Dirk Roorda. Proof Nets for Lambek Calculus, *Journal of Logic and Computation*, 2(2): 211-233.

[Wansing, 1990] Heinrich Wansing. Formulaes-as-types for a Hierarchy of Sublogics of Intuitionistic Propositional Logic. Gruppe für Logik, Wissenstheorie und Information an der Freien Universität Berlin.