Chart Parsing Multimodal Grammars

Richard Moot

Chart Parsing Multimodal Grammars

Richard Moot

University of Montpellier, CNRS (LIRMM)

1 Introduction

The short note describes the chart parser for multimodal categorial grammars which has been developed in conjunction with the type-logical treebank for French, which is described in more detail in (Moot 2010, Moot 2012, Moot 2015*b*) and which is available at (Moot 2015*a*). The chart parser itself can be downloaded as a part of Grail light at (Moot 2018).

`https://github.com/RichardMoot/GrailLight`

The chart parser is an instance of the "deductive parsing" technology of (Shieber, Schabes & Pereira 1995) and the core parsing engine of their implementation has been retained in the source coude, with only some minor modifications. I am grateful to the authors for having made their source code available.

The current chart parser was originally introduced as a preprocessing step for a proof net algorithm (Moot 2017). However, this preprocessing step turned out to be so effective that it soon handled a bit under 98% of the complete French Type-Logical Treebank and therefore it made sense to add additional chart rules to handle the remaining few percent as well (these are briefly sketched in Section 2.4, the rest of this paper focuses on the basic rules of the chart parser).

This paper presupposes the reader has at least a basic familiarity with multimodal categorial grammars (Moortgat 2010, Moortgat 2011, Moot & Retoré 2012) and with chart parsing (Shieber et al. 1995).

2 Chart rules

In this section, I will dicuss the inference rules used by the chart parser. I will start with the simplest rules and gradually introduce more detail.

2.1 AB rules

The elimination rules $/E$ and $\backslash E$ appear already in (Shieber et al. 1995). For AB grammars, the chart rules are very simple and shown in Figure 1. Chart items are tuples $\langle \Gamma, F, L, R \rangle$ where Γ is an antecedent term, F is a formula, and

$$\frac{\langle \Gamma, A/B, I, J\rangle \quad \langle \Delta, B, J, K\rangle}{\langle \Gamma \circ \Delta, A, I, K\rangle}\ /E \qquad \frac{\langle \Gamma, B, I, J\rangle \quad \langle \Delta, B\backslash A, J, K\rangle}{\langle \Gamma \circ \Delta, A, I, K\rangle}\ \backslash E$$

Figure 1: AB grammar chart rules

L and R are integers representing the leftmost and rightmost string positions respectively. The meaning of such a tuple is that we have derived a formula F, using the hypotheses in Γ, spanning exactly the positions from L on the left to R on the right.[1]

With this in mind, the chart rule for $/E$ indicates that if we have derived a a formula A/B spanning string positions $I-J$ and a formula B spanning string positions $J-K$ (that is, A/B and B are adjacent with B immediately to the right of A/B), then we can conclude that we can derive a constituent A from positions I to K (that is, the concatenation of the strings assigned to A/B and B).

Given these rules, proving an AB sequent $A_1, \ldots, A_n \vdash B$ corresponds to starting from axioms $\langle w_1, A_1, 0, 1\rangle \ldots \langle w_n, A_n, n-1, n\rangle$ and deriving the goal $\langle \Gamma, B, 0, N\rangle$ with $\mathit{yield}(\Gamma) = w_1, \ldots, w_n$. To facilitate inspection of the chart items, Γ will not be a binary tree of *formulas*, but a binary tree of the corresponding *words*. Therefore, a lexical entry for the verb "dort" (*sleeps*) with formula $np\backslash s$ at position 1-2 will correspond not to the item $\langle np\backslash s, np\backslash s, 1, 2\rangle$ but to the item $\langle \mathit{dort}, np\backslash s, 1, 2\rangle$.

Example As an example, the table below shows how the chart is filled for "Le marché financier de Paris" (*the financial market of Paris*).

	Chart Item	Justification
1	$\langle \mathit{le}, np/n, 0, 1\rangle$	Lexicon
2	$\langle \mathit{marché}, n, 1, 2\rangle$	Lexicon
3	$\langle \mathit{financier}, n\backslash n, 2, 3\rangle$	Lexicon
4	$\langle \mathit{de}, (n\backslash n)/np, 3, 4\rangle$	Lexicon
5	$\langle \mathit{Paris}, np, 4, 5\rangle$	Lexicon
6	$\langle \mathit{le} \circ \mathit{marché}, np, 0, 2\rangle$	From 1,2 by $/E$
7	$\langle \mathit{marché} \circ \mathit{financier}, n, 1, 3\rangle$	From 2,3 by $\backslash E$
8	$\langle \mathit{de} \circ \mathit{Paris}, n\backslash n, 3, 5\rangle$	From 4,5 by $/E$
9	$\langle \mathit{le} \circ (\mathit{marché} \circ \mathit{financier}), np, 0, 3\rangle$	From 1,7 by $/E$
10	$\langle (\mathit{marché} \circ \mathit{financier}) \circ (\mathit{de} \circ \mathit{Paris}), n, 1, 5\rangle$	From 7,8 by $\backslash E$
11	$\langle \mathit{le} \circ ((\mathit{marché} \circ \mathit{financier}) \circ (\mathit{de} \circ \mathit{Paris})), np, 0, 5\rangle$	From 1,10 by $/E$

The chart items are labeled from 1 to 11 indicating the order they are entered in the chart. We use a general chart parser of the type explained in (Shieber

[1] The use of pairs of string positions to represent substrings of an input string is widely used in parsing algorithms; see for example (Pereira & Shieber 1987, Shieber et al. 1995, Jurafsky & Martin 2009).

et al. 1995), so we start with an agenda containing items 1-5 (the lexical lookup for the words in the sentence) and then successively add the items of the agenda to the chart. When we add an item from the agenda to the chart, we compute all consequences according to the rules of the grammar of this item with all items already in the chart. So once item 2 is added to the chart, item 6 is added to the agenda, since it is the combination of item 2 with item 1 (already in the chart) by means of rule $/E$. Similarly, item 7 is added to the agenda when item 3 is added to the chart and item 8 is added to the agenda when item 5 is added to the chart, etc.

We complete the parse when item 11 is added to the chart. If desired, we can recover the proof by recursively finding the justification of each of the rules, going back from 11 to 1 and 10, from 10 to 7 and 8 (1 is in the lexicon and so an axiom of the proof) until we have reached all the axioms, which are justified by their respective lexical entries. The chart items marked in gray do not contribute to the proof of 11.

Implementation notes The actual implementation keeps track of several types of additional information: it computes the semantics of the derivation and there is also a mechanism for computing the (log-)probabilities of the rules.

The implementation also uses an important simplification: once we have computed a chart item for a formula A over span I and J then we will treat this as known and reject any further derivations of this formula A over the same string (if probabilities are used, only the most probably derivation of A is kept). This can throw away alternative semantic readings for a phrase, but reduces the size of the chart. If desired, this behavior can easily by changed by replacing the don't care variables `_` in the predicate `subsumes_data` by a test for α-equivalence of the lambda-terms.

2.2 Hypothetical Reasoning

Hypothetical reasoning is implemented using a strategy very similar to "gap threading" in the parsing literature. Chart items are now of the form $\langle \Gamma, F, L, R, e \rangle$, where e is a set of pairs of the form $P - A$, with P a position integer and A a formula; the set e is the set of "extracted" constituents which have been used to compute F. The rules for extraction (hypothetical reasoning) are shown in Figure 2. The $/E$ and $\backslash E$ rules of Figure 1 have been updated to include the new set e of extracted items.

The set union $e_1 \cup e_2$ of two such sets e_1 and e_2 is defined only if $e_1 \cap e_2$ is empty; this reflects that fact that a hypothesis to be discharged later can only be used once.

The *e_start* rule states that if we have a formula $X/(Y/\Diamond_1\Box_1 B)$ with rightmost position K and a formula A/B spanning positions $I - J$ then we can conclude there is a formula A spanning positions $I - J$ depending on an extracted element $K - B$. The underscores _ indicate we do note care about this value for the chart item. So for the leftmost premiss of the *e_start*, we do not

$$\frac{\langle \Gamma, A/B, I, J, e_1\rangle \quad \langle \Delta, B, J, K, e_2\rangle}{\langle \Gamma \circ \Delta, A, I, K, e_1 \cup e_2\rangle}\ /E$$

$$\frac{\langle \Gamma, B, I, J, e_1\rangle \quad \langle \Delta, B\backslash A, J, K, e_2\rangle}{\langle \Gamma \circ \Delta, A, I, K, e_1 \cup e_2\rangle}\ \backslash E$$

$$\frac{\langle \_, X/(Y/\Diamond_1\Box_1 B), \_, K, \_\rangle \quad \langle \Gamma, A/B, I, J, e\rangle}{\langle \Gamma, A, I, J, e \cup \{K - B\}\rangle}\ e\_start$$

$$\frac{\langle \Gamma, X/(Y/\Diamond_1\Box_1 B), I, J, e_1\rangle \quad \langle \Delta, Y, J, K, e_2 \cup \{J - B\}\rangle}{\langle \Gamma \circ \Delta, X, I, K, e_1 \cup e_2\rangle}\ e\_end$$

Figure 2: Hypothetical reasoning chart rules

care about the antecedent, about the leftmost position or about the stack of extractions: the formula $X/(Y/\Diamond_1\Box_1 B)$ functions as a sort of "trigger" allowing extraction of a B formula to take place to its right.

The rule *e_start* has as side conditions that $K \leq I$ and that $K - B$ is not a member of e (this is a general consequence of the disjoint set union used).

The *e_start* rule is a combination of using a $B \vdash B$ axiom in combination with a previous proof of $\Gamma \vdash A/B$ to derive by $/E$ that $\Gamma, B \vdash A$, with the condition that the $B \vdash B$ hypothesis must be discharged at position K by the formula $X/(Y/\Diamond_1\Box_1 B)$ which licensed this rule. This discharge is taken care of by the *e_end* rule.

The *e_end* rule states that if we have derived a Y using a hypothetical B to the immediate right of a formula $X/(Y/\Diamond_1\Box_1 B)$, then we can derive an X spanning the total positions, removing the formula $J - B$ from the set of extracted elements; the notation $e_2 \cup \{J - B\}$ indicates that the Y formula was derived using the formula B exactly once (plus some additional, possibly empty, set of items e_2).

A chart item is *coherent*, if for all $P - A \in e$, $L \leq P$. This is because formulas of the form $X/(Y/\Diamond_1\Box_1 B)$ are looking to their right for a constituent Y missing a B somewhere.

We initialize all lexical entries with the empty set and at the end of a derivation, we require that the set of traces is empty. That is, our lexical entries are now of the form $\langle w_i, A_i, i-1, i, \emptyset\rangle$ and our goal is of the form $\langle \Gamma, C, 0, N, \emptyset\rangle$ for some formula C and with the antecedent term Γ such that $\mathit{yield}(\Gamma) = w_1, \ldots, w_n$.

Typical instantiations of the formula $X/(Y/\Diamond_1\Box_1 B)$ are $(n\backslash n)/(s/\Diamond_1\Box_1 np)$ (for relativizers) and $(np\backslash s)/((np\backslash s)/\Diamond_1\Box_1 np)$ (for clitics).

Example The chart rules for extraction/hypothetical reasoning are perhaps the easiest to understand by seeing them in action. We can derive the sentence fragment "qu'on emprunte" (*that we borrow*) to be of type $n\backslash n$ as follows.

	Chart Item	Justification
1	$\langle qu', (n\backslash n)/(s/\Diamond_1\Box_1 np), 2, 3, \emptyset\rangle$	Lex
2	$\langle on, np, 3, 4, \emptyset\rangle$	Lex
3	$\langle emprunte, (np\backslash s)/np, 4, 5, \emptyset\rangle$	Lex
4	$\langle emprunte, np\backslash s, 4, 5, \{3 - np\}\rangle$	1,3 *e_start*
5	$\langle on \circ emprunte, s, 3, 5, \emptyset \cup \{3 - np\}\rangle$	2,4 $\backslash E$
6	$\langle qu' \circ (on \circ emprunte), n\backslash n, 2, 5, \emptyset\rangle$	1,5 *e_end*

Incompleteness of the rules As can be seen from the rules, they are incomplete. The *extraction start* rule can apply only to formulas of the form $X/(Y/\Diamond_1\Box_1 B)$, with a fixed combination of implications (excluding, for example $(Y/\Diamond_1\Box_1 B)\backslash X$ or $X/(Y \bullet \Diamond_1\Box_1 B)$ and only when the B formula is an argument, since the *extraction start* rule is essentially the $/E$ rule applied to a B hypothesis "at a distance". Another restriction is that each combination of rightmost position and extracted formula $R - B$ can introduce only one hypothetical item. We would need additional chart rules if we want to treat these other cases. The treatment of gapping, discussed briefly in Section 2.4, allows the extracted element to be the functor of an elimination rule.

Though this formula restriction and the resulting incompleteness are unfortunate, since it requires us to be careful in case the algorithm doesn't find a proof, this rule captures most of the occurrences of the $\Diamond_1\Box_1$ mixed associativity/commutativity rather nicely.

Implementation notes The actual implementation also keeps track of the rightmost position J used for the *e_start* rule. So the set of items e takes the form triples $K - J - B$ where K is the rightmost position of the licensor formula and J is the rightmost position of the extracted B formula. This allows us to use a single rule schema for a combination of mixed associativity and mixed commutativity — the rules for $\Diamond_1\Box_1$ shown — and for $\Diamond_0\Box_0$ which only allow mixed associativity (or "right-node raising"). The *e_end* rule in this case requires that the rightmost position K of the constituent Y is also the rightmost position of the extracted B formula. This right-node raising analysis also has a rule for formulas of the form $(Y/\Diamond_0\Box_0 B)\backslash X$ and can therefore treat lexical formulas such as $((((np\backslash s)/\Diamond_0\Box_0 np)\backslash(np\backslash s)/np))/((np\backslash s)/\Diamond_0\Box_0 np)$, which is a transitive verb conjunction type but which allows combinations such as the following.

$$(np\backslash s)/(np\backslash s), (np\backslash s)/np \vdash (np\backslash s)/\Diamond_0\Box_0 np$$

This is useful for patterns like "has read and might implement (Dijkstra's algorithm" , where both "has read" and "might implement" require the derivation pattern shown above.

2.3 Head wrap

French adverbs can occur at the start of the sentence, at the end of the sentence and before the verb (where we can assign them the formulas s/s, $s\backslash s$ and $(np\backslash s)/(np\backslash s)$ respectively.[2] In addition, French adverbs can occur directly after the verb but also between a verb and its arguments. In order to avoid unnecessary duplication in the lexicon, we assign adverbs the type $s\backslash_1 s$ (or, in some cases, $(np\backslash s)\backslash_1(np\backslash s)$) and use structural rules to move the verb to a sentence-final position.

In Figure 3 we see how this idea translates into chart rules. In addition to the set of extracted items, our chart items now contain a *stack* of head-wrapped elements. We have chosen a stack instead of a set here to avoid generating readings which would correspond to permutations of the adverbs. With few exceptions, adverbs take scope from left to right. In the chart rules, "+" corresponds to stack concatenation, $[H|T]$ indicates a stack with first element H and rest of the stack T (which is itself a valid stack) and [] is the empty stack. We both end and start our proof with empty stacks ($h = []$) and empty sets of traces ($e = \emptyset$). That is, our lexical entries are of the form $\langle w_i, A_i, i-1, i, \emptyset, []\rangle$ and the goal is $\langle \Gamma, B, 0, N, \emptyset, []\rangle$ with $\mathit{yield}(\Gamma) = w_1, \ldots, w_n$

The *wr* rule wraps a chart entry with formula $X\backslash_1 X$ to its correct *syntactic* position, but also pushes it onto the stack h_2. As can been seen from the rule, the stack h_1 is then prefixed to this new stack, thereby keeping all the stack elements in the desired order: the elements in h_1 before the new item and the elements in h_2 after it.

Finally, the *wpop* rule simply allows us to pop a stack element $X\backslash_1 X$ whenever the current chart item containing the stack is of type X.

Example The wrapping rules are best illustrated by example. The sentence "il occupera ensuite diverses fonctions" (*he will occupy various functions afterwards*) is analysed as follows.

	Chart Item	Just.
1	$\langle il, np, 0, 1, \emptyset, []\rangle$	Lex
2	$\langle occupera, (np\backslash s)/np, 1, 2, \emptyset, []\rangle$	Lex
3	$\langle ensuite, s\backslash_1 s, 2, 3, \emptyset, []\rangle$	Lex
4	$\langle diverses, np/n, 3, 4, \emptyset, []\rangle$	Lex
5	$\langle fonctions, n, 4, 5, \emptyset, []\rangle$	Lex
6	$\langle occupera \circ_1 ensuite, (np\backslash s)/np, 1, 3, \emptyset, [2{,}3 - s\backslash_1 s]\rangle$	2,3 *wr*
7	$\langle diverses \circ fonctions, np, 3, 5, \emptyset, []\rangle$	4,5 $/E$
8	$\langle (occupera \circ_1 ensuite) \circ (diverses \circ fonctions), np\backslash s, 1, 5, \emptyset, [2{,}3 - s\backslash_1 s]\rangle$	6,7 $/E$
9	$\langle il \circ ((occupera \circ_1 ensuite) \circ (diverses \circ fonctions)), s, 0, 5, \emptyset, [2{,}3 - s\backslash_1 s]\rangle$	1,8 $\backslash E$
10	$\langle il \circ ((occupera \circ_1 ensuite) \circ (diverses \circ fonctions)), s, 0, 5, \emptyset, []\rangle$	9 *wpop*

[2] We have chosen an event semantics in the style of Davidson for adverbs, which means that we can treat many adverbs as sentence modifiers. Some subject-oriented adverbs, such as "ensemble" (*together*) need both the subject *np* and the sentence for their semantics and are assigned $(s/(np\backslash s))/np$ and $(np\backslash s)\backslash(np\backslash s)$ instead.

$$\frac{\langle \Gamma, A/B, I, J, e_1, h_1\rangle \quad \langle \Delta, B, J, K, e_2, h_2\rangle}{\langle \Gamma \circ \Delta, A, I, K, e_1 \cup e_2, h_1 + h_2\rangle}\ /E$$

$$\frac{\langle \Gamma, B, I, J, e_1, h_1\rangle \quad \langle \Delta, B\backslash A, J, K, e_2, h_2\rangle}{\langle \Gamma \circ \Delta, A, I, K, e_1 \cup e_2, h_1 + h_2\rangle}\ \backslash E$$

$$\frac{\langle \_, X/(Y/\Diamond_1\Box_1 B), \_, K, \_, \_\rangle \quad \langle \Gamma, A/B, I, J, e, h\rangle}{\langle \Gamma, A, I, J, e \cup \{K - B\}, h\rangle}\ e\_start$$

$$\frac{\langle \Gamma, X/(Y/\Diamond_1\Box_1 B), I, J, e_1, h\rangle \quad \langle \Delta, Y, J, K, e_2 \cup \{J - B\}, []\rangle}{\langle \Gamma \circ \Delta, X, I, K, e_1 \cup e_2, h\rangle}\ e\_end$$

$$\frac{\langle \Gamma, X, I, J, e_1, h_1\rangle \quad \langle \Delta, Y\backslash_1 Y, J, K, e_2, h_2\rangle}{\langle \Gamma \circ_1 \Delta, X, I, K, e_1 \cup e_2, h_1 + [\text{J-K} - Y\backslash_1 Y | h_2]\rangle}\ wr$$

$$\frac{\langle \Gamma, X, I, J, e, [\text{K-L} - X\backslash_1 X | h]\rangle}{\langle \Gamma, X, I, J, e, h\rangle}\ wpop$$

Figure 3: Head wrap chart rules

The parse first combines the transitive verb "occupera" (*will occupy*, chart item 2) with the adverb "ensuite" (*afterwards*, chart item 3) by pushing the adverb on the stack and by combining the lexical strings, producing chart item 6. We continue the proof with elimination rules until we derive s from positions 0 to 5 but with the adverb still on the stack. Since s and $s\backslash_1 s$ match the formulas of a *wpop* rule, we pop the adverb from the stack and produce the final item 10.

The example below shows the interaction of the head wrap and the extraction rules.

	Chart Item	Just.
1	$\langle qu', (n\backslash n)/(s/\Diamond_1\Box_1 np), 0, 1, \emptyset, []\rangle$	Lex
2	$\langle il, np, 1, 2, \emptyset, []\rangle$	Lex
3	$\langle occupera, (np\backslash s)/np, 2, 3, \emptyset, []\rangle$	Lex
4	$\langle ensuite, s\backslash_1 s, 3, 4, \emptyset, []\rangle$	Lex
5	$\langle occupera, np\backslash s, 2, 3, \{1 - np\}, []\rangle$	1,3 *e_start*
6	$\langle occupera \circ_1 ensuite, (np\backslash s)/np, 2, 4, \emptyset, [2{,}3 - s\backslash_1 s]\rangle$	3,4 *wr*
7	$\langle il \circ occupera, s, 1, 3, \{1 - np\}, []\rangle$	2,5 $\backslash E$
8	$\langle occupera \circ_1 ensuite, np\backslash s, 2, 4, \{1 - np\}, [2{,}3 - s\backslash_1 s]\rangle$	4,5 *wr*
9	$\langle (il \circ occupera) \circ_1 ensuite, s, 1, 4, \{1 - np\}, [2{,}3 - s\backslash_1 s]\rangle$	4,7 *wr*
10	$\langle (il \circ occupera) \circ_1 ensuite, s, 1, 4, \{1 - np\}, []\rangle$	9 *wpop*
11	$\langle qu' \circ ((il \circ occupera) \circ_1 ensuite), n\backslash n, 0, 4, \emptyset, []\rangle$	1,10 *e_end*

Using chart items 2 and 8 above, we could have applied the $\backslash E$ rule to produce $il \circ (occupera \circ_1 ensuite)$, resulting in a chart item which would otherwise

be identical to item 9. Therefore, according the the implementation note discussed at the end of Section 2.1, this entry is treated as "already known" and not entered in the chart. Other chart items have multiple equivalent derivations (including even the antecedent term): for example, as shown in the table above, chart item 8 has been derived from 4 and 5 using *wr* but it has an alternative derivation from 1 and 6 using *e_start*: there are two equivalent ways to apply *e_start* and *wr* to the transitive verb to produce chart item 8.

Since the *e_end* rule requires an empty stack to apply, we cannot apply the *e_end* rule to chart item 9 and need to pop the stack first using *wpop*, producing chart item 10, which is the proper configuration for an application of *e_end*.

Implementation details The implementation allows us to pop $s\backslash_1 s$ elements from the stack at the $np\backslash s$ level as well. This allows infinitive arguments to take adverbs of the form $s\backslash_1 s$.

2.4 Other chart rules

Quoted speech In newspaper articles, quotes speech is rather frequent. Most frequently, this takes the form of a tag like "said the Prime Minister", and this does not necessarily occur at the end of a sentence. To complicate matters, we even have sentences like the following.

(1) [sl Les conservateurs], a ajouté le premier ministre ..., [sr "ne sont
[sl The Conservatives], has added the Prime Minister ..., [sr " are
pas des opportunistes qui virevoltent d'une politique à l'autre]
not opportunists who flip-flop from one policy to another]

In this sentence the quoted sentence is split into two parts (marked *sl* and *sr*) and there two parts together are the arguments of the past participle "ajouté" (*added*), which itself is the argument of the auxiliary verb form "a" (*has*) (and the elided material "..." includes an adverb modifying the past participle).

As a solution, the additional chart rules treat these combinations much like complex adverbs. For example, we can derive "a ajouté to be for type $s\backslash_1 s$ as follows.

$$\dfrac{\dfrac{\dfrac{\dfrac{a}{(s/np)/(np\backslash s_{ppart})}\,Lex \quad \dfrac{\dfrac{}{x \vdash s}\,Hyp \quad \dfrac{ajouté}{s\backslash_1(np\backslash s_{ppart})}\,Lex}{x \circ_1 ajouté \vdash np\backslash s_{ppart}}\,\backslash E}{a \circ (x \circ_1 ajouté) \vdash s/np}\,/E \quad \dfrac{}{np \vdash np}}{(a \circ (x \circ_1 ajouté)) \circ np \vdash s}\,/E \\ \dfrac{(x \circ_1 (a \circ ajouté)) \circ np \vdash s}{x \circ_1 ((a \circ ajouté) \circ np) \vdash s}\,MC_1 \\ \dfrac{x \circ_1 ((a \circ ajouté) \circ np) \vdash s}{(a \circ ajouté) \circ np \vdash s\backslash_1 s}\,MA_1 \\ \backslash I$$

Gapping Gapping includes cases like those shown below.

(2) Le véhicule pourrait être immobilisé et la carte grise
The car could be immobilised and the registration certificate
retenue.
retained

This sentence can be paraphrased along the lines "the car could be immobilised and the registration certificate *could be* retained", with the verb group "pourrait être" (*could be*) occurring only in the first sentence syntactically, but semantically it fills the same role in both sentences. This type of sentences is treated along the lines of (Hendriks 1995), though recast in the framework of (Moortgat 1996). The central idea of this analysis is that the verb group is extracted from both sentences and then infixed (at the place of the original verb group) in the first sentence.

Product rules Some conjunctions have the simplest analysis when we use the product formula. Look, for example, at the following sentence.

(3) augmenter [np ses fonds propres] [pp de 90 millions de francs] et
increase [np its equity] [pp by 90 million francs] and
[np les quasi-fonds propres] [pp de 30 millions]
[np its quasi-equity] [pp by 30 million]

Here the verb "augmenter" (*to augment*) takes both an *np* and a *pp* argument.

We can derive these cases by assigning "et" the following formula.

$$((np \bullet pp)\backslash(np \bullet \diamond_0\square_0 pp))/(np \bullet pp)$$

The $\bullet I$ rule is easy to add to the chart parser. The implementation is careful to use to product introduction rule only when an adjacent chart item requires a product argument (a naive implementation would concluded $A \bullet B$ from *any* adjacent chart items A and B).

The elimination rules are more delicate and involve patterns such as the following (these are easy to show valid using associativity of $\diamond_0\square_0 C$).

$$\frac{A/B \quad B \bullet \diamond_0\square_0 C}{A \bullet \diamond_0\square_0 C}\ prod\_c \qquad \frac{(A/C) \bullet \diamond_0\square_0 C}{A}\ prod\_e$$

Together, these allow us to combine $((np\backslash s)/pp)/np$ with $np \bullet \diamond_0\square_0 pp$ as follows.

$$\frac{\dfrac{((np\backslash s)/pp)/np \quad np \bullet \diamond_0\square_0 pp}{((np\backslash s)/pp) \bullet \diamond_0\square_0 pp}\ prod\_c}{np\backslash s}\ prod\_e$$

Left-node raising Very rarely, for a total of nine times in the entire corpus, we need left-node raising, the symmetric operation of right-node raising. In the example below, we have a conjunction of two combinations of two noun post-modifiers $n\backslash n$: "français Aérospatiale" and "italien Alenia".

(4) ... des groupes français Aérospatiale et italien Alenia ...
... of the groups french Aérospatiale and italian Alenia ...
... of the french group Aérospatiale and italian (group) Alenia ...

By analysing "et" (and) as $((\Diamond_0\Box_0 n\backslash n)\backslash(n\backslash n))/(\Diamond_0\Box_0 n\backslash n)$ we can use the derivability of $n\backslash n, n\backslash n \vdash \Diamond_0\Box_0 n\backslash n$ as follows.

$$\dfrac{\dfrac{}{x \vdash \Diamond_0\Box_0 n}\ Hyp \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{y \vdash \Box_0 n}\ Hyp}{\langle y\rangle^0 \vdash n}\ \Box E \qquad \dfrac{italien}{n\backslash n}\ Lex}{\langle y\rangle^0 \circ italien \vdash n}\ \backslash E \qquad \dfrac{Alenia}{n\backslash n}\ L}{(\langle y\rangle^0 \circ italien) \circ Alenia \vdash n}\ \backslash E}{\langle y\rangle^0 \circ (italien \circ Alenia) \vdash n}\ MA_l\Diamond_0}{x \circ (italien \circ Alenia) \vdash n}\ \Diamond E_2$$

$$\dfrac{x \circ (italien \circ Alenia) \vdash n}{italien \circ Alenia \vdash \Diamond_0\Box_0 n\backslash n}\ \backslash I_1$$

Final implementation notes Since the final chart parser has many inference rules which apply only in specific situations (essentially all rules, except for the basic AB rules) and since the chart parser has a fair amount of overhead trying (and failing) to match each of these rules, there is a separate mechanism which verifies if the formulas contain any patterns which trigger rules beyond the AB rules and if so, activate all potentially useful rules. Therefore, the product rules are only active if there is a formula of the form $A \bullet B$, the wrapping rules only if there is a formula $A\backslash_1 A$, etc.

3 Conclusion

We have given a fairly high-level description of the multimodal chart parser which is part of the type-logical treebank for French. The source code, issued under the GNU Lesser General Public License, contains much more detail.

References

Hendriks, P. (1995), Ellipsis and multimodal categorial type logic, *in* G. Morrill & R. T. Oehrle, eds, 'Proceedings of Formal Grammar 1995', Barcelona, Spain, pp. 107–122.

Jurafsky, D. & Martin, J. H. (2009), *Speech and Language Processing*, 2 edn, Pearson.

Moortgat, M. (1996), In situ binding: A modal analysis, *in* P. Dekker & M. Stokhof, eds, 'Proceedings 10th Amsterdam Colloquium', ILLC, Amsterdam, pp. 539–549.

Moortgat, M. (2010), 'Typelogical grammar', Stanford Encyclopedia of Philosophy Website. `http://plato.stanford.edu/entries/typelogical-grammar/`.

Moortgat, M. (2011), Categorial type logics, *in* J. van Benthem & A. ter Meulen, eds, 'Handbook of Logic and Language', North-Holland Elsevier, Amsterdam, chapter 2, pp. 95–179.

Moot, R. (2010), Semi-automated extraction of a wide-coverage type-logical grammar for French, *in* 'Proceedings of Traitement Automatique des Langues Naturelles (TALN)', Montreal.

Moot, R. (2012), 'Wide-coverage semantics for spatio-temporal reasoning', *Traitement Automatique des Languages* **53**(2), 115–142.

Moot, R. (2015*a*), 'TLGbank: A type-logical treebank for French', `http://richardmoot.github.io/TLGbank/`.

Moot, R. (2015*b*), 'A type-logical treebank for french', *Journal of Language Modelling* **3**(1), 229–264.

Moot, R. (2017), The Grail theorem prover: Type theory for syntax and semantics, *in* Z. Luo & S. Chatzikyriakidis, eds, 'Modern Perspectives in Type Theoretical Semantics', Studies in Linguistics and Philosophy, Springer, pp. 247–277.

Moot, R. (2018), 'Grail light', `https://github.com/RichardMoot/GrailLight`. Chart-based parser for type-logical grammars.

Moot, R. & Retoré, C. (2012), *The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics*, number 6850 *in* 'Lecture Notes in Artificial Intelligence', Springer.

Pereira, F. & Shieber, S. (1987), *Prolog and Natural Language Analysis*, CSLI, Stanford.

Shieber, S., Schabes, Y. & Pereira, F. (1995), 'Principles and implementation of deductive parsing', *Journal of Logic Programming* **24**(1–2), 3–36.