

INSTITUT NATIONAL DES SCIENCES APPLIQUEES

\*\*\*\*\*

ÉCOLE DOCTORALE SYSTÈMES

TOULOUSE



Rapport de D.E.A.  
Systèmes Informatiques

*Calcul des relations  
temporelles du discours*

*Institut de Recherche en Informatique de Toulouse*

*sous la direction de Philippe Muller*

juillet 2003

*Tannier Xavier*



# REMERCIEMENTS

Merci tout d'abord à Philippe Muller de m'avoir accueilli, donné de la lecture, soutenu et aussi laissé tranquille. Merci à Laurent de m'avoir laissé profiter du ventilateur, et à Sylvie de nous avoir rappelé tous les jours à 11h45 qu'il n'y a pas que la science dans la vie. Merci à tous ceux qui ont contribué à me faire comprendre comment ce monde fonctionne.

Merci à Mimosa de m'avoir hébergé tout au long de ce stage, et à Capella, Fusion et Cosmo de m'avoir régulièrement reçu.

Merci aussi aux professeurs qui, en deux mots ou en longues discussions, m'ont dit que c'était une idée étrange, mais qu'il fallait que je le fasse.



# SOMMAIRE

<b>Introduction</b>	<b>5</b>
<b>La linguistique informatique</b>	<b>9</b>
<b>1 Quelques aspects de la linguistique informatique</b>	<b>11</b>
1.1 La linguistique, la linguistique informatique . . . . .	11
1.2 Les différents niveaux d'analyse du langage . . . . .	12
1.3 Syntaxe, grammaire formelle et grammaire hors-contexte . . . . .	13
1.4 De la syntaxe à la sémantique . . . . .	14
1.5 La pragmatique . . . . .	16
<b>2 Le problème du temps</b>	<b>19</b>
2.1 Les structures temporelles . . . . .	19
2.1.1 La structure de points . . . . .	19
2.1.2 La structure d'intervalles . . . . .	20
2.1.3 La structure d'événements . . . . .	21
2.1.4 La théorie d'Allen . . . . .	22
2.2 Le temps dans la langue . . . . .	24
2.2.1 Les temps grammaticaux . . . . .	24
2.2.2 Les connecteurs et adverbiaux temporels . . . . .	26
<b>Calcul des relations temporelles</b>	<b>29</b>
<b>3 Les fondations : l'analyse syntaxique</b>	<b>33</b>
3.1 Analyse morpho-syntaxique . . . . .	34
3.2 Analyse syntaxique . . . . .	35
3.3 Problèmes soulevés . . . . .	36
3.4 La gestion des adverbiaux temporels . . . . .	41
3.5 Conclusion . . . . .	45

<b>4</b>	<b>Les adverbiaux temporels</b>	<b>47</b>
4.1	Reconnaissance de l'événement . . . . .	47
4.2	Déduction des informations temporelles . . . . .	50
4.3	Conclusion . . . . .	51
<b>5</b>	<b>Traitement des temps grammaticaux</b>	<b>53</b>
5.1	Relations entre les événements verbaux . . . . .	53
5.2	Incorporation à l'étage précédent . . . . .	54
5.3	Conclusion . . . . .	55
<b>6</b>	<b>Validation des premiers résultats</b>	<b>57</b>
6.1	Annotation humaine . . . . .	57
6.2	Méthode de comparaison des résultats . . . . .	59
6.3	"L'annotation est humaine" . . . . .	61
6.3.1	Distinction des événements . . . . .	61
6.3.2	Niveau "d'inférence" . . . . .	62
6.3.3	Annotateurs spécialistes et annotateurs profanes . . . . .	62
6.4	Les résultats . . . . .	62
6.4.1	Traitement des marqueurs temporels . . . . .	63
6.4.2	Application des règles entre les temps grammaticaux . . . . .	63
6.4.3	Commentaires . . . . .	63
6.5	Conclusion . . . . .	64
	<b>Conclusion</b>	<b>65</b>
	<b>Glossaire</b>	<b>68</b>
	<b>Bibliographie</b>	<b>70</b>
<b>A</b>	<b>Quelques outils utilisés</b>	<b>i</b>
<b>B</b>	<b>Analyse syntaxique : dictionnaire des données</b>	<b>v</b>
<b>C</b>	<b>Propagation de contraintes d'Allen</b>	<b>ix</b>
<b>D</b>	<b>Liste des relations induites par les temps grammaticaux</b>	<b>xi</b>
<b>E</b>	<b>Guide de l'annotateur</b>	<b>xv</b>
<b>F</b>	<b>Textes étudiés</b>	<b>xix</b>

# INTRODUCTION

Le traitement automatique du langage naturel (TALN) est une discipline qui se situe à la frontière de plusieurs domaines scientifiques, en particulier la linguistique et l'intelligence artificielle. Celle-ci, dernière arrivée dans le cartel des sciences s'intéressant à ce sujet, a profité des avancées des autres disciplines au fil des décennies précédentes. Elle a également permis de définir de nombreux nouveaux enjeux à l'étude du langage et des langues. Parmi ces enjeux, on peut citer la traduction automatique, le dialogue homme-machine ou la recherche d'informations dans des masses de données textuelles dont le volume est aujourd'hui hors de portée de l'humain.

Dans cette perspective générale, nous nous sommes pendant ce stage focalisé sur un type d'information spécifique et crucial, à savoir les informations temporelles. En effet, connaître l'agencement temporel des événements décrits par un document est de première importance pour l'analyser de façon efficace, en particulier lorsqu'il s'agit de textes narratifs tels que les articles de journaux ou la littérature. Les travaux à ce sujet, s'ils mobilisent de plus en plus de chercheurs, n'en sont qu'à leurs débuts ([Harper *et al.*, 2001]) et se heurtent encore à des problèmes de formalisme : l'annotation temporelle des textes, c'est-à-dire le mode de recueil des informations nécessaires pour opérer les déductions temporelles, ne fait pas l'objet d'un consensus, même quand il s'agit d'annotateurs humains [Setzer, 2001; Katz and Arosio, 2001; Schilder and Habel, 2001].

À la lumière des travaux existant, et après une étude préalable de nombreux textes à forte composante narrative, nous avons fait des choix d'ontologies\* temporelles et de méthodes de travail qui nous semblaient appropriées. Nous avons ensuite déterminé et organisé les divers éléments qui permettent de déterminer les relations temporelles dans des textes en Français (dates, marqueurs et connecteurs temporels, temps des verbes, etc...). Nous avons défini une hiérarchie entre ces paramètres et bâti une esquisse de théorie du calcul des relations temporelles dans le discours. Nous avons par la suite implémenté une partie du produit de ces réflexions.

Ce type de travail étant relativement nouveau dans le domaine, nous avons de plus proposé une méthode d'évaluation et de validation des résultats obtenus. Cette méthode est basée sur la comparaison de l'annotation temporelle automatique d'un certain nombre de textes avec des annotations effectuées par des humains. Elle pose de nombreux problèmes, qui ne sont pas tous résolus.

Le présent rapport a pour but d'exposer le contexte et le contenu du travail réalisé lors de ce stage. Dans un premier temps, nous exposerons rapidement les différents aspects de la linguistique et de la linguistique informatique, puis du problème particulier des processus temporels, dont la compréhension nous semble nécessaire à une bonne appréhension des avancées auxquelles nous avons participées. Puis nous entrerons dans les détails du travail effectué, des étapes préliminaires à la phase d'évaluation.

*N.B.* : Vous pouvez trouver une définition des termes suivis d'une astérisque\* dans le glossaire situé à la page 67.



# L'ÉQUIPE LILAC

Le stage que j'ai effectué s'est déroulé à l'Institut de Recherche en Informatique de Toulouse, et plus particulièrement avec Philippe Muller, au sein de l'équipe LiLac (Logic, Interaction, Language, and Computation).

Le thème de recherche central de l'équipe LiLac est la modélisation du raisonnement et la représentation des connaissances. Ce thème se situe à l'intersection de plusieurs disciplines : l'informatique, la logique, la linguistique et la philosophie. En particulier il est placé au cœur du domaine de l'Intelligence Artificielle, dans ses aspects à la fois fondamentaux et appliqués.

Ce groupe adopte une approche normative, ayant pour but l'identification d'un ensemble de propriétés caractérisant le raisonnement humain. Son programme se focalise sur les modèles de logique formelle.

Les axes principaux de recherche sont les suivants :

- Espace et Mouvement, et en particulier :
  - ◇ Méthodes axiomatiques et modélisation<sup>1</sup>, avec pour sources d'inspiration le langage naturel, le monde physique et l'ontologie.
  - ◇ Complexité et déduction<sup>2</sup>, développement d'algèbres d'intervalles en dimension  $n$ , axes rectilignes ou cercle : extension de la théorie d'Allen à des dimensions d'ordres supérieurs.
- Langage et Interaction<sup>3</sup>, avec :
  - ◇ Sémantique lexicale
  - ◇ Analyse du discours
  - ◇ Sémantique\* et pragmatique\* du dialogue
  - ◇ Théories de l'interaction rationnelle
  - ◇ Théorie des actes de langage
  - ◇ Méthodes de déductions
- Preuves et validation, et en particulier les algorithmes pour les logiques non-classiques (notamment logiques temporelles)<sup>4</sup>

---

<sup>1</sup>Laure Vieu, Philippe Muller, Claudio Masolo, Vincent Dugat, Yannick Larvor, Luis Fariñas del Cerro

<sup>2</sup>Luis Fariñas del Cerro, Philippe Balbiani

<sup>3</sup>Mario Borillo, Olivier Gasquet, Andreas Hertzog, Laure Vieu, Philippe Muller, Bruno Gaume, Laurent Perrussel

<sup>4</sup>Philippe Balbiani, Olivier Gasquet, Luis Fariñas del Cerro



Première partie

La linguistique informatique



# CHAPITRE 1

## Quelques aspects de la linguistique informatique

### 1.1 La linguistique, la linguistique informatique

On s'accorde généralement à reconnaître que le statut de la linguistique comme étude scientifique du langage a été assuré par Ferdinand de Saussure à partir de 1916 [de Saussure, 1916]. Ce linguiste suisse distingue les concepts de “langage” et de “langue”, la seconde étant la composante sociale du premier, imposant aux individus des conventions nécessaires à l'exercice de la parole, qui en est la manifestation individuelle et volontaire.

Depuis, de nombreuses disciplines se sont emparées du langage pour en faire un objet d'étude prépondérant dans la science de la fin du XX<sup>ième</sup> siècle. Lorsque les linguistes étudient la structure du langage lui-même, la façon qu'ont les mots de former des phrases correctes et formant un sens, les psycholinguistes, par exemple, étudient les processus de production et de compréhension du langage, la traduction par l'être humain de suites de mots en une structure de phrases et en une signification particulière. Les philosophes s'interrogent eux sur les notions de sens et de vérité, sur les relations entre les mots et les objets du monde réel.

Enfin, la linguistique informatique, science la plus récente de toutes celles citées, prend avantage de toutes les acquisitions des autres disciplines dans le but de développer des théories permettant l'identification des structures de phrases, la modélisation de la connaissance et du raisonnement, et enfin d'utiliser le langage naturel pour faire accomplir des tâches spécifiques à des machines.

Parmi les nombreuses applications envisagées par les chercheurs en linguistique computationnelle, on trouve :

- la recherche automatique de documents sur un certain sujet (dans une bibliothèque ou sur le Web) ;
- l'extraction d'informations dans des documents ;
- la traduction automatique de documents ;
- la génération automatique de résumé de textes ;
- la facilitation du dialogue homme-machine.

## 1.2 Les différents niveaux d'analyse du langage

L'analyse du langage nécessite une connaissance de sa structure sur de nombreux niveaux : que sont les mots ? Que signifient-ils ? Comment se combinent-ils pour former la phrase ? Comment contribuent-ils au sens de la phrase ? Et, par ailleurs, comment fonctionnent le monde et le raisonnement de l'humain dans le monde ? Par exemple, pour participer à une conversation, un être humain doit connaître non seulement le langage utilisé, mais également les règles du monde autour duquel lui et son interlocuteur vivent, ainsi que les règles élémentaires de la conversation.

Voici donc quelques formes de connaissance qui comptent parmi les domaines de recherche pour la reconnaissance et la génération du langage naturel :

- la phonétique et la phonologie, ou comment les mots et les phrases sont liés aux sons qui les réalisent à l'oral. Ne traitant que l'écrit, nous ne reviendrons pas sur ce domaine.
- la morphologie, ou comment les mots sont construits à partir d'entités plus petites (les *morphèmes*). Là encore, ce domaine n'entre pas dans le cadre de ce rapport.
- la syntaxe<sup>\*</sup>, ou comment les mots sont groupés pour former des phrases correctes, et quelles sont leurs fonctions grammaticales.
- la sémantique<sup>\*</sup>, ou comment les mots font du sens lorsqu'ils sont insérés dans une phrase (indépendamment du contexte).
- le pragmatique<sup>\*</sup>, ou comment les phrases peuvent être interprétées selon leur contexte d'énonciation (interlocuteurs, phrases précédentes, connaissance commune du monde, ...)

James Allen [Allen, 1994] donne les exemples suivant pour faire la distinction entre syntaxe, sémantique et pragmatique. Considérons que ces phrases étaient candidates pour figurer en tête du présent rapport, c'est-à-dire qu'elles sont énoncées en l'absence totale de contexte :

- (1.1) Le traitement automatique du langage naturel est une discipline qui se situe à la frontière de plusieurs domaines scientifiques.
- (1.2) Les grenouilles vertes ont des gros nez.
- (1.3) Les idées vertes ont des gros nez.
- (1.4) Des vertes ont les idées nez gros.

La première phrase semble être un début raisonnable pour un tel rapport. Elle correspond à tout ce qui est connu en matière de syntaxe, de sémantique et de pragmatique. La phrase 1.2 est bien formée sur les plans syntaxique et sémantique, mais pas pragmatique. En effet, elle conviendrait mal comme première phrase de ce rapport, et le lecteur ne verrait aucune raison valable de la voir utilisée.

Mais l'exemple 1.3 serait pire encore : il est à la fois pragmatiquement et sémantiquement mal formé. On remarque en effet qu'il est possible d'affirmer que la phrase 1.2 est vraie ou fausse, tandis que c'est impossible pour 1.3 dans une conversation cohérente. La structure en est pourtant correcte, mais des idées ne peuvent pas être vertes et, même si elles le peuvent dans certains contextes, elles n'ont certainement pas de larges nez.

Enfin, la phrase 1.4 est tout simplement inintelligible : elle contient pourtant les mêmes mots que la précédente, mais ne respecte aucune des structures grammaticales admises en Français.

Dans le cadre de notre travail, nous nous basons uniquement sur la langue écrite. Cela implique que les entités les plus petites que nous allons étudier (les *unités grammaticales*) sont les mots, déjà clairement distingués dans un texte par des espaces (ce n'est pas aussi aisé pour la langue orale). Cela ne signifie pas que la structure interne du mot n'est pas prise en compte du tout (nous le verrons lors de l'analyse morpho-syntaxique\*). Mais nous n'utiliserons en pratique que les trois étapes que sont la syntaxe, la sémantique et la pragmatique, à des degrés divers. Les trois sections qui suivent tentent d'éclaircir le rôle de chacune dans l'analyse d'un texte. Pour rester clair et concis, nous nous focaliserons uniquement sur les points qui seront par la suite utiles à la compréhension de notre travail.

### 1.3 Syntaxe, grammaire formelle et grammaire hors-contexte

Dans le cadre d'une étude syntaxique d'un texte, il est nécessaire de bâtir une **grammaire\***, c'est-à-dire une spécification formelle des structures autorisées par une langue. La manière courante de représenter la structure d'une phrase est l'arbre syntaxique. Ainsi, la structure de la phrase 1.5 (syntaxiquement correcte, mais pragmatiquement surprenante, en raison de notre connaissance du monde) est représentée à la figure 1.2.

(1.5) Une souris a mangé Jean.

Voici comment la comprendre : La phrase (**S** pour *sentence*) est composé par un syntagme nominal\* (NP pour *noun phrase*) et un syntagme verbal\* (VP pour *verbal phrase*). Ce NP consiste en un article (**ART**) *une* et un nom commun (**N**) *souris*. Le VP, lui, est constitué d'un verbe (**V**) *a mangé* et d'un autre NP, lui-même composé d'un simple nom propre (**NAME**).

Pour construire ainsi la structure de la phrase, nous devons savoir quelles structures sont correctes dans la langue étudiée (le français en l'occurrence). Nous décrivons donc un ensemble de **règles**. Une règle dit qu'un certain symbole peut être créé dans l'arbre en tant que père d'une séquence d'autres symboles. Une petite grammaire, qui permettrait d'aboutir à la structure de la figure 1.2, est définie dans la figure 1.1. Les règles 2 et 3, par exemple, disent qu'un NP peut être constitué par un nom propre **NAME** ou par un article suivi d'un nom (**ART N**). Les quatre dernières règles définissent des mots possibles pour chaque catégorie. Ces grammaires, construites entièrement avec un symbole unique à gauche, sont appelées **grammaires hors-contexte**. La première colonne de la figure 1.1 énumère les symboles **non-terminaux**, qui peuvent être décomposés, la seconde, les symboles **terminaux**. Enfin, il est possible de faire une utilisation récursive des règles. Ainsi, on pourrait définir la règle suivante :

$$\text{NP} \rightarrow \text{NP CONJ NP}$$

En considérant que le mot *et* est une conjonction (**CONJ**), on pourrait ainsi traiter les groupes nominaux tels que "*une souris et Jean*".

1. S $\rightarrow$ NP VP	5. NAME $\rightarrow$ <i>Jean</i>
2. NP $\rightarrow$ NAME	6. ART $\rightarrow$ <i>une</i>
3. NP $\rightarrow$ ART N	7. N $\rightarrow$ <i>souris</i>
4. VP $\rightarrow$ V NP	8. V $\rightarrow$ <i>a mangé</i>

FIG. 1.1: Une grammaire simple

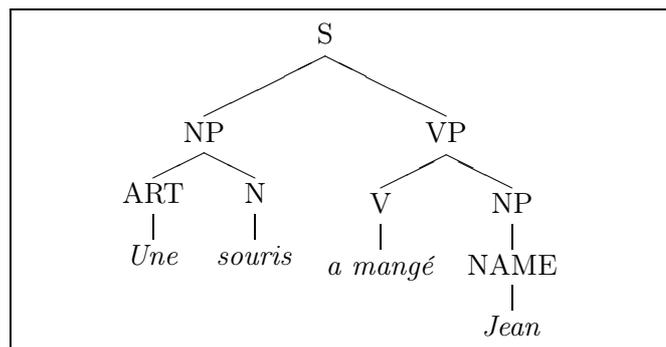


FIG. 1.2: Représentation de la phrase 1.5

Ces techniques de représentation grammaticale sont à l'origine de nombreux travaux et de nombreux formalismes plus complexes, tels que les grammaires d'unification. Ces grammaires prennent en compte de nombreux autres paramètres, comme par exemple le genre et le nombre des mots concernés. Pour plus de détails sur ces formalismes, vous pouvez consulter [Abeillé, 1993].

## 1.4 De la syntaxe à la sémantique

Nous présentons ici le principe général de la *sémantique formelle*, relativement répandue, mais qui n'est pas la seule méthode de traitement de l'information sémantique. Comme nous l'avons déjà expliqué, le niveau sémantique traite de la *signification* des phrases indépendamment du contexte. La représentation de cette signification est appelée la *forme logique*<sup>\*</sup>, et on y parvient par un processus d'*interprétation sémantique*. Les sens des mots sont des constantes classifiées selon ce qu'elles décrivent. Par exemple, les constantes décrivant des objets (concrets ou abstraits) du monde, comme les mots *Médor* ou *liberté*, sont appelés *termes*. Les constantes décrivant des relations ou des propriétés, comme *chien* ou *aimer*, sont des *prédicats*. On crée ainsi des propositions représentant la signification d'une phrase. Ainsi, dans les exemples suivants, chaque proposition b est la représentation logique de la phrase a.

- (1.6) a. Médor est un chien.  
b.  $chien(Médor)$
- (1.7) a. Marie aime Pierre.  
b.  $aime(Marie, Pierre)$
- (1.8) a. Une souris mange Pierre.  
b.  $\exists x (souris(x) \wedge mange(x, Pierre))$
- (1.9) a. Tous les chiens aboient.  
b.  $\forall x (chien(x) \rightarrow aboyer(x))$

Dans cette optique, la morpho-syntaxe<sup>\*</sup> et la syntaxe<sup>\*</sup> nous fournissent une structure d'aide à cette interprétation, complétée par l'introduction du lambda-calcul (voir par exemple [Berline, 2002] pour un détail sur le lambda-calcul). Ainsi, tous les prédicats unaires, c'est-à-dire qui s'appliquent à un seul terme, seront représentés de la même façon, comme par exemple :

$\lambda x. CHIEN(x)$   
 $\lambda x. SOURIS(x)$

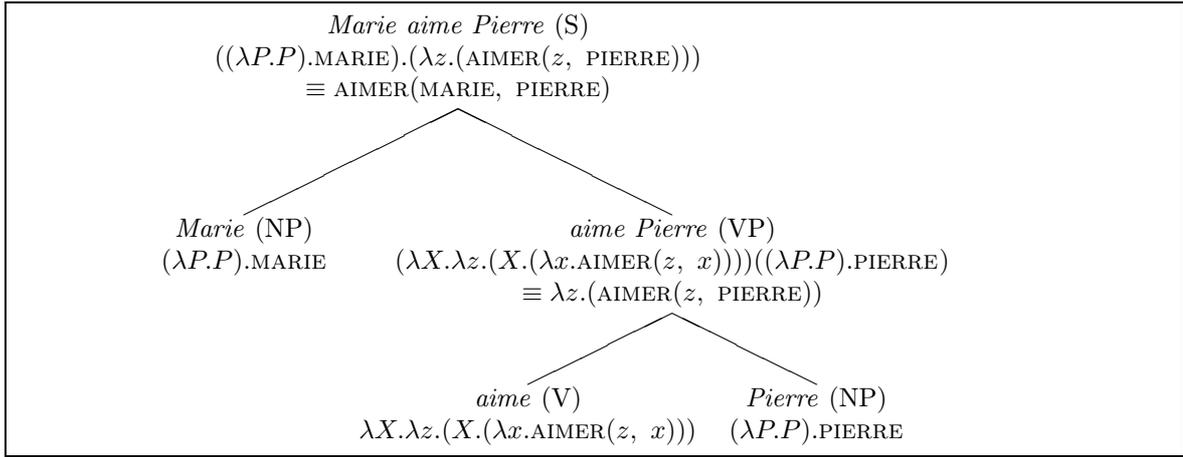


FIG. 1.3: Arbre sémantique de la phrase 1.7

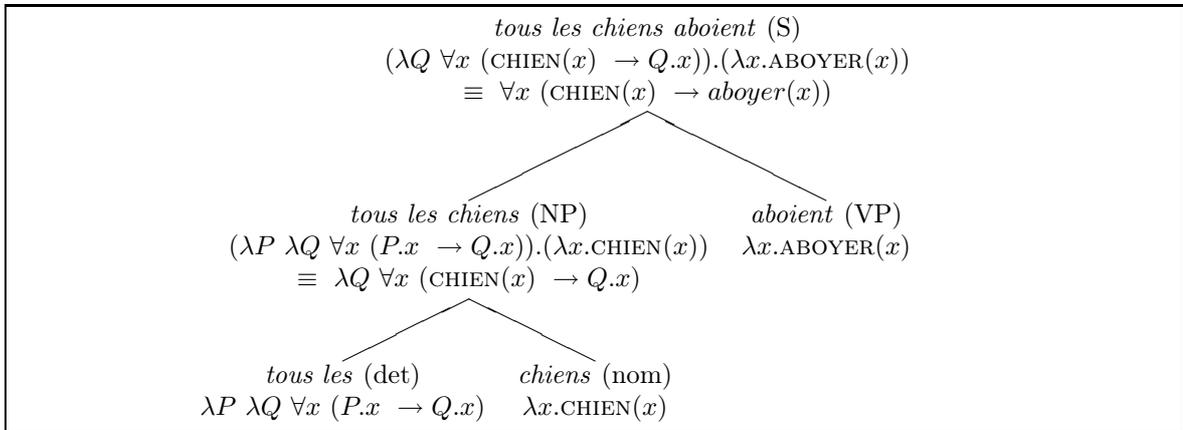


FIG. 1.4: Arbre sémantique de la phrase 1.9

On pourra alors appliquer une constante, notée ainsi :

$$(\lambda P.P).MEDOR$$

... et nous obtenons, par  $\beta$ -réduction :

$$(\lambda x.CHIEN(x)).((\lambda P.P).MEDOR) \equiv CHIEN(MEDOR)$$

Voici quelques exemples de transcription, qui nous permettront, avec nos règles syntaxiques, d'obtenir une forme logique de nos autres exemples :

- article indéterminé (*un*) :  $\lambda P \lambda Q \exists x (P.x \vee Q(x))$
- déterminant universel (*tous*) :  $\lambda P \lambda Q \forall x (P.x \rightarrow Q.x)$
- verbe intransitif\* (*aboyer*) :  $\lambda x.ABOYER(x)$
- verbe transitif direct\* (*aimer*) :  $\lambda X. \lambda z. (X.(\lambda x.AIMER(z, x)))$

Les figures 1.3 et 1.4 représentent l'application de nos grammaires sur ces définitions. Par exemple, la création d'un NP en assemblant un déterminant avec un nom se fera en appliquant la représentation du second sur celle du premier par  $\beta$ -réduction.

Dans le cadre de l'analyse sémantique d'un texte, de nombreux problèmes sont soulevés. Chacun d'eux est l'objet de nombreuses recherches, s'étendant souvent sur le plan pragmatique. Nous n'en citerons ici que deux, sans entrer dans les détails. Nous ne reviendrons pas

dessus, puisque nous n'avons pas pris en compte ces aspects lors de notre travail. Cependant, il est important de souligner que tous ces domaines sont inter-connectés, et qu'une recherche (beaucoup) plus approfondie sur le calcul des relations temporelles passera(it) par un traitement de ces problèmes.

**La résolution des pronoms.** Les humains utilisent des pronoms dans le langage écrit pour faire référence à des personnes, des choses ou des propositions qu'ils ont déjà introduites précédemment. Ceci est très difficile à repérer (à *résoudre*) automatiquement. Parfois, la référence est évidente, il s'agit du dernier élément introduit, comme dans la phrase 1.10. Les exemples suivant suffisent pour comprendre la difficulté de la résolution des pronoms. Ils sont de plus en plus complexes, jusqu'aux phrases 1.15 qui ne peuvent clairement être distinguées qu'avec l'aide d'une connaissance complexe du monde (ici sur les notions de Bien, de Mal, de loi et de biologie !). C'est ici le domaine de la pragmatique.

- (1.10) Jacques<sub>1</sub> était furieux. Il<sub>1</sub> s'était disputé avec Georges.
- (1.11) Jacques<sub>1</sub> était content. Son<sub>1</sub> ami Dominique était d'accord avec lui.
- (1.12) Dominique<sub>1</sub> rencontra Collin<sub>2</sub> à un congrès. Ils<sub>1+2</sub> se réconcilièrent.
- (1.13) Pierre<sub>1</sub> rencontra Sam<sub>2</sub> dans un bar. Il<sub>?</sub> lui<sub>?</sub> offrit un verre.
- (1.14) Marie<sub>1</sub> a construit un château de sable<sub>2</sub>. Cela<sub>2</sub> lui<sub>1</sub> a plu.
- (1.15) a. Pierre<sub>1</sub> empoisonna Sam<sub>2</sub>. Il<sub>2</sub> mourut.  
b. Pierre<sub>1</sub> empoisonna Sam<sub>2</sub>. Il<sub>1</sub> fut arrêté.

**La portée des quantificateurs.** Ce problème concerne l'ambiguïté parfois provoquée par l'introduction de quantificateurs (*tous, chaque, un*) dont la portée est incertaine, parfois même pour l'humain qui lit la phrase. Ainsi, dans les exemples suivants, un seul et unique chien suit-il tous les hommes, ou chaque homme est-il accompagné d'un chien différent ? Existe-t-il un seul piano soulevé par tous les hommes, ou soulèvent-t-ils chacun leur piano ? Y a-t-il un ou plusieurs problèmes d'environnement préoccupants ? Pierre a-t-il oublié tous ses amis, ou seulement certains d'entre eux ?

- (1.16) Un chien suit chaque homme qui passe la porte du bar.
- (1.17) Tous les hommes soulevèrent un piano.
- (1.18) Un problème d'environnement préoccupe tous les politiciens sérieux.

## 1.5 La pragmatique

Le terme de *pragmatique* regroupe un grand nombre de domaines de recherche, englobant en fait tous les problèmes qui ne peuvent pas être résolus (ou que l'on pense résoudre de façon plus efficace) avec la syntaxe et la sémantique. Pour donner très rapidement une idée des implications que cela comporte, énumérons quelques-uns des sujets étudiés :

**La déictique** concerne la façon avec laquelle le langage s'accommode des caractéristiques du contexte de l'énonciation : il peut s'agir de référence à des entités déjà introduites (*anaphores*, phrase 1.19), supposées connues par les interlocuteurs (comme le lieu ou l'heure courante, phrase 1.20) ou même par exemple montrées avec le doigt dans le langage oral (*déictique gestuelle*, phrase 1.21).

- (1.19) Pierre empoisonna Sam. *Il* fut arrêté.
- (1.20) J'ai rencontré la reine d'Angleterre *ici même*. C'était *l'année dernière*.

(1.21) Séparons-nous. Je pars avec *toi, toi et toi*, et *vous* partez de votre côté.

Les approches philosophiques s'intéressent également aux contextes dans lesquels une phrase est vraie ou fausse. Par exemple, les phrases 1.22a et 1.22b n'ont les mêmes *conditions de vérité* que si la personne qui dit la seconde est Leatizia de Ramolino ; de même, la phrase 1.23 n'a de sens que prononcée dans une période où la France est une monarchie.

(1.22) a. Leatizia de Ramolino est la mère de Napoléon.

b. Je suis la mère de Napoléon.

(1.23) Le Roi de France se repose avec sa favorite.

**Les implicatures conversationnelles** concernent ce que l'on peut déduire d'une phrase énoncée en dehors de sa signification littérale. Dans les dialogues qui suivent, la phrase prononcée par B comporte des implications de ce type, dont la signification est résumée après.

(1.24) A : Le voisin est-il chez lui ?

B : Sa voiture est devant le portail.

⇒ le voisin est probablement chez lui, car sa voiture est devant le portail.

(1.25) A : Je suis en panne d'essence.

B : Il y a un garage à deux pas d'ici.

⇒ Il y a un garage près d'ici, vous pourriez y aller pour alimenter votre voiture en essence.

(1.26) A : Georges et Donald vont-ils s'arrêter là ?

B : Sont-ils des hommes modérés ?

⇒ Georges et Donald ne vont bien sûr pas s'arrêter là !

(1.27) A : Ce professeur est un vrai sadique !

B : Hum, il fait frais, pour un mois de mai, tu ne trouves pas ?

⇒ Attention, sa femme est juste derrière toi !

**Les présuppositions** regroupent les informations que l'on peut déduire sur le contexte lors de l'énonciation de certaines phrases. Encore une fois, contentons-nous d'exemples (ce qui suit la flèche est la présupposition) :

(1.28) Le Roi de France est sage.

⇒ il existe un Roi de France.

(1.29) Jean regrette d'avoir fait ses études à Toulouse.

⇒ Jean est une personne identifiable par le locuteur et le destinataire du message. Jean a fait ses études à Toulouse.

(1.30) Jean ne bat plus sa femme.

⇒ Jean battait sa femme.

(1.31) Si le vice-chancelier invite Simone de Beauvoir, il regrettera d'avoir une féministe à sa table.<sup>1</sup>

⇒ Simone de Beauvoir est une féministe.

(1.32) Si le vice-chancelier invite le président des États-Unis, il regrettera d'avoir une féministe à sa table.<sup>1</sup>

<sup>1</sup>Exemple de Karttunen cité par [Levinson, 1983]

⇒ le vice-chancelier a également invité une féministe.

Pour avoir plus de détails sur ces problèmes et d'autres (actes de langage, structure de la conversation, etc. . . ), vous pouvez consulter [Levinson, 1983] et [Davis, 1991].

## CHAPITRE 2

# Le problème du temps

### 2.1 Les structures temporelles

Le choix de la représentation du temps est le premier problème à traiter, et il n'est pas du tout spécifique à l'étude du langage naturel. La linguistique a hérité cette réflexion des philosophes, mathématiciens et physiciens. Nous explorerons de façon très brève les choix possibles d'ontologies (de définition des entités et des relations de base), à savoir les structures de points, d'intervalles et d'événements, avant de nous arrêter sur la théorie de James Allen, que nous avons choisie pour la représentation des relations du discours.

#### 2.1.1 La structure de points

La conception du temps est couramment reliée à la notion de point ou d'instant sur un axe temporel. Les points permettent en effet d'utiliser les structures de nombres (entiers, rationnels ou réels). Cette conception est largement utilisée dans la modélisation de phénomènes évoluant dans le temps.

Une structure de points est définie par  $\langle T, < \rangle$ , avec  $T$  un ensemble non vide et  $<$  une relation binaire sur  $T$ . Les propriétés de transitivité et d'asymétrie de cette relation expriment le fait que le temps "s'écoule" :

$$\begin{aligned} &(\forall x \in T)(\forall y \in T)(\forall z \in T)(x < y \wedge y < z \Rightarrow x < z) \\ &(\forall x \in T)(\forall y \in T)(x < y \Rightarrow \neg y < x) \end{aligned}$$

De plus, il s'agit d'une structure linéaire :

$$(\forall x \in T)(\forall y \in T)(x \neq y \Rightarrow x < y \vee y < x)$$

Ces trois propriétés confèrent à notre structure de points le grade d'**ordre total**.

Cette structure temporelle doit être manipulée avec un *langage logique* ; la logique du temps, historiquement très liée au développement des logiques modales, a été introduite par Prior et est basée sur les connecteurs logiques habituels ( $\wedge$ ,  $\vee$ ,  $\neg$ ,  $\Rightarrow$ ,  $\Leftrightarrow$ ) et les opérateurs temporels P (passé) et F (futur). Ainsi, si l'action de chanter effectuée par Paul est notée p, on aura les représentations suivantes :

- *Paul chante* : p
- *Paul chanta* : Pp
- *Paul chantera* : Fp
- *Paul avait chanté* : Pp (on se place dans le passé d'un point situé au passé lui-même)

- *Paul aura chanté* : FPp
- ...

Ces formules seront enrichies avec des nouveaux opérateurs similaires à ceux utilisés en logique modale, et interprétées à l'aide des modèles de Kripke. Il ne semble pas utile ici de détailler ces mécanismes. Pour un résumé plus fourni, vous pouvez consulter [Bras, 1990].

Toutes les logiques dérivées de la logique du temps de Prior sont basées sur une ontologie de points. Nous allons maintenant nous intéresser à des ontologies d'intervalles.

### 2.1.2 La structure d'intervalles

#### Pourquoi un changement ?

Les arguments en faveur d'un changement d'ontologie relèvent tous de la recherche d'une conception du temps plus proche d'une conception de sens commun.

D'un point de vue philosophique, il semble que le concept de point dépourvu de durée ne corresponde pas à la réalité : la représentation de la continuité à partir de telles entités est contraire à notre intuition, ainsi le fait d'affirmer qu'une proposition est vraie en un point du temps de durée nulle.

Du point de vue linguistique, il est encore plus évident qu'une entité ponctuelle est mal adaptée pour l'expression de la référence temporelle. Même les expressions dites ponctuelles se réfèrent à des périodes étendues, comme dans les exemples suivants :

(2.1) À six heures précises, Harry quitta son bureau.

(2.2) À cet instant, l'immeuble s'effondra.

De plus, les phrases à la forme progressive (en général, à l'imparfait) ne peuvent être évaluées correctement que sur des intervalles. On pourrait objecter qu'un intervalle n'est qu'un ensemble de points, et définir la véracité d'une propriété sur un intervalle comme étant l'ensemble de tous les points sur lesquels cette propriété est vraie. Cet argument est contré par les exemples suivants, donné par Dowty :

(2.3) John took an hour to draw a circle. (Jean a mis une heure à tracer un cercle)

(2.4) John played the piano for an hour. (Jean a joué du piano pendant une heure)

La vérité de la phrase 2.3 dépend de la longueur de l'intervalle sur lequel Jean a tracé un cercle et de la localisation dans le passé de cet intervalle. Mais celui-ci ne peut être construit à partir de tous les points sur lesquels la proposition *Jean trace un cercle* est vraie ! De même, dans l'exemple 2.4, Jean a pu s'arrêter de jouer pendant quelque temps dans l'intervalle décrit sans que la phrase ne devienne fausse. Ainsi la vérité de la phrase sur l'intervalle est indépendante de la vérité de la phrase en chaque point de l'intervalle.

#### Les structures d'intervalles

Une structure d'intervalles est définie par  $\langle I, <, \subseteq \rangle$ , avec  $I$  un ensemble non vide d'entités temporelles et des relations de précédence ( $<$ ) et d'inclusion ( $\subseteq$ ). Voici quelques propriétés de cette structure :

- $<$  est asymétrique et transitive, c'est donc un **ordre partiel strict**.

- $\subseteq$  est un **ordre partiel**, elle est en effet :

- ◊ réflexive :  $(\forall x \in I)x \subseteq x$
- ◊ anti-symétrique :  $(\forall x \in I)(\forall y \in I)(x \subseteq y \wedge y \subseteq x \Rightarrow x = y)$
- ◊ transitive :  $(\forall x \in I)(\forall y \in I)(x \subseteq y \wedge y \subseteq z \Rightarrow x \subseteq z)$

On peut également remplacer la relation  $\subseteq$  par la relation notée  $O$  (overlap) qui exprime que deux événements ont une partie commune, et définie par rapport à l'inclusion :

$$x O y \Leftrightarrow (\exists z)(z \subseteq x \wedge z \subseteq y)$$

La mise en place des logiques temporelles basées sur les sémantiques d'intervalles amènent à des résultats relativement complexes, qu'il n'est pas nécessaire d'exposer ici. Reportez-vous à [Bras, 1990] pour un résumé de ces résultats, ainsi que pour un détail des techniques de construction des structures de points à partir de structures d'intervalles, et *vice versa*.

Des critiques ont été adressées aux sémantiques d'intervalles, notamment en ce qui concerne la difficulté de définir la vérité d'une proposition (vraie sur tous les sous-intervalles ? Sur au moins l'un d'eux ?). Ces problèmes ont provoqué la nécessité d'une entité plus globale et plus complète.

### 2.1.3 La structure d'événements

L'événement est une nouvelle entité primitive, de durée non nulle et finie, correspondant intuitivement à des fragments de notre perception du monde. Pour les linguistes comme pour les philosophes, les logiciens et les spécialistes de l'intelligence artificielle, la tendance est de préférer les événements aux intervalles car les événements ont une structure à portée non simplement temporelle, mais aussi spatiale et modale. Ce sont également des entités plus concrètes et plus faciles à appréhender que les propriétés abstraites du temps. Enfin, les problèmes de définition de la vérité sur un intervalle a vite posé des difficultés insurmontables.

Davidson a proposé de traiter les événements comme des objets, ajoutant à l'ensemble des individus d'un modèle, un ensemble d'événements. Par exemple, la phrase *Marie aime Paul* n'est plus représentée par *aimer(Paul, Marie)*, mais par :

$$\exists e \text{ aimer}(e, \text{Paul}, \text{Marie})$$

Si la plupart des événements sont décrits par des verbes, certains le sont par des noms (*accident, mariage...*), et la sémantique de séquences de phrases telles que celles de l'exemple 2.5 ne peut être exprimée que si des variables d'événements sont utilisées (un exemple de formulation des phrases i sous forme de prédicats est donnée par ii).

- (2.5)
- a. i. Tout couronnement nécessite une couronne.
  - ii.  $\forall e (\text{couronnement}(e) \Rightarrow \exists x \exists e' \text{nécessiter}(e') \wedge \text{couronne}(x) \wedge \text{objet}(x, e'))$
  - b. i. Arthur a été couronné.
  - ii.  $\exists e \exists x (\text{couronnement}(e) \wedge \text{couronne}(x) \wedge \text{agent}(\text{Arthur}, e) \wedge \text{objet}(x, e))$
  - c. i. Une couronne a été nécessaire.
  - ii.  $\exists x \exists e' (\text{couronne}(x) \wedge \text{nécessiter}(e') \wedge \text{objet}(x, e'))$

Notons que des représentations de 2.5a et de 2.5b on peut définir celle de 2.5c avec les règles logiques habituelles, il est donc indispensable de quantifier sur les événements.

Une structure d'événements est définie par Kamp par le triplet  $\langle E, \alpha, O \rangle$ , où  $E$  est un ensemble d'entités de base non nul,  $\alpha$  est la relation de précédence, et  $O$  la relation de recouvrement (si  $e_1 O e_2$ , une partie de  $e_1$  au moins a lieu en même temps que  $e_2$ ). Cette structure vérifie les conditions minimales suivantes (avec  $e_1, e_2, e_3, e_4$  des éléments de  $E$ ) :

- $\alpha$  est asymétrique :  $(e_1 \alpha e_2) \Rightarrow \neg (e_2 \alpha e_1)$
- $\alpha$  est transitive :  $(e_1 \alpha e_2) \wedge (e_2 \alpha e_3) \Rightarrow (e_1 \alpha e_3)$
- $O$  est symétrique :  $(e_1 O e_2) \Rightarrow (e_2 O e_1)$
- $O$  est réflexive :  $(e_1 O e_1)$
- principe de séparation :  $(e_1 \alpha e_2) \Rightarrow \neg (e_2 O e_1)$
- transitivité mixte :  $(e_1 \alpha e_2) \wedge (e_2 O e_3) \wedge (e_3 \alpha e_4) \Rightarrow (e_1 \alpha e_4)$
- principe de linéarité :  $(e_1 \alpha e_2) \vee (e_1 O e_2) \vee (e_2 \alpha e_1)$

Ces conditions minimales sont dictées par l’intuition que nous avons des événements et des relations qui les lient.

Nous avons donc présenté trois ontologies – structures de points, structures d’intervalles et structures d’événements. Il est fondamental de séparer le niveau temporel (points et intervalles) du niveau relatif à l’expérience du monde (événements). En effet, si les relations définies dans les structures d’événements sont des relations temporelles, les événements sont également des expériences, des “faits” qui ont lieu et qui déterminent la structure du temps. C’est pourquoi on peut dire que la logique d’Allen, que nous allons maintenant présenter, est à rapprocher, nonobstant son nom de “logique d’intervalles”, des structures d’événements, en ce sens que les intervalles d’Allen sont plutôt des entités de description de faits et d’événements, et non pas simplement des descriptions de simples “morceaux” de temps. Ainsi, il existe une relation d’identité entre deux intervalles – pourtant considérés comme différents – car deux événements distincts peuvent avoir lieu exactement en même temps.

### 2.1.4 La théorie d’Allen

Selon Allen, deux intervalles peuvent être liés entre eux par les 13 relations primitives suivantes, où  $X$  et  $Y$  sont des termes de type intervalle de temps (on appelle “relation inverse” la relation correspondante entre  $Y$  et  $X$ ) :

Relation	Symbole	Symbole relation-inverse
X before Y	<	>
X equals Y	=	=
X meets Y	m	mi
X overlaps Y	o	oi
X during Y	d	di
X starts Y	s	si
X finishes Y	f	fi

La figure 2.1 illustre de façon schématique la nature de ces relations. Remarquons que :

- les relations sont mutuellement exclusives : une seule relation est possible entre deux intervalles.
- il est possible de composer les relations, on définit la transitivité des relations entre intervalles, par exemple :

$$\forall i \forall j \forall k ((i < j) \wedge (j < k) \Rightarrow i < k)$$

$$\forall i \forall j \forall k ((i \text{ m } j) \wedge (j \text{ d } k) \Rightarrow (i \text{ o } k) \vee (i \text{ d } k) \vee (i \text{ s } k))$$

Il existe 169 relations de transitivité de ce type.

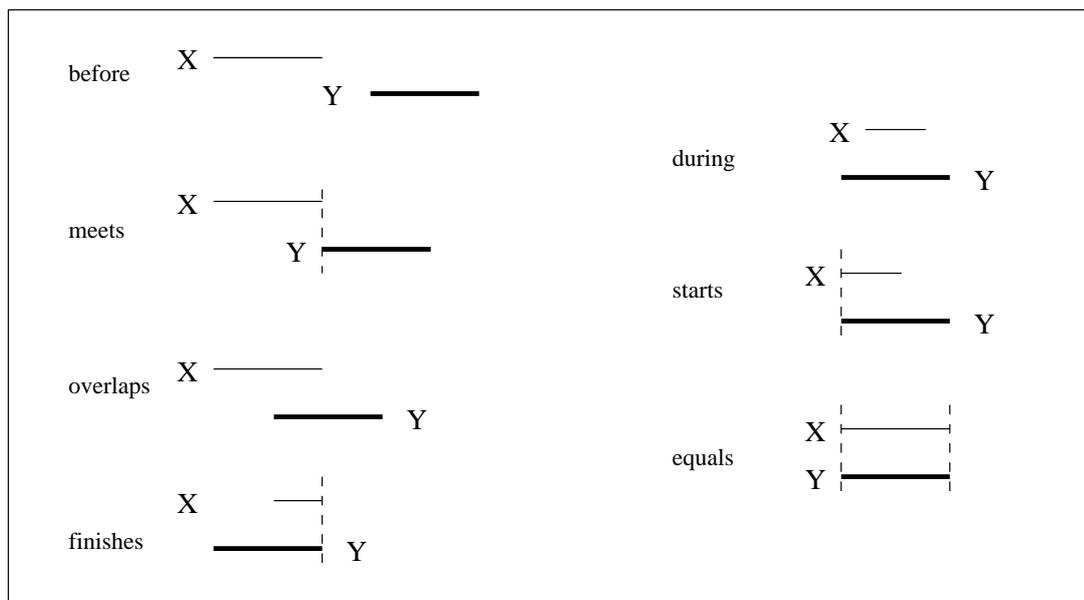


FIG. 2.1: Les relations d'Allen

- on peut également définir des relations complexes, comme la relation d'inclusion, par simple disjonction de relations primitives :

$$\forall i \forall j (i \text{ in } j) \Leftrightarrow ((i \text{ d } j) \vee (i \text{ s } j) \vee (i \text{ f } j))$$

Deux intervalles peuvent donc être reliés par une relation primitive, mais aussi par une relation complexe ; il est ainsi possible de représenter une connaissance incomplète des relations.

La connaissance temporelle sur un ensemble d'intervalles peut être représentée par un réseau de contraintes. Il s'agit d'un graphe orienté dont les nœuds représentent les intervalles et dont les arcs sont étiquetés par la relation entre les intervalles. Ce graphe peut ensuite être complété grâce aux axiomes de transitivité décrits plus haut. Allen [Allen, 1983] donne un algorithme de propagation de contraintes basé sur la fermeture transitive.

L'exemple suivant, très simple, permet d'illustrer rapidement le raisonnement sur les intervalles :

(2.6) Paul entra dans la pièce (1). Marie regardait la télévision (2). Elle l'éteignit (3).

(1) et (2) introduisent l'assertion temporelle suivante :

$$I_{\text{entrer}} \text{ d } I_{\text{regarder\_television}}$$

Puis, en examinant (2) et (3), on obtient :

$$I_{\text{regarder\_television}} \text{ m } I_{\text{eteindre-television}}$$

Le graphe des relations est celui de la figure 2.2a (les arcs représentant les relations inverses sont implicites).

Enfin, la relation de transitivité :

$$\forall i \forall j \forall k ((i \text{ d } j) \wedge (j \text{ m } k) \Rightarrow i < k)$$

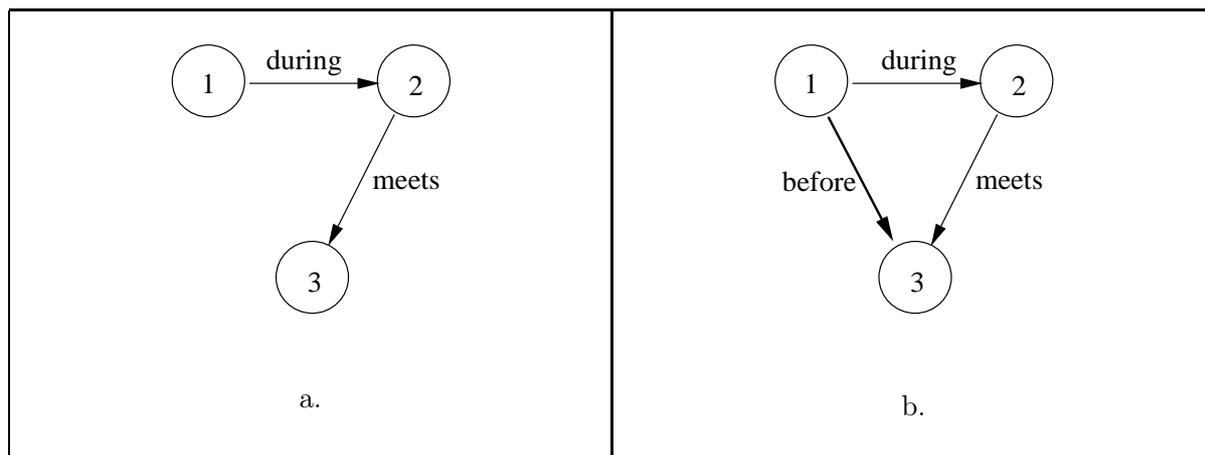


FIG. 2.2: Graphe des relations d'Allen

... permet de compléter ce graphe<sup>1</sup>, pour obtenir finalement celui de la figure 2.2b.

## 2.2 Le temps dans la langue

Abordons maintenant le problème du temps au niveau de son expression dans le langage naturel. Il s'agit donc de répertorier les temps grammaticaux et les marqueurs temporels de la langue et d'étudier leur fonctionnement.

### 2.2.1 Les temps grammaticaux

Le mot "temps" recouvre plusieurs significations en Français, et il est nécessaire pour la compréhension de distinguer le **temps grammatical**\* du **temps notionnel**. Le second ("time" en Anglais) est ce que l'on représente habituellement par une ligne droite et infinie, avec un point marquant le présent et séparant le passé du futur. Le temps grammatical ("tense" en Anglais) désigne les marques linguistiques utilisées pour exprimer le temps notionnel dans le langage (l'imparfait, le présent de l'indicatif, etc...).

L'ensemble des temps grammaticaux utilisés en Français est résumé dans le tableau de la figure 2.3.

#### Le modèle de Reichenbach

Reichenbach a proposé, pour modéliser la sémantique des temps grammaticaux, les trois repères suivants :

**E** le moment de l'événement

**S** le moment de l'énonciation ou de la parole (*Speech Time*)

**R** le moment de référence

Les relations possibles entre deux repères sont la relation de concomitance (simultanéité) notée "," et la relation de précédence notée "\_". La nouveauté réside surtout dans l'ajout

<sup>1</sup>La propagation des contraintes dans le graphe (saturation) permet également – mais pas de façon certaine – de détecter d'éventuelles incohérences dans les relations établies.

Formes simples		Formes composées	
<i>Indicatif</i>			
je chante	Présent	j'ai chanté	Passé composé
je chantais	Imparfait	j'avais chanté	Plus-que-parfait
je chantai	Passé simple	j'eus chanté	Passé antérieur
je chanterai	Futur simple	j'aurai chanté	Futur antérieur
<i>Subjonctif</i>			
que je chante	Présent	que j'aie chanté	Passé
que je chantasse	Imparfait	que j'eusse chanté	Plus-que-parfait
<i>Conditionnel</i>			
je chanterais	Présent	j'aurais chanté	Passé 1 <sup>ère</sup> forme
		j'eusse chanté	Passé 2 <sup>nde</sup> forme
<i>Impératif</i>			
chante	Présent	aie chanté	Passé
<b>Temps impersonnels</b>			
chantant	Participe présent	chanté	Participe passé
chenter	Infinitif présent	avoir chanté	Infinitif passé

FIG. 2.3: Les formes verbales du Français

d'un moment de référence, qui permet de prendre en compte certains temps composés. Ainsi, la représentation de quelques temps grammaticaux à l'aide du modèle de Reichenbach est la suivante :

Passé simple	<i>je vis Paul</i>	<b>E,R_S</b>
Plus-que-parfait	<i>j'avais vu Paul</i>	<b>E_R_S</b>
Futur	<i>je verrai Paul</i>	<b>S_E,R</b>
Futur antérieur	<i>j'aurai vu Paul</i>	<b>S_E_R</b>

Dans le cas du plus-que-parfait (*j'avais vu Paul*), par exemple, le moment de l'événement est le moment où la personne qui parle a vu Paul, et le moment de référence est situé entre cet événement et le moment où la personne parle. La position exacte de R doit être déterminée par le contexte, on sait seulement qu'au point R, la personne qui parle a "déjà" vu Paul.

Par la suite, Yip a suggéré de remplacer ces relations par des relations d'Allen (voir section 2.1.4). Ainsi, on obtient les représentations de la figure 2.4. Ceci peut déjà être intégré dans une réflexion sur le discours. En effet, en ajoutant par exemple comme contrainte que les moment d'élocution S sont égaux lors de l'énonciation de plusieurs phrases, on peut en déduire certaines disjonctions de relations entre les événements (voir aussi section 5).

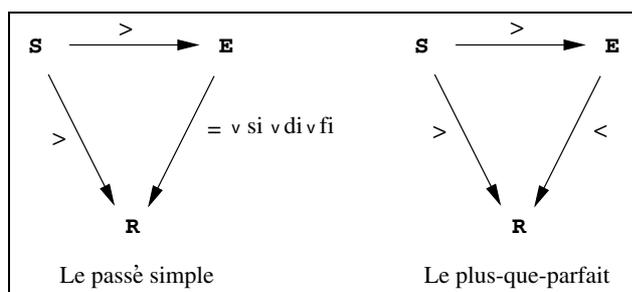


FIG. 2.4: Exemples de représentations : modèle de Reichenbach et relations d'Allen.

### 2.2.2 Les connecteurs et adverbiaux temporels

Un *connecteur temporel*, ou plus précisément ici une conjonction de subordination temporelle, introduit et précise une proposition subordonnée, comme dans :

- (2.7) Je partirai quand tu reviendras.  
(2.8) Bill s'occupa de la maison pendant que Georges grandissait.  
(2.9) Charles prenait des vacances avant qu'on ne lui demande de revenir.

Un *adverbial* est un élément (mot ou groupe de mots) ayant une fonction similaire à celle d'un adverbe ou d'un complément circonstanciel, c'est-à-dire qu'il modifie le verbe auquel il est rattaché. On peut le supprimer sans rendre la syntaxe ni la sémantique de la phrase incorrectes. Ainsi, les passages soulignés des exemples suivants ont une fonction adverbiale temporelle :

- (2.10) Paul arrivera demain.  
(2.11) Marie est revenue à cinq heures.  
(2.12) Victor a écrit son livre pendant toute la nuit.  
(2.13) Napoléon se maria la veille de son couronnement.

On peut distinguer ([Bras, 1990]) :

- Les adverbiaux de référence temporelle dont le rôle est d'exprimer la localisation d'un événement dans le temps : *à cinq heures, demain*
- Les adverbiaux de durée : *pendant une heure, en trois jours*
- Les adverbiaux de fréquence : *souvent, tous les mois, deux fois par jours*
- Les adverbiaux itératifs : *trois fois, plusieurs fois*
- Les adverbiaux de quantification : *toujours, quelquefois*
- Les adverbiaux présuppositionnels : *encore, déjà*

Dans le cadre de notre travail, nous avons par la suite opéré une distinction plus affinée pour le traitement des connecteurs et des adverbiaux (section 3.4).





## Deuxième partie

# Calcul des relations temporelles



L'objectif final du travail dans lequel nous nous sommes engagé est de réaliser un calcul des relations temporelles dans un discours écrit en Français. Il s'agit d'un but fixé à long terme, dont la durée du stage ne permettait bien évidemment pas de traiter tous les aspects en profondeur.

Dans la perspective de cet objectif, mais aussi dans le cadre plus général du temps dans le langage naturel, de nombreuses contributions sont à apporter, aussi bien sur le plan théorique que sur le plan pratique.

La première question lorsqu'on aborde un tel domaine est celle de la représentation formelle que l'on va utiliser ou mettre en place. En ce qui concerne le calcul des relations temporelles entre les événements dans le discours, aucune ontologie\* ne s'est encore véritablement imposée. Avant de commencer, il fallait donc se poser les questions suivantes :

1. *Qu'est-ce qu'un événement ?* Quelles catégories grammaticales expriment l'existence d'une entité (ou information) temporelle ?

- Les verbes ? Cela semble logique, et pourtant tous les verbes ne sont pas égaux devant l'événement. Leur aspect importe [Vendler, 1967; Steedman, 1997; Vet, 1994]. Même grossièrement, on peut distinguer au moins les événements (actions qui changent les caractéristiques du monde, comme la phrase 2.14) des états (qui décrivent ce monde, comme l'exemple 2.15)

(2.14) Victor écrivit un sonnet (en deux heures).

(2.15) Victor écrivait un sonnet.

- Les syntagmes nominaux ? Si certaines propositions de *tous* les considérer comme des événements (un *chien* consistant en un événement durant de la naissance à la mort de l'animal) semblent trop complexes à mettre en place, il reste indispensable de ne pas oublier des informations temporelles précieuses fournies par des groupes nominaux, comme dans les exemples suivants :

(2.16) L'équilibre international a profondément changé après la chute du mur de Berlin.

(2.17) L'intervention du chef de l'État a mis fin aux rumeurs.

Dans l'impossibilité de reconnaître efficacement l'aspect d'un verbe de façon automatique, nous choisissons de considérer tous les verbes comme des événements. En ce qui concerne les syntagmes nominaux, nous ne conservons que ceux qui interviennent directement dans un adverbial temporel\*, comme dans l'exemple 2.16 (*après NP, pendant NP, ...*).

2. *Quelles relations peuvent exister entre des événements ?* Le nombre de relations temporelles possibles varient selon les chercheurs, avec des sémantiques différentes. Andreas Setzer [Setzer, 2001] en utilisent cinq (BEFORE, AFTER, INCLUDES, IS\_INCLUDED, SIMULTANEOUS), Franck Schilder et Christopher Habel en proposent sept [Schilder and Habel, 2001], d'autres ([Grover *et al.*, 1994; 1995]) quatre (JUST\_AFTER, PRECEDE, OVERLAP, SAME\_EVENT).

Quant à nous, nous distinguerons l'analyse automatique de l'annotation humaine (que nous aurons besoin de mettre en place pour la validation de notre travail, mais qui comporte aussi un intérêt linguistique en soi). Pour le calcul automatique des dates et des intervalles de temps, il serait en effet dommage de ne pas utiliser les treize relations d'Allen (voir section 2.1.4), qui ont donné lieu à la mise en place d'une algèbre très efficace. En revanche, un humain peut difficilement annoter sans erreur un texte de plusieurs dizaines d'événements en choisissant dans un panel de treize relations. Nous choisirons donc pour lui des disjonctions des relations d'Allen, que nous nommerons

---

BEFORE, AFTER, OVERLAPS, IS\_OVERLAPED, INCLUDES, IS\_INCLUDED, avec :

$$\forall i \forall j (i \text{ before } j \Leftrightarrow ((i < j) \vee (i \text{ m } j)))$$

$$\forall i \forall j (i \text{ after } j \Leftrightarrow ((i > j) \vee (i \text{ mi } j)))$$

$$\forall i \forall j (i \text{ overlaps } j \Leftrightarrow ((i \text{ o } j)))$$

$$\forall i \forall j (i \text{ is\_overlaped } j \Leftrightarrow ((i \text{ oi } j)))$$

$$\forall i \forall j (i \text{ includes } j \Leftrightarrow ((i \text{ di } j) \vee (i \text{ si } j) \vee (i \text{ fi } j) \vee (i = j)))$$

$$\forall i \forall j (i \text{ is\_included } j \Leftrightarrow ((i \text{ d } j) \vee (i \text{ s } j) \vee (i \text{ f } j) \vee (i = j)))$$

Remarquons que bien sûr toutes les relations d'Allen sont représentées une et une seule fois dans ces disjonctions (ce qui n'est le cas dans aucune des théories présentées ci-dessus ; leurs sémantiques sont par ailleurs imprécises). Cela préserve les propriétés de l'algèbre et permet de passer sans problème de nos relations à celles d'Allen. La relation d'égalité fait exception, elle est utilisée dans les deux relations d'inclusion ; mais on considère qu'il s'agit de "*equals*" et de son "inverse". Il s'agit d'une entorse au principe de mutuelle exclusion des relations, mais cela permet de conserver le principe d'inversion qui veut que l'inverse d'une relation appartienne également à l'ensemble des relations définies (ainsi INCLUDES est bien l'inverse de IS\_INCLUDED et *vice-versa*).

3. *Comment déterminer à quel événement réfère un adverbial temporel\** ? Autrement dit, comment va-t-on séparer notre texte, une fois l'analyse syntaxique effectuée, en clauses dans lesquelles tous les éléments pourront être rattachés à un événement au maximum ? Nous donnerons notre réponse à cette question plus loin.

4. *Comment représenter la structure temporelle du discours* ? Sans originalité, nous choisissons un graphe orienté dans lequel les nœuds sont les événements du texte et les arcs sont les relations (ou les disjonctions de relations) d'Allen entre ces événements. Notons que si un arc existe entre les événements  $e1$  et  $e2$ , l'arc contenant les relations "inverses" au premier en est déduit entre  $e2$  et  $e1$ . Par exemple :

$$\forall i \forall j ((i \text{ b } j) \vee (i \text{ m } j)) \Leftrightarrow ((j \text{ bi } i) \vee (j \text{ mi } i))$$

Pour compléter à chaque étape les relations obtenues par notre traitement, il sera nécessaire de saturer ce graphe, c'est-à-dire de propager toutes les contraintes à travers les arcs par transitivité. L'algorithme de propagation est donné en annexe C.

Les chapitres qui suivent relatent les étapes que nous avons effectuées pour arriver à un calcul des relations temporelles du discours. L'analyse syntaxique ne concerne pas à proprement parler l'aspect temporel, mais les travaux réalisés jusqu'à présent dans ce domaine n'opèrent pas une analyse suffisamment fine des adverbiaux temporels. Il s'agit donc d'un premier passage obligé. Puis nous exposerons la gestion des adverbiaux temporels, les indicateurs prioritaires du temps dans les phrases, suivie de l'analyse des relations entre les temps grammaticaux. Nous avons ensuite mis en place une étape de validation pour évaluer nos résultats. Cette validation n'est pas encore terminée, nous en exposerons donc les principes, mais pas les résultats.

## CHAPITRE 3

# Les fondations : l'analyse syntaxique

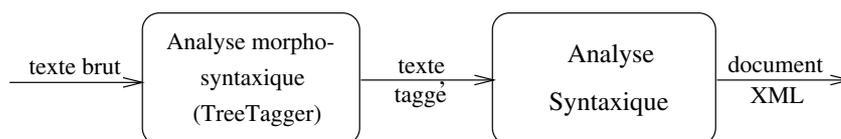
Avant d'effectuer à proprement parler un travail sur les relations temporelles dans le discours, il est nécessaire de connaître de façon assez précise la syntaxe\* du texte étudié, et pour cela d'en effectuer une analyse syntaxique\*. En effet, nous devons pouvoir reconnaître les verbes et leurs temps, les adverbiaux temporels\* de toutes sortes, ainsi que tous les éléments qui alimenteront par la suite notre théorie sur le calcul des relations temporelles.

Cette première étape, objet de nombreux travaux en ce qui concerne l'Anglais, reste assez peu avancée pour le Français. En particulier, les analyseurs syntaxiques généraux existant ou en projet ne traitent pas le problème du temps de façon suffisamment fine [Abeillé and Clément, 2003]. De plus, pour des analyses en profondeur se pose très vite le problème de la complexité computationnelle. Nous ne prétendons bien sûr pas surpasser ces analyseurs, mais tout simplement créer un outil plus adapté à notre besoin : une analyse en surface de la grammaire générale, c'est-à-dire partielle et approximative, accompagnée d'un traitement approfondi des éléments apportant des informations temporelles.

Nous allons séparer ce travail en deux parties :

- L'analyse morpho-syntaxique nous permet de connaître la catégorie grammaticale de tous les mots que contient le texte fourni en entrée. Il s'agit uniquement d'une analyse mot-à-mot, qui ne regroupe pas les unités élémentaires en syntagmes.
- L'analyse syntaxique utilise les catégories grammaticales de la première phase pour regrouper les mots et former des syntagmes\*, comme on l'a vu à la section 1.2.

XML est un langage à balises qui convient parfaitement à la représentation et au traitement des structures syntaxiques d'un texte. Nous utiliserons donc ce langage dès la sortie de l'analyseur syntaxique.



Un outil effectuant la reconnaissance morpho-syntaxique\* des fonctions grammaticales existe, il s'agit de *TreeTagger*, que nous utilisons pour poser les premiers piliers de notre entreprise. Nous avons ensuite décrit et implémenté les règles de la grammaire française, et en particulier les règles de l'expression de la temporalité. Les sections qui suivent détaillent ces travaux préliminaires.

### 3.1 Analyse morpho-syntaxique

La première étape à accomplir est d'effectuer une analyse grammaticale mot-à-mot de notre texte ; c'est ce que l'on appelle l'analyse morpho-syntaxique. Le logiciel *TreeTagger*, un outil développé par Helmut Schmid, de l'Université de Stuttgart, permet d'effectuer cette analyse pour des textes en plusieurs langues, dont le Français.

Ainsi, dans la version française, pour l'entrée suivante :

(3.1) Je suis le roi du monde.

... la sortie sera :

je	PRO:PER	je
suis	VER:pres	suivre être
le	DET:ART	le
roi	NOM	roi
du	PRP:det	du
monde	NOM	monde
.	SENT	.

La première colonne reproduit le texte entré, la deuxième donne la fonction du mot dans la phrase (ici, pronom personnel, verbe au présent, article défini, nom, préposition et point). Enfin, dans la troisième colonne sont indiqués les *lemmes*\* correspondants, c'est-à-dire la racine du mot. Dans l'exemple, le lemme du verbe est sa forme infinitive. Pour le mot *chevaux* serait indiqué *cheval*, etc... Ceci permet une utilisation plus aisée des données par la suite.

Expliquons en quelques phrases comment procède *TreeTagger*. Le problème principal réside dans le fait que dans un langage, la plupart des mots sont ambigus, c'est-à-dire qu'ils peuvent avoir plusieurs catégories différentes : par exemple, le mot *été* peut être à la fois un nom commun (la saison), et le participe passé du verbe être ; le mot *orange* peut être un nom ou un adjectif, etc...

La méthode utilisée pour résoudre ce problème est basée sur une analyse de corpus\* à laquelle on applique une analyse probabiliste sur les séquences de mots. En effet, il est possible – c'est d'ailleurs comme cela que procède l'être humain – de déterminer la catégorie d'un mot ambigu en observant le contexte grammatical dans lequel il a été énoncé. En pratique, Schmid utilise une méthode basée sur des arbres de décisions. Pour plus de détails, vous pouvez consulter [Schmid, 1994].

Ainsi, le mot *été* sera facilement déterminé. S'il est énoncé après un pronom personnel et un auxiliaire avoir (*il a été ivre toute la journée*), la probabilité d'être en présence d'un participe passé sera proche de 1. En revanche, après un article et avant un adjectif puis un verbe (*un été chaud débuta*), *TreeTagger* penchera avec raison en faveur du nom.

De nombreuses formes sont ainsi traitées avec succès, et les concepteurs de *TreeTagger* annoncent une précision de 96,36 % pour l'Anglais [Schmid, 1994]. Notons qu'en revanche il ne s'agit là que d'une analyse syntaxique, et aucunement sémantique. Ainsi, les différents sens d'une même catégorie grammaticale (comme pour *suis* dans notre premier exemple) ne sont pas traités.

Ce traitement mot-à-mot effectué, il est nécessaire, dans un second temps, d'effectuer une analyse syntaxique de notre texte au niveau de la phrase (voir aussi section 1.2).

## 3.2 Analyse syntaxique

Comme nous l'avons suggéré dans l'introduction, nous avons fait le choix d'une analyse partielle. Cela implique de renoncer aux méthodes utilisant la récursivité, trop lourdes à mettre en place (ce qui nous posera les problèmes détaillés à la section suivante). Le mode de fonctionnement que nous avons choisi pour cette opération est la mise en cascade de plusieurs automates à états finis. Chacun de ces automates représente un niveau de reconnaissance. Au niveau le plus bas, le niveau 0, l'entrée est une suite de mots avec leur catégorie grammaticale (ce qui représente la sortie de TreeTagger, en réalité fortement modifiée – nous y reviendrons plus tard). L'automate de niveau 1 trouve toutes les séquences du niveau 0 correspondant à un certain motif exprimé à l'aide d'expressions régulières\* (voir l'annexe A) – par exemple, des expressions de dates – et les réduit à des éléments simples portant un nouveau nom – par exemple, **date**. La sortie de cet automate devient l'entrée de l'automate suivant, et ainsi de suite.

Supposons que nous ayons la phrase suivante :

(3.2) Le 2 décembre 1805, Napoléon Bonaparte devint l'empereur du peuple français.

Après passage dans TreeTagger et quelques modifications, nous obtenons une entrée pour notre premier automate. Cette entrée a le format suivant<sup>1</sup> :

<i>Le</i>	art
<i>2</i>	nombre
<i>décembre</i>	mois
<i>1805</i>	nombre
<i>,</i>	ponct
<i>Napoléon</i>	prenom
<i>Bonaparte</i>	nom_propre
<i>devenir</i>	verbe_passe_simple
<i>le</i>	art
<i>empereur</i>	nom
<i>de</i>	prep
<i>le</i>	art
<i>peuple</i>	nom
<i>français</i>	adjectif
<i>point</i>	point

Pour traiter cette phrase, considérons les règles suivantes :

```

:niveau0
  date      → art? nombre mois nombre?
  person    → prenom nom_propre?
  nom_etendu → art nom adjectif?

:niveau1
  groupe_nom → nom_etendu (prep nom_etendu)*
              | person
  verb       → verbe_passe_simple

:niveau2
  phrase     → (date ponct)* groupe_nom verbe groupe_nom point

```

Les expressions régulières utilisées dans cet exemple sont extrêmement simples : le point

<sup>1</sup>précisons que les tags ont été rendus plus explicites que les tags réellement utilisés, pour une meilleure compréhension.

d'interrogation ? signifie que l'élément est facultatif et l'astérisque \* qu'il est facultatif et qu'il peut être répété une infinité de fois. La barre verticale | représente l'opérateur logique "ou". Ainsi une date peut être un nombre et un mois seuls (*23 février*), ou accompagnés d'un article et/ou d'un autre nombre (*le 2 décembre 1804*). Un groupe nominal peut être composé d'un nom seul (*Casimir*) ou d'un nom suivis de plusieurs compléments du nom (*le mari de la concierge de mon immeuble*). Notons que ce système d'automates en cascade ne permet pas la récursivité, ce qui nous empêche de distinguer *[la concierge de mon immeuble]* comme groupe nominal puis *[le mari de [la concierge de mon immeuble]]*, ce qui serait grammaticalement plus correct. Une solution serait de multiplier les niveaux, mais de nombreux combinaisons resteraient malgré tout impossibles. Nous verrons par la suite que cette absence de récursivité pose d'autres problèmes bien plus importants.

À l'issue du niveau 0, nous aurons produit une structure de la forme :

```
[date
  [art le]
  [nombre 2]
  [mois décembre]
  [nombre 1804]]
[ponct ,]
[person
  [prenom Napoléon]
  [nom_propre Bonaparte]]
[verbe_passe_simple devenir]
[nom_etendu
  [art le]
  [nom empereur]]
[prep de]
[nom_etendu
  [art le]
  [nom peuple]
  [adjectif français]]
```

Notons que les éléments utilisés dans les expressions régulières, s'ils participent à la création d'un nouvel élément, ne sont plus accessibles au niveau suivant (par exemple, une référence à un nom au niveau 1 serait inopérante).

À la sortie du niveau 1, nos seuls éléments accessibles seront donc :

```
date  ponct  groupe_nom  verbe  groupe_nom .
```

Enfin, la sortie finale que nous obtiendrons est représentée à la figure 3.1.

Pour effectuer ce travail, nous avons utilisé l'outil *Cass*, développé par Steven Abney, de l'université du Michigan. À partir des règles que nous avons élaborées et séparées en niveau (voir annexe B), *Cass* implémente et lance les automates à états finis. Pour des explications plus détaillées du fonctionnement de *Cass*, voir [Abney, 1996; 1997].

### 3.3 Problèmes soulevés

Étant donné le fonctionnement choisi de l'analyse syntaxique, quelques problèmes se posent qui empêchent un déroulement correct de cette analyse d'une seule traite. L'écueil le plus important concerne la reconnaissance correcte et complète des propositions subordonnées.

Rappelons qu'une proposition subordonnée est une proposition placée sous la dépendance d'une autre proposition (principale, ou subordonnée d'un niveau supérieur). La subordonnée

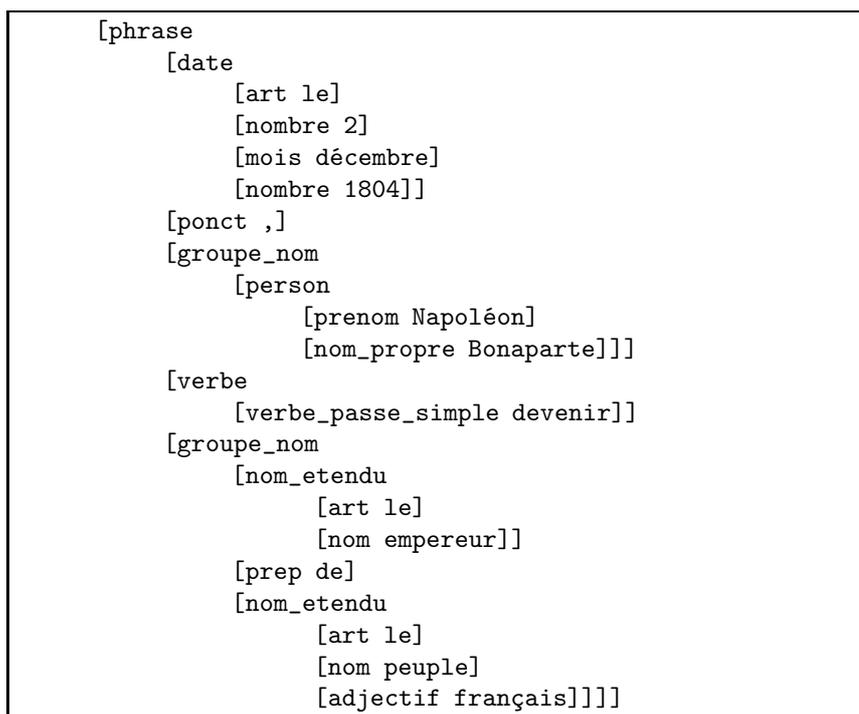


FIG. 3.1: Analyse syntaxique de la phrase 3.2

ne peut assurer un message cohérent en dehors de cette dépendance. On distingue, selon les fonctions de la subordonnée, quatre grandes catégories :

- les complétives sont, la plupart du temps, complément d’objet direct (*Je veux que tu viennes*).
- les relatives sont complément de nom ou de pronom (*Je connais l’homme qui a vu l’ours*).
- les circonstancielles, ou adverbiales, qui peuvent être temporelles, consécutives, causales, comparatives, etc. . . (*Il se mariera quand les poules auront des dents*).
- les participiales possèdent un noyau qui est un verbe au participe passé ou présent (*Mon rapport achevé, je partirai en vacances*).

Les participes passés placés en apposition et utilisés comme adjectifs, comme dans la phrase 3.3, ne sont pas des propositions subordonnées, mais posent le même type de problèmes que celles-ci.

(3.3) Ce rapport, rédigé en quelques jours, est pourtant de très bonne qualité.

Pour bien comprendre le problème, focalisons-nous sur les propositions subordonnées relatives. Une telle proposition, qui est destinée à être contenue dans un syntagme nominal (NP) peut contenir exactement les mêmes éléments qu’une proposition principale (notamment un NP !). Les règles qu’il semblerait logique de mettre en place sont celles représentées en figure 3.2, et auraient pour résultat l’arbre de la figure 3.3 pour la phrase “*Je connais l’homme que l’ours a attaqué*”. Mais comme on le voit, elles utilisent à la fois la récursivité simple (une règle qui fait référence à elle-même) et la récursivité croisée, c’est-à-dire que chaque règle contient une référence à une autre règle, qui reporte à la première.

Or, nous l’avons déjà fait remarquer, l’analyse syntaxique par cascade d’automates à états finis n’autorise pas la récursivité. Une solution envisageable pourrait alors être de multiplier

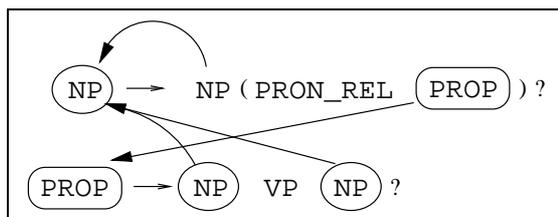


FIG. 3.2: Règles récursives et récursives croisées

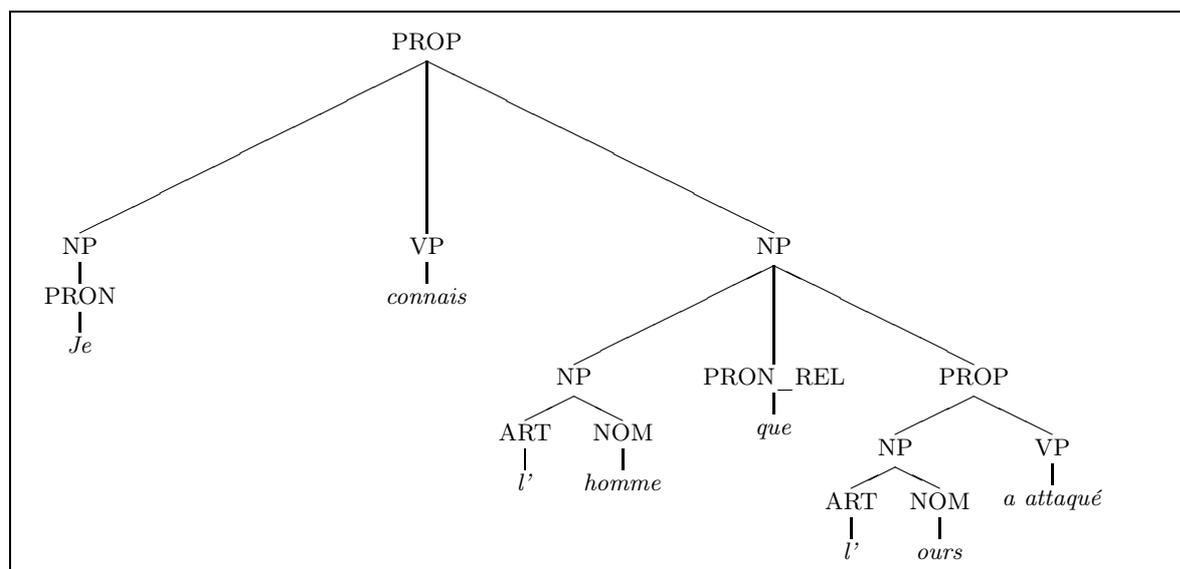


FIG. 3.3: Règles récursives appliquées au NP *l'homme que l'ours a attaqué*

les niveaux, et d'écrire par exemple :

PROP → NP VP NP?  
 NP2 → NP (PRON\_REL PROP)?  
 PROP2 → NP2 VP NP2  
 ... ..

... en superposant autant de niveaux que l'on juge nécessaire.

Cette méthode ne résoud pas le problème non plus. Examinons rapidement les deux phrases suivantes, toutes deux issues d'un article du *Monde Diplomatique* de février 2003, concernant l'enlèvement d'Ingrid Betancourt en Colombie :

- (3.4) a. Une vidéo où elle apparaît en compagnie de Clara Rojas, son amie de coeur et directrice de campagne également séquestrée, avait été diffusée par la guérilla.
- b. La vague d'attentats meurtriers que le pays a connue et les moyens donnés à la lutte contre le terrorisme par un gouvernement aux accents "bushiens" ne laissent guère de chance au dialogue.

Le résultat final de l'analyse syntaxique de la phrase 3.4a, effectuée en appliquant des règles par niveaux superposés proches de celles énoncées ci-dessus, est schématisé à la figure 3.4. Il est satisfaisant.

Traitions maintenant la phrase 3.4b avec ces mêmes règles. Un premier niveau de reconnaissance des NP nous donne le résultat suivant :

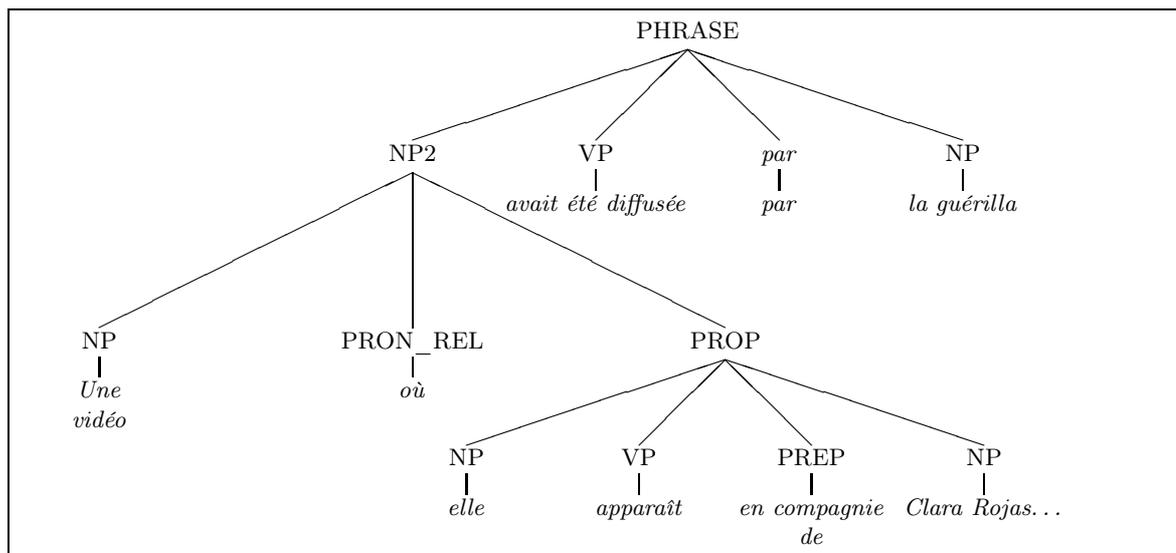


FIG. 3.4: Proposition d'analyse syntaxique de la phrase 3.4a

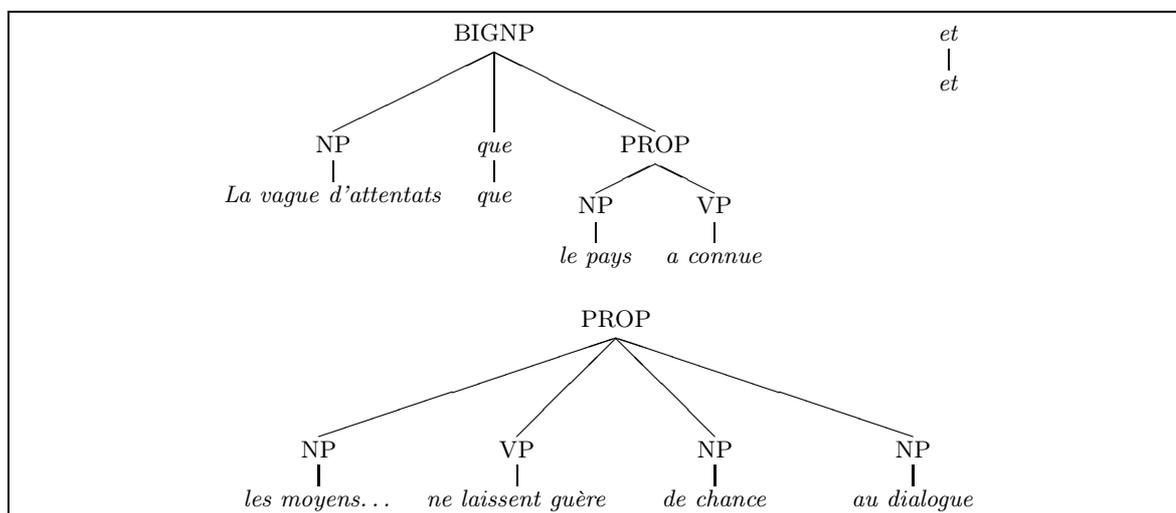
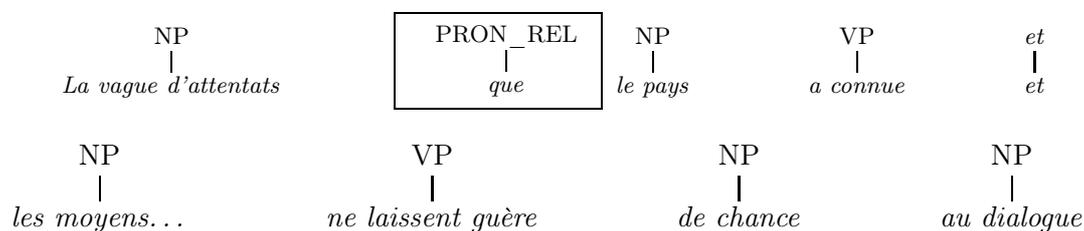


FIG. 3.5: Échec de l'analyse syntaxique de la phrase 3.4b



On voit que dès ce moment la conjonction *que* empêche un regroupement correct (la reconnaissance du groupe nominal "la vague d'attentats meurtriers que le pays a connue et les moyens [...]"). Ainsi, toute la deuxième ligne ("les moyens [...] ne laissent guère de chance au dialogue" : NP VP NP NP) sera prématurément considérée comme une proposition. On obtient finalement la représentation erronée de la figure 3.5 (voir règles ci-dessus).

La solution "idéale" semble donc être de mettre en place une méthode permettant la récursivité. En effet, des règles du type de celles vues à la figure 3.2, produiraient l'arbre de la figure 3.6 pour la phrase 3.4a.

Cet exemple concernait les propositions subordonnées relatives, mais le problème est le

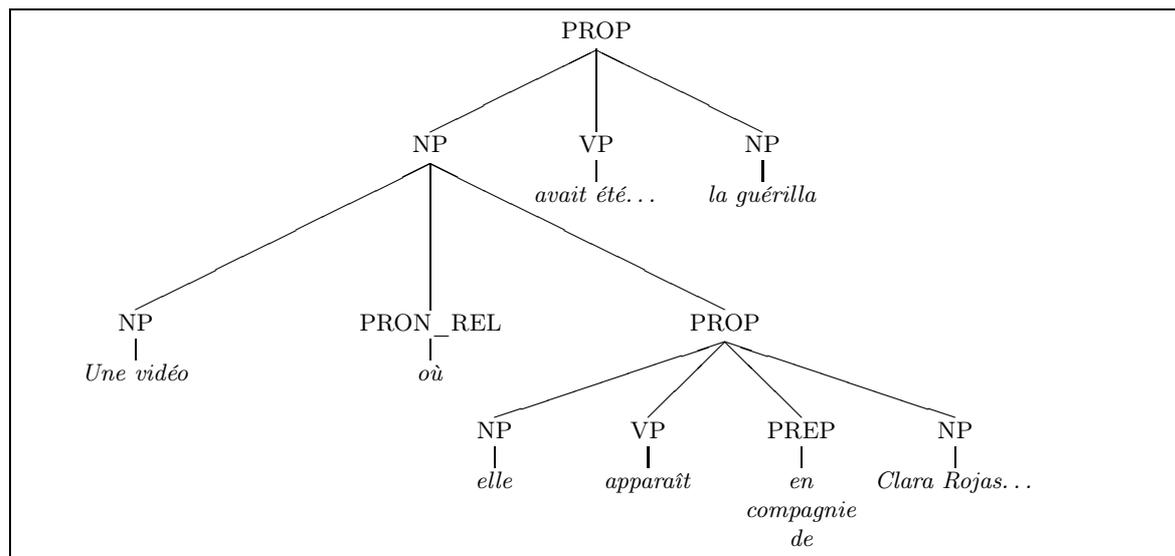


FIG. 3.6: Analyse récursive de la phrase 3.4a

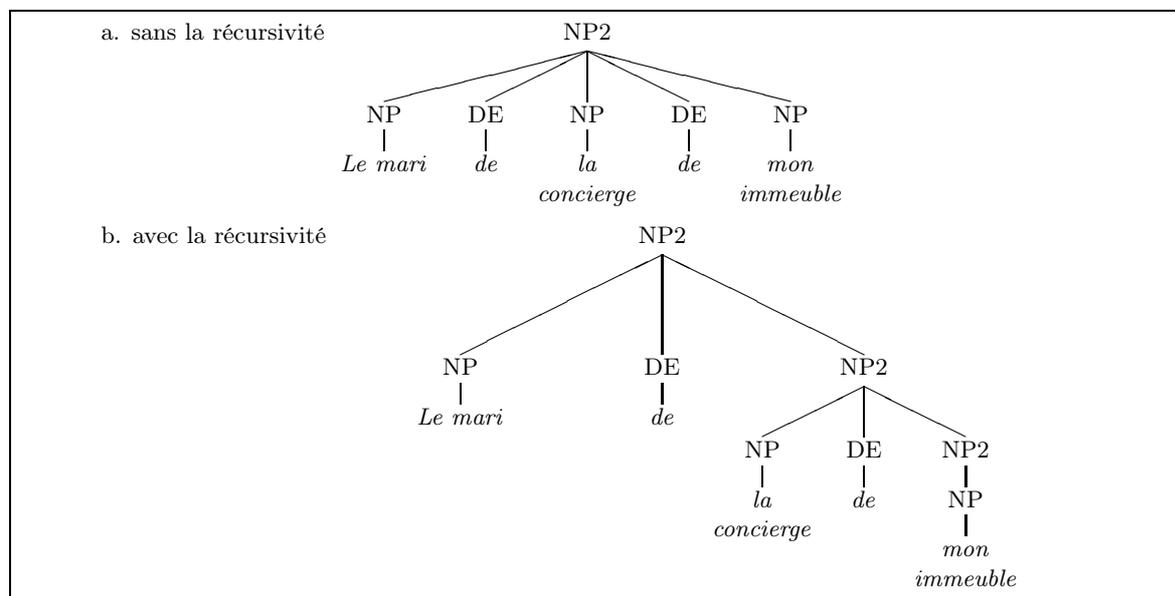


FIG. 3.7: Deux analyses du texte *le mari de la concierge de mon immeuble*

même avec les autres types de subordinées, ainsi qu'avec les participes passés placés en apposition.

De plus, la récursivité apporterait, comme on l'a déjà expliqué, des solutions plus satisfaisantes au phénomène des compléments du nom imbriqués. En effet, la règle suivante est élégante et plus conforme à la réalité :

$$\text{NP2} \rightarrow \text{NP (DE NP2)?}$$

... avec DE : la préposition *de*. Ainsi, notre exemple "*le mari de la concierge de mon immeuble*" serait traité comme le montre l'arbre de la figure 3.7.

Mais nous avons expliqué pour pour des raisons de temps et de complexité computationnelle, il était difficile de mettre en place une méthode permettant la récursivité. En pratique, il était tout de même indispensable de résoudre le problème des propositions subordinées. En effet, la reconnaissance correcte des propositions est extrêmement importante dans notre cas, puisqu'il s'agit des clauses de base que nous utiliserons par la suite. La réponse à cette difficulté est une double-exécution de la cascade d'automates : nous conservons l'ensemble de

règles présenté de façon très simplifiée plus haut. De plus, nous ajoutons des règles reconnaissant les propositions subordonnées ; par exemple, pour les relatives :

SUB → (où | que | qui) PROP

À l'issue de la première exécution, nous allons retirer les propositions subordonnées ainsi repérées, et les isoler (en pratique, les placer à la fin de la phrase). La seconde exécution reconnaîtra donc correctement la phrase épurée, ainsi que la proposition déplacée (avec un éventail de règles *ad hoc*). Ainsi, pour la phrase 3.4a, on obtient les deux propositions suivantes (incorrectes en Français mais qui facilitent le traitement) :

(3.5) Une vidéo avait été diffusée par la guérilla. Une vidéo où elle apparaît en compagnie de Clara Rojas [...].

Maintenant que nous avons esquissé les grandes lignes du fonctionnement de l'analyseur syntaxique, observons plus en détail les techniques de reconnaissance et de classement mises en place pour les éléments qui nous intéressent tout particulièrement : les adverbiaux temporels\*.

### 3.4 La gestion des adverbiaux temporels

Voyons dans les détails comment nous gérons les adverbiaux temporels\* dans l'analyseur syntaxique. Un petit lexique des éléments utilisés dans les règles qui suivent se trouve à la figure 3.8. Voir aussi l'annexe A pour une explication détaillée de la syntaxe des expressions régulières.

#### Nombres

nombre → advspec? (card ((et (un | onze)) | card)?) | onze advspec?  
avec :

- *advspec* = adverbe de spécification (*environ, exactement, ...*).
- *card* = nombre en un mot ou écrit numériquement.
- *onze* = le nombre 11 (spécifique, pour des nombres tels que *soixante et onze*).
- *un* = le chiffre *un*, considéré comme un article par TreeTagger (on le récupérera par la suite en tant qu'article également)

On traite ainsi les nombres numériques et les nombres littéraux (en un ou plusieurs mots), le tout assorti le cas échéant d'un adverbe.

#### Dates spécifiques

Avant d'aborder l'étape suivante des atomes de temps, où les unités de temps (minutes, heures, mois, années, ...) seront toutes confondues dans les termes TUNIT ou TUNITS, il faut indiquer quelques formes de dates spécifiques à certains types d'unités. *datespecabs* représente une date absolue, *datespecrel* une date qui peut être indiquée par rapport à un événement ou un focus temporel\*. Le terme *dta* indique un article.

*datespecabs* → *dta* nombre siècle  
Exemple : le XVIII<sup>ième</sup> siècle.

<b>Éléments à la sortie de TreeTagger après modifications</b>	
<b>advspec</b> : adverbe de spécification ( <i>environ, exactement, ...</i> ).	<b>month</b> : un mot représentant un mois.
<b>advtemp</b> : tous les adverbiaux temporels * reconnus.	<b>nuit</b> : le mot <i>nuit</i> .
<b>et</b> : le mot <i>et</i> .	<b>de</b> : le mot <i>de</i> .
<b>a_partir_de</b> : tous les synonymes de <i>à partir de</i> .	<b>onze</b> : le mot <i>onze</i> .
<b>après</b> : tous les synonymes de <i>après</i> .	<b>pendant</b> : tous les synonymes de <i>pendant</i> .
<b>au_debut</b> : tous les synonymes de <i>au début</i> .	<b>plusieurs</b> : tous les synonymes de <i>plusieurs</i> .
<b>avant</b> : tous les synonymes de <i>avant</i> .	<b>plus_tard</b> : tous les synonymes de <i>plus tard</i> .
<b>card</b> : cardinal, en lettres ou en chiffres.	<b>plus_tot</b> : tous les synonymes de <i>plus tôt</i> .
<b>virg</b> : virgule.	<b>pour_1_heure</b> : tous les synonymes de <i>maintenant</i> .
<b>jour</b> : un jour de la semaine.	<b>quelque(s)</b> : tous les synonymes de <i>quelque(s)</i> .
<b>debut</b> : le mot <i>début</i> .	<b>relst</b> : modificateur temporel relatif au moment d'élocution ( <i>dernier, prochain...</i> ).
<b>dta</b> : article.	<b>reltf</b> : modificateur temporel relatif au focus temporel* ( <i>précédent, suivant...</i> ).
<b>en_ce_moment</b> : tous les synonymes de <i>en ce moment</i> .	<b>saison</b> : un mot représentant une saison.
<b>fin</b> : le mot <i>fin</i> .	<b>soudain</b> : tous les synonymes de <i>soudain</i> .
<b>in</b> : préposition ( <i>à</i> en l'occurrence)	<b>un</b> : le mot <i>un</i> .
<b>modift</b> : modificateur temporel ( <i>plus de, moins de...</i> )	

<b>Éléments <u>temporels</u> générés par nos règles</b>	
<b>date(1/2)</b> : date non absolue.	<b>duree2</b> : durée contenant deux dates ( <i>du 11 février au 27 octobre...</i> ).
<b>dateabs</b> : date absolue.	<b>dureerelabs</b> : durée absolue.
<b>daterelST(1/2)</b> : date relative au moment d'élocution.	<b>dureerelST</b> : durée relative au moment d'élocution.
<b>daterelTF(1/2/3)</b> : date relative au focus temporel*.	<b>dureerelTF</b> : durée relative au focus temporel*.
<b>datespecabs</b> : date absolue, de forme particulière.	<b>nombre</b> : un dérivé de <b>card</b> , incluant les nombres composés ( <i>soixante et un, ...</i> ).
<b>datespecrel</b> : date relative, de forme particulière.	<b>tatome</b> : atome de temps ( <i>trois jours, 4 ans, ...</i> ).
<b>duree</b> : durée quelconque.	

FIG. 3.8: Lexique des éléments utilisés pour la gestion des adverbiaux temporels

```
datespecrel → dta mois de month | dta saison
```

*Exemples* : le mois de mai, l'automne.

### Atomes de temps

On regroupe tout d'abord les unités de temps, à l'aide de macros qui permettent de réunir plusieurs termes en un seul (sans les insérer à proprement parler dans un automate) :

```
TUNIT = annee | mois | semaine | jour | heure | minute
        | seconde | decennie | siecle
TUNITS = annees | mois | semaines | jours | heures | minutes
        | secondes | decennies | siecles
```

... puis on définit les atomes de temps de la façon suivante :

```
tatome → advspec? modift? (
        (un TUNIT) | (NOMBREUX TUNITS)
        )
        advspec?
```

avec :

- *advspec* = adverbe de spécification (*environ, exactement, ...*).
- *modift* = modificateur temporel (*plus de, moins de ...*)
- *NOMBREUX* = *quelques / plusieurs / nombre*

*Exemples* : Environ trois jours, plus de 5 semaines, plusieurs mois, ...

### Date générale

On distingue les dates qui constituent un NP – *date* : *le 14 juillet (fut un jour historique)* – de celles qui n'en sont pas – *date2* : *en été*. Le signe # représente un commentaire.

```
date → dta jour? (nombre (virg? et? nombre virg?)) month
        # le mercredi 25 mai, les 24, 25 et 26 avril
        | datespecrel
        # voir plus haut
        | dta nuit (de dta nombre a dta nombre) month
        # la nuit "de le" 3 "à le" 4 août
        | dta saison
        | fin month | debut month;
date2 → en (month | saison);
```

### Plus de précisions...

Introduisons d'abord quelques définitions supplémentaires :

```
DATE = date | dateabs | nombre; # dateabs : voir plus loin
MODIFTF1 = plus_tard | plus_tot;
MODIFTF2 = apres | avant;
NOMBREUX = quelque | quelques | plusieurs | nombre | des;
ADVERBES = enfin | alors | puis | pour_l_heure;
AUTOURTF = lendemain | veille;
SUITE = enfin | alors | et? puis;
```

On introduit ensuite les indicateurs de durée :

```

duree  → pendant tatome
        # pendant 3 jours
    |    pendant date;
        # durant la nuit du...

duree2 → de DATE a DATE | entre DATE et DATE;
        # du 3 aout au 4 septembre

```

Il est possible de distinguer, pour les dates comme pour les durées, si le repère temporel est donné de façon absolue (*le 18 juin 1815* ne nécessite aucun contexte pour être placé de façon unique sur l'axe des temps) ou de façon relative (*le 18 juin (dernier), mercredi soir, il y a 3 jours, ...*).

De plus, pour les données relatives, on peut déterminer si le repère auquel il faut rattacher la date ou la durée est le moment d'élocution (ST pour *Speech Time*, c'est-à-dire le moment où le discours est énoncé) ou le focus temporel (TF pour *Temporal Focus*, c'est-à-dire en général le dernier événement introduit dans le discours). C'est pourquoi les entités `datere1ST` et `datere1TF` d'une part (ci-dessous), `dureere1ST` et `dureere1TF` d'autre part (page suivante), ont été distinguées.

Quant à la différence entre `datere1ST1` et `datere1ST2`, elle se situe au niveau de la portée du modificateur temporel. Ainsi, si les phrases de l'exemple 3.6 sont énoncées le 21 avril 2003, la signification est extrêmement différente selon le cas : dans la phrase a, le locuteur a triché aux élections, car on comprend qu'il réfère au 21 avril 2002. Dans le cas b, tout va bien, puisqu'il fait allusion à l'ensemble de l'année 2002. On distingue donc les expressions qui font référence à une durée inférieure à l'unité introduite (en 3.6a, le jour pour l'année – `datere1ST1`) et celles qui représentent l'unité dans son ensemble (`datere1ST2`).

- (3.6) a. Il y a un an, je suis allé voter quatre fois.  
 b. L'année dernière, je suis allé voter quatre fois.

```

datere1ST1 → (il_y_a | dans) tatome | (date | date2) relst
            # il y a deux ans | le 5 mai dernier
    |      et? ADVERBES;

datere1ST2 → tatome relst
            # la semaine dernière
    |      aujourd'hui;
            # aujourd'hui

```

Par la suite, on a également distingué certains groupes dans les catégories citées selon qu'ils peuvent admettre ou non des syntagmes nominaux\* pour les compléter. Ainsi, l'élément `datere1TF1` regroupe les éléments qui n'acceptent aucun complément, `datere1TF2` ceux qui acceptent un complément sans préposition – *trois jours après (la chute du mur de Berlin)* – et `datere1TF3` ceux qui peuvent être suivis d'un syntagme nominal précédé de la préposition *de*, comme "*au début (de la Révolution Française)*".

```

datere1TF1 → tatome MODIFTF1
            # 3 jours plus tard
    |      soudain
            # tout à coup
    |      (dans | a) dta moment reltf
            # dans le moment suivant
    |      (date | date2) reltf?;
            # le soir du 4 aout dernier

```

```

daterelTF2      → tatome t=MODIFTF2
                  # 3 jours après
                  |      n=NOMBREUX u=TUNITS (apres | avant);
                  # quelques mois après

daterelTF3      → t=au_debut
                  # A l'origine
                  |      t=MODIFTF tatome
                  # après 3 jours
                  |      t=en_ce_moment
                  # en ce moment
                  |      (dans | a) dta t=relst u=moment
                  # dans le premier moment
                  |      dta AUTOURTF;
                  # le lendemain

dateabs         → datespecabs
                  |      (date | date2) a=nombre
                  # le 11 février 1980
                  |      a? dta s=saison a=nombre ;
                  # Au printemps 1792

dureerelST      → t=depuis tatome;
                  # depuis 1 an / depuis plus de cinq ans

dureerelTF      → t=depuis virg;
                  # depuis, ...

dureerelabs     → a_partir_de DATE ;

```

### Enfin...

Tout ceci permet de définir les adverbiaux temporels temps (`adv_temp`) de la façon suivante :

```

adv_temp  → ADVT
           |      pendant NOM
           |      REPTMP;

```

avec :

```

REPTMP = datespecabs | datespecrel | date | date2 | duree | duree2
         | daterelST1 | daterelST2
         | daterelTF | daterelTF2 NOM? | daterelTF3 (de NOM)?
         | dateabs | dureerelST | dureerelTF | dureerelabs;
ADVT = advt | soudain | en_ce_moment | pour_1_heure | apres | avant;

```

## 3.5 Conclusion

En pratique, voici les différentes étapes que nous avons effectuées :

1. Notre texte brut est l'entrée du logiciel *TreeTagger*.
2. À la sortie de *TreeTagger*, quelques modifications sont nécessaires pour faciliter l'analyse syntaxique. Quelques exemples :
  - les articles partitifs condensés sont transformés (“*du*” en “*de le*”, “*des*” en “*de les*”, etc...).
  - les mots commençant par une majuscule en cours de phrase sont marqués comme des noms propres.
  - certains groupes de mots sont regroupés (“*plus tard*” en “*plus\_tard*”, “*à l'origine*” en “*a\_l\_origine*”, etc...)

- certaines expressions idiomatiques\* sont également regroupées (“*il\_y\_a*” pour “*il y a*”, “*avec*” pour “*en compagnie de*”, “*pour\_l\_heure*” pour “*pour l’heure*”, etc. . .).

Tous ces remplacements sont effectués avec les programmes `awk` et `sed` d’Unix.

3. Les balises fournies par *TreeTagger* ne conviennent pas toujours, on veut par exemple pouvoir distinguer certains mots : les mots *jour*, *mois*, *nuit*, *veille* ne sont plus de simples noms communs, mais deviennent important pour la détection des structures temporelles. De même, *pendant*, *durant* sont des prépositions qui doivent être distinguées, tout comme les adjectifs *prochain*, *précédent*. . . De nombreux exemples existent ainsi, et l’outil *Tagfixes*, fourni avec *Cass*, permet d’effectuer ces modifications sans problème.
4. Les règles grammaticales définies, l’entrée formatée, notre texte est prêt pour être traité par *Cass*.
5. À la sortie de *Cass*, des classes Java écrites pour l’occasion transforment le résultat en XML, ce qui permet d’effectuer les modifications nécessaires, en particulier l’isolement des subordonnées justifié plus haut.

Les structures emboîtées telles que celles que l’on obtient sont représentées généralement par des documents à balises, on l’a vu dans les exemples présentés. Dès lors, il semblait logique de choisir un langage normalisé, standardisé et très couramment employé comme XML. À partir du texte brut, nous avons donc obtenu un document XML structuré de façon à être utilisé plus aisément dans les phases suivantes.

---

L’analyse syntaxique est un vaste domaine de recherche, mais elle n’était pas une priorité dans le cadre de l’objectif fixé. C’est pourquoi nous nous sommes contenté de techniques simples à mettre en œuvre et avons cherché à les adapter du mieux possible pour parvenir au but à atteindre : l’obtention d’une structure syntaxique imparfaite mais satisfaisante pour continuer le travail et appropriée pour la gestion du temps.

Autant dans le cas de *TreeTagger*, dont il faut modifier la sortie pour pallier certains manques ou certaines réponses qui ne conviennent pas tout à fait, que dans le cas de l’analyse syntaxique par automates à états finis, nous utilisons des méthodes *ad hoc* pour nous accommoder des imperfections. Dans l’hypothèse d’un travail plus approfondi sur le même sujet, sur une durée plus longue, il faudrait probablement revenir sur ces étapes pour réaliser un produit plus abouti, rigoureux et adapté.

Les fondations de notre entreprise posée, l’analyse syntaxique effectuée, nous entrons maintenant dans le vif du sujet. Le premier étage sera la collecte des marqueurs et adverbiaux temporels de base, rares mais extrêmement riches en informations.

## CHAPITRE 4

# Les adverbiaux temporels

Les adverbiaux temporels\* apportent des informations précieuses sur les relations qui lient les événements d'un texte entre eux. Les plus riches en enseignement, mais aussi les plus rares, sont bien sûr les dates absolues, c'est-à-dire qui marquent de façon unique et certaine un intervalle sur l'axe des temps (*le XX<sup>ième</sup> siècle, le 21 août 1769, etc. . .*). Même dans les articles de journaux, qui représentent l'essentiel du corpus que nous avons étudié, elles sont peu nombreuses, mais il est malgré tout très important de les collecter et de les utiliser au maximum. Ensuite viennent les dates relatives, qui apportent des repères par rapport à des événements précédemment énoncés, mais qui eux ne sont pas forcément placés sur l'axe des temps. Il s'agit d'adverbiaux comme "*le 21 avril (dernier)*", "*trois jours avant*", ou même "*quelques temps après*". Les indicateurs de durée (*pendant cent jours*) peuvent également être utiles. Une autre catégorie regroupe les connecteurs temporels, comme "*puis*", "*ensuite*", "*auparavant*". Ils fournissent nettement une relation entre l'événement décrit et celui qui le précède dans le texte. Enfin, des événements introduits par des syntagmes nominaux\*, non datables en l'absence de connaissance du monde ou du contexte, doivent être collectés (exemples : *la chute du mur de Berlin, la mort de mon chat, etc. . .*).

Le problème de leur détection et de leur classement étant supposé réglé (voir la section 3.4), deux problèmes majeurs vont se poser pour l'étude des adverbiaux temporels :

- La reconnaissance de l'événement auquel ils sont rattachés dans le texte
- La déduction des informations temporelles qu'ils délivrent

### 4.1 Reconnaissance de l'événement

Une fois que nous avons collecté tous les adverbiaux temporels\* et tous les événements\* lors de l'analyse syntaxique, il reste à déterminer à quel événement réfère chaque adverbial. Cette question est bien plus difficile qu'il n'y paraît. Cependant, le document XML que nous avons nous facilite la tâche pour y répondre de façon approximative mais assez satisfaisante pour le but recherché dans un premier temps.

Nous avons donc décidé la chose suivante : un adverbial est attaché à un événement si les deux éléments possèdent une balise "ancêtre" en commun (c'est-à-dire s'il existe une balise contenant les deux éléments). Assurons-nous maintenant qu'il est impossible de trouver un adverbial temporel attaché à deux événements. Ceci nous est fortement facilité par la méthode choisie de gestion des propositions subordonnées. En effet, dans l'exemple 4.1, une proposition est imbriquée dans l'autre, et chacune comporte un adverbial temporel : l'élément (1) doit être rattaché à la proposition subordonnée relative (en italique), tandis que le (2) correspond au verbe "*a donné*". Des configurations bien plus complexes peuvent exister, et il est alors difficile de faire la part des choses de façon automatique.

(4.1) [Le match de tennis [*que j'ai fait la semaine dernière (1)*] m'a donné des courbatures pendant trois jours (2)].

Il est expliqué dans la section 3.3 comment ce genre de cas est traité dans notre projet. C'est un travail sur la structure syntaxique (document XML) et non sur le texte brut, mais en pratique, la phrase est transformée de la façon suivante :

(4.2) Le match de tennis m'a donné des courbatures pendant trois jours (2). *Le match de tennis que j'ai fait la semaine dernière* (1).

Ici, on le voit, plus aucun problème, les propositions sont bien distinctes, chaque adverbial peut être associé à un verbe sans aucune ambiguïté : (2) avec le verbe "a donné" et (1) avec le verbe "ai fait".

Malgré cela, quelques cas peuvent subsister où une clause contient deux événements. Or il ne s'agit plus de propositions imbriquées, mais plutôt d'événements qui se suivent dans des clauses filles distinctes : il suffit alors de considérer ces clauses filles séparément, et non plus cette balise contenant deux événements.

### Reconnaissance de l'événement

Reprenons les éléments que nous possédons à ce stade. Nous avons séparé toutes les clauses événementielles de la structure syntaxique (c'est-à-dire qu'aucune balise de notre document XML ne contient plus d'un événement). De plus, les éléments remarquables des adverbiaux (année, mois...) et des événements (texte, temps...) sont collectés et placés comme arguments de balises XML. Un exemple de document XML que nous obtenons est donné à la figure 4.1, il illustre la phrase 4.3 que nous avons déjà étudiée.

(4.3) Le 2 décembre 1804, Napoléon Bonaparte devint l'empereur du peuple français.

Au final, pour chaque balise représentant un événement, nous connaissons son type et le texte énoncé. Pour chaque balise représentant un adverbial temporel, voici toutes les informations recueillies :

- le numéro (établi arbitrairement dans l'ordre chronologique de l'énonciation)
- le numéro de l'adverbial fils, s'il existe (dans la figure 4.1, la balise `date` est la fille de `dateabs`)
- le numéro de l'événement correspondant à l'adverbial étudié
- le numéro du focus temporel\*, c'est-à-dire en pratique le dernier événement rencontré avant celui que l'on étudie
- le type (i.e. le nom de la balise)

... et, éventuellement et selon les types de balises :

- l'année : valable par exemple pour une date absolue
- le mois : valable pour une date
- l'heure, les minutes, etc... : valable pour une date / heure
- la valeur : valable pour une durée ou un atome de temps, cela peut être un nombre ou un adjectif indéfini comme *plusieurs, quelques...*
- l'unité : valable pour une durée ou un atome de temps, il s'agit d'une unité de temps (*année, mois, jours, etc...*). Nous aurions pu nous passer des deux derniers paramètres et les intégrer aux précédents (mettre la valeur dans l'argument correspondant à la bonne unité, par exemple 2 dans l'argument `jours` pour la durée *2 jours*), mais nous avons choisi de ne pas le faire pour éviter des ambiguïtés lors de la programmation.

```

<phrase>
  <adv_temp>
    <dateabs a=1804>
      <date n=2 m=décembre>
        <art>
          le
        </art>
        <nombre>
          2
        </nombre>
        <mois>
          décembre
        </mois>
      </date>
      <nombre>
        1804
      </nombre>
    </dateabs>
  </adv_temp>
  <ponct>
    ,
  </ponct>
  <prop>
    <groupe_nom>
      <person>
        <prenom>
          Napoléon
        </prenom>
        <nom_propre>
          Bonaparte
        </nom_propre>
      </person>
    </groupe_nom>
    <verbe t='devenir'>
      <verbe_passe_simple>
        devenir
      </verbe_passe_simple>
    </verbe>
    <groupe_nom>
      <np>
        <art>
          le
        </art>
        <nom>
          empereur
        </nom>
      </np>
      <prep>
        de
      </prep>
      <np>
        <art>
          le
        </art>
        <nom>
          peuple
        </nom>
        <adjectif>
          français
        </adjectif>
      </np>
    </groupe_nom>
  </prop>
</phrase>

```

FIG. 4.1: Schéma XML type obtenu à ce stade pour la phrase 4.3.

- l'indicateur de position relative, valable pour les dates relatives : les mots *avant*, *après*, *il\_y\_a*, *dans*... qui indiquent si le nouvel événement se situe avant ou après le précédent.
- la précision, valable pour tout ou presque : en général, il s'agit de l'éventuel adjectif qui accompagne les adverbiaux : *exactement*, *environ*, *vers le*, *approximativement*...
- un argument supplémentaire, utilisé pour recueillir des informations diverses (pour une date, par exemple : le *matin* ou la *nuit* de tel jour, la *fin* de tel mois, ou le *printemps* de telle année...)

## 4.2 Dédution des informations temporelles

Voici la structure que nous avons choisie pour les dates, les durées, les événements et les relations :

- une durée est représentée par sept éléments, représentant les diverses échelles de temps nécessaires :

*Durée* = < *années, mois, jours, heures, minutes, secondes, millisecondes* >

exemple : la durée de 3 ans 6 mois et 3 secondes (!) sera représentée par :

< 3, 6, 0, 0, 0, 0, 3, 0 >

Le choix de s'arrêter à la milliseconde a été fait de façon arbitraire.

- une date est un intervalle entre deux instants :

*Instant* = < *année, mois, jour, heure, minute, seconde, milliseconde* >

Ici encore il s'agit d'une imprécision, on ne peut en fait représenter les dates que jusqu'à la seconde.

*Date* = < *debut, fin* >

*debut* : *Instant*

*fin* : *Instant*

exemple : la date du 14 juillet 1789 est l'intervalle compris entre les instants choisis pour indiquer le début et la fin de cette journée historique :

< < 1789, 7, 14, 0, 0, 0, 0 >, < 1789, 7, 14, 23, 59, 59, 999 > >

- un événement est représenté de la façon suivante :

*Événement* = < *type, texte, début, fin, durée* >

*type* : liste d'attributs/valeurs, représentant, par exemple, la catégorie grammaticale de l'événement, le temps du verbe, etc...

*texte* : sert uniquement à faciliter la compréhension du résultat par l'humain

*debut* : *Instant* (facultatif)

*fin* : *Instant* (facultatif)

*durée* : *Durée* (facultatif)

- chacun des éléments précédents est indexé avec un numéro unique ; les relations temporelles liant les dates et les événements sont des disjonctions de relations d'Allen.

Complétons notre exemple préféré pour mieux comprendre :

- (4.4) a. *Le 2 décembre 1804*, Napoléon Bonaparte *devint* l'empereur du peuple français.  
 b. *Un an après exactement*, il *gagna* la bataille d'Austerlitz.

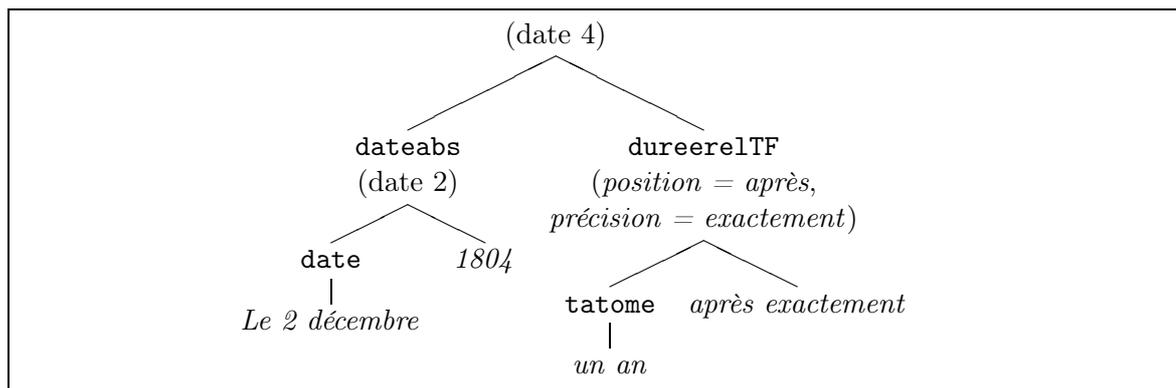


FIG. 4.2: Analyse syntaxique de deux adverbiaux temporels (exemple 4.4)

Voici les éléments repérés par notre analyse syntaxique pour les phrases 4.4 :

– pour 4.4a

- ◇ événement 1 :  $\langle \text{verbe\_passe\_simple}, \text{“devenir”}, x, y, z \rangle$
- ◇ date 2 :  $\langle \langle 1804, 12, 2, 0, 0, 0 \rangle, \langle 1804, 12, 2, 23, 59, 59, 999 \rangle \rangle$ , déterminée en deux fois comme le montre la figure 4.2
- ◇ relation temporelle entre 1 et 2 :  
 $1 \text{ s } 2 \vee 1 \text{ d } 2 \vee 1 \text{ f } 2 \vee 1 \text{ e } 2$

– pour 4.4b

- ◇ événement 3 :  $\langle \text{verbe\_passe\_simple}, \text{“gagner”}, x', y', z' \rangle$
- ◇ date 4 :  $\langle \langle 1805, 12, 2, 0, 0, 0 \rangle, \langle 1805, 12, 2, 23, 59, 59, 999 \rangle \rangle$ , déterminée en fonction de la date 2 (figure 4.2)
- ◇ relation temporelle entre 3 et 4 :  
 $3 \text{ s } 4 \vee 3 \text{ d } 4 \vee 3 \text{ f } 4 \vee 3 \text{ e } 4$

Notons que la disjonction des relations temporelles *starts*, *during*, *finishes* et *equals*, utilisée à deux reprises, représente l’inclusion au sens large (inclu ou égal). En effet, on peut dire qu’un événement qui a lieu à une certaine date, sans plus de précision, est *inclu* dans cette date.

De plus nous avons mis en place un module de comparaison des dates entre elles, ce qui permet, par exemple, de savoir que la date 2 est inférieure à la date 4 :

$$2 < 4$$

Cela nous permettra plus tard d’obtenir un lien entre les événements 1 et 3, grâce à la transitivité des relations, vue à la section 2.1.4.

### 4.3 Conclusion

La construction de notre étage est terminée. À partir d’un document XML décrivant la syntaxe du texte étudié, nous avons collecté toutes les informations nécessaires à l’étude des adverbiaux temporels\*. Nous avons déterminé de façon satisfaisante (mais imparfaite) à quels événements était lié chacun de ces adverbiaux, et nous avons déduit de tout cela les premières relations qu’il existait entre les événements. La suite de notre travail va maintenant consister à étudier les verbes eux-mêmes, et principalement leur temps.



## CHAPITRE 5

# Traitement des temps grammaticaux

Comme nous l'avons déjà entrevu à la section 2.2.1 les temps grammaticaux ont une grande importance dans la détermination de la position des référents introduits sur l'axe des temps. Nous avons expliqué comment, en associant la représentation de Reichenbach et l'algèbre d'Allen, on pouvait arriver à déduire des disjonctions de relations entre deux événements. Mais en pratique, dans le langage courant, les différentes utilisations de certains temps peuvent contredire ces représentations logiques. Notre travail s'est alors inspiré de celui de Janet Hitzeman, Marc Moens et Claire Grover [Grover *et al.*, 1994; 1995], que nous allons présenter avant d'expliquer comment notre travail se distingue et comment cette partie s'intègre aux résultats déjà obtenus par l'étude des adverbiaux temporels.

### 5.1 Relations entre les événements verbaux

Le but de Hitzeman, Moens et Grover est de réduire l'ambiguïté des structures temporelles. Dans cette optique, ils ont tout simplement établi une liste des relations temporelles qu'ils jugeaient possibles entre deux événements décrits par des verbes. Pour cela, ils ont utilisé les temps grammaticaux et l'aspect\* de ces verbes. Cette liste a été élaborée pour la langue anglaise et les temps du passé suivants : le "simple past" et le "past perfect". Le tableau de la figure 5.1 donne un court exemple de la liste ainsi obtenue.

Nous avons repris le même principe pour le Français, en élargissant le nombre de temps concernés et en modifiant les points suivants :

- L'utilisation des aspects combinés aux temps est très intéressante et permet sans aucun doute de réduire les contraintes de façon plus efficace qu'en utilisant uniquement les temps. Mais notre travail était plus axé sur la pratique ; or il s'avère difficile pour un humain et *a fortiori* pour un système automatique de distinguer sans erreur les différents aspects dans un texte. Nous nous sommes donc limités à l'étude des temps grammaticaux.

phrase 1 (S1)	phrase 2	Relation	Exemple
past event	past event	just_after S1 precede S1 overlap S1 same_event S1	Marie pushed John. He fell. John fell. Mary pushed him. NO I assembled my desk myself. The drawers only took me ten minutes.

FIG. 5.1: Exemple de la liste de Hitzeman, Moens et Grover

phrase 1	phrase 2	Relation	Exemple
PQP	PS	before	Sam était arrivé à la maison. Il frappa la porte.
		after	∅
		overlaps	Gringoire [...] était resté sur le pavé [...]. Il reprit ses sens. <i>Notre-Dame de Paris</i>
		is_overlaped	∅
		includes	∅
		is_included	∅

FIG. 5.2: Exemple de notre liste

- Les relations choisies par [Grover *et al.*, 1994] (JUST\_AFTER, PRECEDE, OVERLAP, SAME\_EVENT ou SIMULT) nous ont semblé sujettes à critiques. Tout d’abord, elles étaient inadaptées à notre choix de base qui était les relations d’Allen, puisque certaines de celles-ci n’étaient intégrées dans aucune des relations proposées. De plus, leur dénomination n’était pas claire : certains exemples donnés semblaient indiquer que SAME\_EVENT englobait l’inclusion ; JUST\_AFTER n’était pas très explicite quant à la signification de “JUST”. Pour nous, il était toutefois impossible de choisir les treize relations primitives d’Allen, beaucoup trop précises pour le travail à accomplir. Nous avons donc opté pour les six relations BEFORE, AFTER, OVERLAPS, IS\_OVERLAPED, INCLUDES et IS\_INCLUDED, détaillées en introduction de cette partie à la page 31.

La figure 5.2 donne un exemple de notre liste. Vous pouvez consulter sa version complète en annexe D.

## 5.2 Incorporation à l’étage précédent

En appliquant les règles ainsi établies aux paires d’événements verbaux d’un texte, nous obtenons de nouvelles disjonctions de relations d’Allen. Ainsi, pour l’exemple suivant (extrait de *Relation d’un voyage d’exploration au Soudan (1863-1866)* d’Eugène Mage) qui associe un plus-que-parfait à un passé simple, la consultation de la liste de la figure 5.2 nous propose la disjonction de BEFORE (relations de Allen *before* et *meets*) et OVERLAPS.

- (5.1) Il s’était couché (1) dans des herbes épaisses et hautes de 4 à 5 mètres ; cela nous donna (2) bien de l’inquiétude.

Ainsi, connaissant le lien entre ces relations et celles d’Allen (page 31), on obtient la disjonction de contraintes suivantes :

$$(1 < 2) \vee (1 \text{ m } 2) \vee (1 \text{ o } 2)$$

Il s’agit maintenant de coupler les informations obtenues par cette méthode avec celles de “l’étage” précédent. Comme nous l’avons expliqué dans l’introduction, nous représentons les informations temporelles par un graphe orienté. Les nœuds de ce graphe sont les événements du texte. Les arcs représentent les relations d’Allen possibles entre les nœuds. En première approximation il semblerait logique de faire l’intersection entre les relations nouvellement obtenues et les relations déjà existantes à l’issue de l’étape précédente. L’exemple suivant, issu d’un article du *Monde Diplomatique* de février 2003, nous montre que cette solution conduirait à des inconsistances, évitables en s’y prenant différemment.

- (5.2) Le 23 février 2002, Ingrid Betancourt [...] était enlevée (1) par les FARC. En juillet dernier, une vidéo où elle apparaît [...] avait été diffusée (2) par la guérilla.

Il n'est pas nécessaire de revenir sur le traitement des adverbiaux temporels. Connaissant la date de parution de l'article, on conclut aisément que l'expression "*en juillet dernier*" réfère au mois de juillet 2003, et donc que l'événement 1 arrive sans doute possible *avant* l'événement 2 :

$$1 < 2$$

Or l'événement 1 est un verbe à l'imparfait de l'indicatif (voix passive), et l'événement 2 est au plus-que-parfait. En appliquant nos règles sur les temps grammaticaux, on obtient :

$$(1 > 2) \vee (1 \text{ m } 2) \vee (1 \text{ o } 2)$$

Si l'on enlève les marqueurs de date des deux phrases (phrase 5.3), il semble d'ailleurs logique de ne plus retenir l'hypothèse selon laquelle 1 est avant 2.

(5.3) Ingrid Betancourt [...] était enlevée (1) par les FARC. Une vidéo où elle apparaît [...] avait été diffusée (2) par la guérilla.

Si l'on choisit de faire l'intersection des deux disjonctions de relations obtenues selon les deux méthodes, on arrivera donc à l'ensemble vide. À la lumière de cet exemple, on voit que dans certains cas il faut faire un choix. Il semble logique de donner la priorité aux marqueurs de date, qui sont une marque beaucoup plus certaine de la position relative et absolue d'un événement sur l'axe des temps. Il serait malgré tout dommage de ne pas considérer la deuxième règle qui, dans la plupart des cas, nous permet de restreindre les contraintes de façon efficace.

Voici l'algorithme que nous appliquons :

1. À l'issue du marquage des adverbiaux temporels, on sature le graphe obtenu pour récupérer le maximum d'informations possibles, c'est-à-dire que l'on propage toutes les contraintes à travers les arcs (voir l'introduction et l'annexe C).
2. Pour chaque arc liant deux événements verbaux du graphe, on applique la règle des temps grammaticaux.
3. On fait l'intersection de la disjonction de relations obtenues avec l'arc déjà existant (s'il n'en existe pas, on ajoute l'arc tout simplement)
  - (a) Si l'intersection est l'ensemble vide (inconsistance), alors on renonce et on conserve l'arc original.
  - (b) Si l'intersection contient des éléments, alors on met à jour l'arc avec les relations restreintes.
4. On sature le graphe à nouveau pour mettre à jour toutes les informations temporelles.

## 5.3 Conclusion

Après le traitement des marqueurs temporels et celui des temps grammaticaux du discours, nous obtenons un graphe de relations temporelles saturé, c'est-à-dire que nous avons extrait le plus d'informations temporelles qu'il était possible de le faire à partir de notre travail. Des exemples de la sortie fournie par notre système se trouvent en annexe F. Avant d'entamer une autre phase, il est raisonnable de souhaiter connaître les performances objectives des méthodes mises en place. C'est pourquoi nous mettons en place une technique d'évaluation des résultats, ce qui permettra non seulement de continuer notre travail sur des bases solides, mais aussi de valider nos avancées auprès de la communauté scientifique.



## CHAPITRE 6

# Validation des premiers résultats

La validation, “passage obligé” de tout travail scientifique, pose plus de problèmes qu’il n’y paraît à première vue. Dans un domaine nouveau comme celui dans lequel nous avons travaillé, aucune méthode de validation spécifique n’est vraiment mise en place. Nous avons donc décidé d’une marche à suivre basée sur des comparaisons entre les résultats et des annotations effectuées par des humains, et ceci en adaptant les notions de *précision*\* et de *rappel*\* aux caractéristiques particulières des relations d’Allen.

Le processus de validation comporte deux étapes : l’annotation des textes par des humains, puis la comparaison de ces annotations et des résultats obtenus par la machine.

### 6.1 Annotation humaine

La distinction des relations temporelles dans un texte est à la fois intuitive et difficile à formaliser pour un humain. En effet, il se construit mentalement une sorte de “carte” temporelle des événements qui sont relatés, mais éprouve des difficultés à décrire clairement cette carte. Notre but est donc de lui faciliter la tâche, en lui proposant une marche à suivre détaillée et sans ambiguïté. Le produit de cette démarche, le “guide de l’annotateur”, est inséré en annexe E.

Il existe deux caractéristiques à vérifier :

- Le système attribue-t-il des dates à des événements quand cela est possible, et le fait-il correctement ?
- Et surtout, le système propose-t-il les bonnes relations entre les événements ? (Nous verrons dans la section suivante ce que l’on appelle une “bonne relation”)

De plus, les contraintes principales sont les suivantes :

- L’annotateur doit pouvoir être choisi complètement au hasard, aucun pré-requis scientifique ne peut être exigé.
- Les consignes données doivent être telles que des annotations provenant de plusieurs personnes différentes doivent avoir le moins de divergences possibles (il s’agit de valider le système, et non l’annotation humaine !).
- Par conséquent, la sémantique des relations temporelles proposées doit être claire, ainsi que les instructions de choix des événements (voir l’introduction de cette partie).

N'ayant pas une grande expérience des expérimentations effectuées auprès d'humains, notre guide d'annotation a subi plusieurs modifications au fur et à mesure de la validation, lorsque les indications s'avéraient incomplètes ou ambiguës. Nous n'avons conservé pour l'évaluation que les annotations obtenues avec la dernière version stable. Voici une explications des consignes données.

### Relations utilisées

Un humain ne peut bien sûr pas gérer les treize relations d'Allen lors de l'annotation d'un texte. Cependant, nous avons besoin de pouvoir utiliser les relations annotées, et donc de les convertir sans ambiguïté en disjonctions de relations d'Allen. C'est ce que permet l'ensemble de relations suivant :

BEFORE	$\forall i \forall j (i \text{ before } j \Leftrightarrow ((i < j) \vee (i \text{ m } j)))$
AFTER	$\forall i \forall j (i \text{ after } j \Leftrightarrow ((i > j) \vee (i \text{ mi } j)))$
OVERLAPS	$\forall i \forall j (i \text{ overlaps } j \Leftrightarrow ((i \text{ o } j)))$
IS_OVERLAPED	$\forall i \forall j (i \text{ is\_overlaped } j \Leftrightarrow ((i \text{ oi } j)))$
INCLUDES	$\forall i \forall j (i \text{ includes } j \Leftrightarrow ((i \text{ di } j) \vee (i \text{ si } j) \vee (i \text{ fi } j) \vee (i = j)))$
IS_INCLUDED	$\forall i \forall j (i \text{ is\_included } j \Leftrightarrow ((i \text{ d } j) \vee (i \text{ s } j) \vee (i \text{ f } j) \vee (i = j)))$

En pratique, nous ne proposons au sujet que les trois relations BEFORE, OVERLAPS et INCLUDES, pour ne pas l'embrouiller. En effet ces trois relations suffisent à annoter sans problème, puisque les autres ne sont que des inverses de relations proposées.

### Que va-t-on annoter ?

La question ne réside pas seulement dans la définition de l'événement (que nous avons provisoirement fixé à "verbe conjugué"), mais aussi dans la méthode d'annotation. On ne peut pas demander à un humain de fournir un graphe saturé de relations ; même sur un texte court de 20 phrases, cela représenterait dans le pire des cas  $20 !$  relations ! Nous devons donc construire un graphe avec les relations qu'il fournira et le saturer *a posteriori*.

Doit-on donner comme consigne de n'annoter que deux événements énoncés consécutivement ? Outre qu'il est difficile de déterminer si deux verbes sont consécutifs (dans le cas de propositions subordonnées par exemple), l'infatigable exemple qui suit montre que c'est insuffisant :

- (6.1)    a.    Le 2 décembre 1804, Napoléon Bonaparte devint l'empereur des Français. (1)  
           b.    Un an après exactement, il gagna la bataille d'Austerlitz. (2)  
           c.    Entre-temps, Trafalgar avait été un désastre pour la flotte française. (3)

Si on n'annoté que deux phrases consécutives, tout annotateur sensé déterminera les relations suivantes :

- (1) before (2)
- (2) after (3)

Même en saturant par la suite le graphe des relations obtenues, on omet ainsi une information pourtant évidente : Napoléon était déjà empereur lors de la bataille de Trafalgar !

- (1) before (3)

Partant de ce constat, on s'en tiendra donc à la consigne d'annoter tout ce que le sujet juge "pertinent" de relever, même si cela représente une entorse flagrante à notre volonté d'être "clair" et "précis".

Enfin, en ce qui concerne la forme, tous les travaux sur l'annotation humaine suggèrent d'effectuer celle-ci directement sur le texte, à l'aide de balises au format XML [Harper *et al.*, 2001]. Sans revenir sur ce principe, nous avons jugé ce procédé trop complexe à mettre en place dans notre cas (formation des annotateurs plus difficile, expérimentations plus longues), en comparaison des avantages qu'il procure (toujours dans notre situation). L'annotation s'est donc faite non pas sur le texte directement, mais sur des documents séparés, en transcrivant les événements relevés.

## 6.2 Méthode de comparaison des résultats

Il s'agit maintenant d'effectuer une comparaison judicieuse entre l'annotation humaine et l'annotation automatique. Nous utilisons pour cela des notions couramment employées en linguistique dès que l'on fait intervenir les statistiques : la *précision* et le *rappel*, qui sont des mesures effectuées sur des *tables de contingence*.

Pour expliquer ce dont il s'agit, plaçons-nous dans l'hypothèse d'un choix binaire (acceptation / rejet). Le tableau de la figure 6.1 donne la répartition des possibilités lors de l'évaluation de plusieurs occurrences d'un tel choix.

	L'élément est correct	L'élément est incorrect	
Le système accepte	a	b	a+b
Le système rejette	c	d	c+d
	a+c	b+d	a+b+c+d = n

FIG. 6.1: Table de contingence pour des décisions binaires

La précision est le taux d'acceptations fondées du système :

$$précision = \frac{a}{a + b}$$

Le rappel est le taux d'éléments corrects qui sont effectivement acceptés par le système :

$$rappel = \frac{a}{a + c}$$

On définit le taux d'erreur :

$$erreur = 1 - précision$$

Et on compare usuellement ces résultats à une base, définie par les taux que l'on aurait obtenus en répondant de façon aléatoire (tous les taux sont de 0.5 pour la base d'un choix binaire).

Pour plus de détails sur les probabilités et les statistiques telles qu'elles sont utilisées en linguistique, voir [Kreen and Samuelsson, 1997].

### Application aux relations d'Allen

Notre cas est beaucoup plus particulier, non seulement parce que nous sommes en présence de treize relations, mais également parce qu'il est possible pour le système comme pour l'humain de choisir plusieurs de ces relations (une disjonction de relations). Il nous a semblé

bien trop restrictif de ne considérer une réponse comme bonne que si et seulement si la disjonction choisie était la même que celle annotée. En effet, nous considérons qu'une restriction de contraintes à trois relations sur treize, par exemple, est déjà un très bon résultat, même si l'annotateur a su être plus précis. De plus, il est des cas où le système est plus précis que l'humain, tout simplement parce qu'il utilise les treize relations, tandis que l'annotateur est restreint à six.

Voici la méthode que nous avons adoptée.

### Soient

$\left\{ \begin{array}{l} S = \text{l'ensemble des relations d'Allen attribué par le système entre deux événements} \\ H = \text{l'ensemble des relations d'Allen attribué par l'humain entre deux événements} \end{array} \right.$

On traduit d'abord les relations annotées par les personnes en relations d'Allen. Puis, pour **chaque** disjonction de relations annotée par le système :

$$précision = \frac{Card(\{r / (r \in S \wedge r \in H)\})}{Card(S)}$$

... et pour **chaque** disjonction de relations annotée par l'humain :

$$rappel\ faible = \begin{cases} 1 & \text{si } \exists r (r \in H \wedge r \in S) \\ 0 & \text{sinon} \end{cases}$$

$$rappel\ fort = \frac{Card(\{r / (r \in H \wedge r \in S)\})}{Card(H)}$$

La précision et les rappels totaux sont bien sûr la moyenne des précisions et des rappels obtenus sur l'ensemble des arcs du graphe.

Autrement dit :

- la précision est le pourcentage de **relations d'Allen** proposées par le système qui sont également annotées par l'humain.
- Le rappel dit *fort* est le symétrique de la précision, soit le pourcentage de relations d'Allen annotées par l'humain qui sont également trouvées par le système. Il est plus conforme à la définition de base, mais ne prend pas en compte (il pénalise même) le fait que le système, grâce aux marqueurs temporels et à la propagation des contraintes, peut trouver des relations plus précises que l'humain.
- Le rappel *faible* tient compte de cela ; c'est le pourcentage de **disjonctions de relations** annotées par l'humain qui ont au moins un élément compris dans la disjonction proposée par le système ( $S \cap H \neq \emptyset$ ).

Le but de la mise en place de deux calculs de rappel différents est de comparer les résultats. En l'absence d'un calcul vraiment impartial, on choisit deux formules, l'une trop avantageuse et l'autre trop pénalisante. Notons enfin que, transposées dans le cas précédemment étudié du choix binaire, nos trois définitions donnent le même résultat que les définitions initiales.

Enfin le plus difficile à déterminer est le taux de base de chaque caractéristique. En toute rigueur, selon la définition énoncée plus haut, nous devrions attribuer à la base pour la

précision la probabilité d’obtenir une disjonction donnée des treize relations d’Allen, c’est-à-dire de choisir le bon nombre de relations puis la bonne combinaison, soit :

$$\frac{1}{2^{13} - 1} \approx 0,01\%$$

Cette valeur est pourtant très largement sous-évaluée, pour deux raisons :

- Toutes les combinaisons ne sont pas possibles en pratique.
- L’annotateur humain n’utilise que six relations, et même si les treize sont séparées lors de la saturation du graphe, il serait peu rigoureux de ne pas considérer cela. De plus, l’annotateur attribue rarement une disjonction de ses six relations, mais plutôt une seule.

Tout ceci nous empêche de fournir un calcul précis des taux de base de la précision et du rappel. Pour donner plus de poids à nos résultats, il faut éviter de sous-évaluer ces taux, ce qui contribuerait à décredibiliser le travail effectué. Nous allons donc tenter d’en donner une borne maximale. Celle-ci est très facile à trouver : on considère tout simplement qu’il existe six relations, et jamais de disjonction. On trouve ainsi des probabilités de  $\frac{1}{6}$ . Ainsi on peut dire que la précision et le rappel de base sont (très) inférieurs à 16,6 %.

Une fois ces bases de calculs posées, on peut procéder aux annotations et à leur utilisation. Mais ceci ne va pas sans problème...

## 6.3 “L’annotation est humaine”

Nous savions avant de débiter cette phase de validation que les enjeux et les difficultés posées par le problème de l’annotation humaine dépassaient le cadre de notre travail. Nous avons néanmoins besoin d’utiliser cette technique, qui nous paraissait la plus efficace. Les différents travaux récents à ce sujet (voir [Harper *et al.*, 2001]) prônent des méthodes nécessitant de la part des sujets un minimum de connaissance des langages à balises comme le XML, et surtout une “formation” relativement longue, avec un guide d’annotation allant parfois jusqu’à 49 pages ([Pustejovsky *et al.*, 2002]) !

Le temps dont nous disposions, ainsi que celui que nous pouvions prendre aux annotateurs volontaires et fort occupés, ne nous permettaient pas d’en faire autant. De plus, les recherches en sont toujours à leurs débuts, et même des démarches aussi poussées ne résolvent en aucun cas tous les problèmes. Enfin, ils nous semblaient que l’annotation temporelle appliquée à notre démarche n’était pas si complexe à mettre en place ; nous n’avions pas de soucis de généralisation, et nous avons posé plusieurs hypothèses simplificatrices (par exemple, nous avons limité la définition de l’événement à celle du verbe conjugué). Malgré cela, les annotations pratiquées par des personnes différentes sur un même texte se sont révélées très différentes, et ceci à plusieurs niveaux.

### 6.3.1 Distinction des événements

La définition de l’événement par le guide de l’annotation semble relativement simple. Malgré cela, les différences sont fortes : le nombre d’événements marqués varient parfois de 27 % entre deux annotations du même texte. Cela est dû essentiellement au fait que certains verbes au participe passé ou à l’infinitif sont traités par certaines personnes et pas par d’autres (volontairement ou non).

### 6.3.2 Niveau “d’inférence”

Nous étions conscient du fait que les instructions données en ce qui concerne le marquage des relations temporelles n’étaient pas assez strictes. Nous avons justifié ce choix dans la section 6.1. Les différences sont par conséquent extrêmement fortes : certains annotateurs ont fait le choix de noter les relations temporelles qui leur semblaient importantes au moment de la lecture, tandis que d’autres ont beaucoup plus réfléchi, et ont inféré le plus de relations possibles à partir des éléments temporels à leur disposition. Pour un des textes proposés, par exemple, le nombre de relations annotées va de 17 à 218 ! En moyenne, pour 100 événements, un annotateur relève 151 relations. Il s’avère d’ailleurs qu’à chaque fois que le nombre de relations est très élevé, l’analyse de l’annotation révèle de nombreuses inconsistances, c’est-à-dire des relations contradictoires.

### 6.3.3 Annotateurs spécialistes et annotateurs profanes

La plupart des sujets à qui nous avons demandé d’effectuer les annotations étaient complètement en dehors des recherches sur le temps, et même sur la linguistique informatique. En revanche quelques-uns étaient des “spécialistes” du domaine, et avaient l’habitude de manipuler des textes et des relations temporelles pré-établies. Pour les deux catégories, les réactions face au travail demandé sont les mêmes : ils jugent l’exercice relativement intuitif avant de le commencer, et s’aperçoivent très vite qu’en pratique ce n’est pas si évident qu’ils ne le pensaient. En revanche, il est intéressant d’étudier les résultats à la lumière de la distinction entre annotateurs “spécialistes” et “profanes”. Pour les premiers, il est notable que les inférences effectuées pour trouver des relations sont très “ciblées” : il est rare qu’ils inscrivent des relations temporelles que l’on pourrait déduire en fonction de celles qui sont déjà notées. En revanche, un “profane” réfléchit moins à ces aspects, et marque tout simplement ce qui lui passe par la tête.

Cette remarque n’a pas ou peu d’impact sur le résultat final, puisque nous devons de toutes façons traiter les résultats fournis en inférant les relations automatiquement. En revanche, un autre constat est plus important : la relation de chevauchement (*overlap*) n’est pas considérée de la même façon. Soit l’annotateur novice n’en comprend pas bien le sens, soit le sujet expérimenté l’utilise à outrance. . . toujours est-il que le premier utilise très peu cette relation, tandis que le second la choisit régulièrement.

## 6.4 Les résultats

On donnera d’abord l’ensemble des résultats pour chaque texte et chaque annotateur. Dans le but d’évaluer les deux méthodes mises en place (traitement des marqueurs temporels et traitement des temps grammaticaux), on calculera les valeurs de précision et de rappels sur les résultats obtenus après chacune de ces étapes. Chaque texte étudié est reproduit en annexe F.

Les annotateurs dont le numéro est suivi d’une étoile \* sont considérés comme des “spécialistes” des relations temporelles dans le discours.

### 6.4.1 Traitement des marqueurs temporels

Texte 1 (2597 mots)			
Annotateurs	Précision	Rappel fort	Rappel faible
1	45,18 %	95,10 %	98,04 %
2 <sup>1</sup>	10,55 %	80,63 %	80,63 %
3*	35,57 %	96,51 %	98,04 %
4	65,03 %	71,82 %	75,10 %

Texte 2 (1010 mots)			
Annotateurs	Précision	Rappel fort	Rappel faible
1	52,73 %	93,48 %	100 %
2*	43,36 %	76,00 %	80,00 %

Texte 3 (1557 mots)			
Annotateurs	Précision	Rappel fort	Rappel faible
1	31,72 %	83,33 %	83,33 %
2*	17,45 %	83,33 %	83,33 %
3	33,89 %	62,86 %	62,86 %

### 6.4.2 Application des règles entre les temps grammaticaux

Texte 1			
Annotateurs	Précision	Rappel fort	Rappel faible
1	46,20 %	60,67 %	90,20 %
2 <sup>1</sup>	25,14 %	73,91 %	73,91 %
3*	45,29 %	79,92 %	93,46 %
4	68,04 %	50,92 %	75,10 %

Texte 2			
Annotateurs	Précision	Rappel fort	Rappel faible
1	52,73 %	93,48 %	100 %
2*	43,36 %	76,00 %	80,00 %

Texte 3			
Annotateurs	Précision	Rappel fort	Rappel faible
1	33,17 %	45,82 %	81,82 %
2*	25,86 %	68,79 %	80,30 %
3	34,99 %	52,82 %	61,90 %

### 6.4.3 Commentaires

#### Moyennes

	Précision	Rappel Fort	Rappel faible
Marqueurs temporels	37,28 %	82,56 %	84,59 %
+ Temps grammaticaux	41,61 %	66,92 %	81,86 %

#### Marqueurs temporels

Le résultat concernant le traitement des marqueurs temporels n'est pas surprenant : il

<sup>1</sup>L'analyse de cette annotation a révélé de nombreuses inconsistances dans les relations.

exprime le fait que cette méthode est efficace, mais uniquement en présence de ces marqueurs. En effet, le rappel est élevé, ce qui signifie que le système ne choisit pas de *mauvaises* relations (soit il n'en choisit pas du tout, soit il choisit les bonnes). Il est probable que le facteur qui provoque la baisse du rappel ne soit pas les erreurs de marquage temporel, mais plutôt les différences en amont lors de la reconnaissance de l'événement.

### Temps grammaticaux

Le traitement des temps grammaticaux améliore assez nettement la précision. Il apporte en effet des informations supplémentaires appliquées à chaque paire d'événements (contrairement à la méthode précédente). En revanche, il fait baisser sensiblement le rappel, en particulier le rappel fort, ce qui est une surprise. Il semble donc que de nombreuses règles de relations entre les temps grammaticaux "oublient" des possibilités. Une étude plus approfondie des temps pour lesquels des erreurs sont notées est nécessaire ; une augmentation des possibilités augmenterait bien sûr le rappel fort, mais il ferait également probablement baisser la précision. Un compromis doit être trouvé.

### Différence entre rappel fort et rappel faible

La différence n'est pas extrêmement marquée entre les valeurs du rappel fort et du rappel faible à l'issue de la première étape. Elle est en revanche plus forte après le traitement des temps grammaticaux, pour la raison déjà discutée ci-dessus. Il semble par conséquent judicieux de conserver les deux valeurs dans l'immédiat.

Ces premiers résultats concernant les deux méthodes utilisées de traitement des marqueurs temporels et des temps grammaticaux sont encourageants. La précision est bien au-dessus du taux de base estimé, ainsi que les deux valeurs de rappel. Si cela représente un encouragement pour la poursuite des travaux, il faut rappeler que les méthodes d'annotation et de calcul des valeurs utilisées sont très imparfaites. Par conséquent, les chiffres obtenus peuvent difficilement servir de "preuve" de l'efficacité des méthodes. En revanche, il est clair qu'ils sont relativement bons, et ils incitent à continuer le travail pour parfaire les différentes étapes, y compris celle de l'évaluation.

## 6.5 Conclusion

Pour conclure notre travail, nous avons mis en place une méthode de validation des résultats obtenus. Nous avons écrit un guide de l'annotation temporelle par l'humain et proposé des valeurs particulières de *précision* et de *rappel* dans le but de comparer la sortie de notre système automatique avec le produit de réflexions humaines. Les chiffres obtenus sont prometteurs, mais de nombreuses imperfections ont été relevées au fur et à mesure de notre travail, ce qui devrait inciter par la suite à consolider les premières avancées avant de continuer les recherches.

# CONCLUSION

Au cours des premiers mois de ce stage, nous avons élaboré les premières étapes d'un calcul des relations temporelles du discours. Nous avons tout d'abord mis en place une analyse syntaxique de surface spécifique à notre dessein de représentation temporelle. Puis notre premier travail traitant à proprement parler de la temporalité a consisté à repérer les adverbiaux temporels\* dans un texte, et à en déduire dans la mesure du possible la position et l'étendue des événements sur l'axe des temps. Par la suite, une étude sur les relations temporelles rendues possibles par l'enchaînement des temps grammaticaux\* nous a permis d'ajouter aux informations temporelles déjà obtenues des contraintes supplémentaires. Enfin, nous avons mis en place une méthode d'évaluation de ces deux étapes, et nous l'avons appliqué, non sans mal, à nos résultats.

Si cette évaluation est déjà relativement satisfaisante, certaines idées d'amélioration ou de modifications nous semblent intéressantes. Leur mise en œuvre n'est pas prévue dans le cadre de ce stage, mais elles constituent autant de pistes à suivre pour une éventuelle poursuite du travail. Parmi ces possibilités :

- un travail poussé sur l'analyse syntaxique permettrait d'augmenter le taux de reconnaissance de certaines structures. En effet, il résulte de l'imperfection de l'analyse actuelle que certains adverbiaux temporels ne peuvent être rattachés automatiquement à l'événement auquel ils réfèrent, car la phrase n'est pas reconnue correctement.
- la prise en considération de nouvelles caractéristiques en ce qui concerne les verbes, comme leur aspect (distinction entre événement et état, ou plus encore), affinerait les contraintes mises en place dans les relations temporelles.
- l'élargissement de la notion d'événement, par exemple à certains syntagmes nominaux\* (comme par exemple *la chute du mur de Berlin*) n'augmenterait pas la précision du calcul, mais rendrait notre analyse plus conforme à l'esprit général qui nous anime, basé sur la meilleure "compréhension" possible du texte.
- en ce qui concerne les temps grammaticaux, l'étude effectuée semble montrer que certaines combinaisons de temps et de relations temporelles sont spécifiques à certains types de textes (textes historiques, littéraires, articles de journaux, etc...). Même si rien n'est certain à ce niveau, il est probable que la prise en compte de cette remarque améliorerait la restriction des contraintes sur les relations temporelles.
- toujours au sujet des temps grammaticaux, il est possible qu'une étude statistique sur un large corpus\* permettrait d'étudier le problème sous un angle probabiliste (probabilité de relations en présence de tel enchaînement de temps, ou en présence de tels marqueurs à portée temporelle...). Cependant, aucune réflexion approfondie n'a été menée à ce sujet, même si des travaux de ce type ont porté sur d'autres types de relations (les relations *rhétoriques*) avec un succès discutable ([Marcu and Echiabi, 2002]).

De plus, en dehors des deux méthodes que nous avons décrites, d'autres perspectives s'offrent à nous qui permettraient de compléter efficacement le calcul des relations temporelles. De façon générale, on remarque que les techniques mises en œuvre ici le sont dans un cadre limité à deux phrases consécutives. Or il est évident que la signification d'un texte en général, et sa portée temporelle en particulier, ne se limitent pas à une superposition de phrases, fussent-elles liées aux phrases adjacentes. Au cours de sa lecture l'humain établit une sorte de "carte" temporelle des informations qui lui sont communiquées ; cette carte lui permet souvent d'établir des relations temporelles, voire un placement précis sur l'axe des temps, et cela malgré une information explicite très insuffisante. Pour faire des premiers pas vers une compréhension automatique du discours encore plus évoluée, voici les pistes qu'il serait possible de suivre :

- à l'image de l'étude effectuée sur les combinaisons de temps grammaticaux deux par deux, il est envisageable qu'une réflexion portant sur des successions de temps plus complexes, liée peut-être à une théorie probabiliste, apportent des informations complémentaires.
  - un texte suit généralement un nombre limité de "fils" temporels, c'est-à-dire de points de référence auxquels la narration peut faire allusion sans introduire de marqueur temporel explicite. Souvent "entrelacés", ces fils, pour être compréhensibles par l'humain, répondent à des contraintes de temps grammaticaux, de portée des marqueurs et de pertinence des structures employées qu'il serait extrêmement intéressant d'étudier. Nous avons commencé des observations à ce sujet au début de ce stage, mais nous ne les avons pas mises en pratique, par manque de temps d'abord, mais surtout car une mise en œuvre de ce type demande la réalisation préalable d'une analyse syntaxique et sémantique bien plus évoluée que celle effectuée (non seulement une reconnaissance structurelle correcte des phrases, mais aussi une distinction des fonctions grammaticales des groupes de mots, telles que le sujet ou le complément d'un verbe, etc. . . ).
  - de nombreux sujets d'étude de la linguistique et de la linguistique informatique sont susceptibles de contribuer à l'étude temporelle du discours. Parmi les plus évidents, citons la résolution des anaphores (reconnaissance des entités auxquelles réfèrent les pronoms par exemple – voir section 1.5) : des références implicites à des événements du contexte permettent en effet de préciser les informations temporelles des nouvelles entités introduites. Citons également les représentations de la connaissance du monde (par exemple de la date de la chute du mur de Berlin), même si nous avons justement voulu nous éloigner de ces théories par souci de réalisme – de telles représentations sont en effet impossibles dans l'état actuel de la technique.
- Tous les domaines de la linguistique sont en réalité interconnectés, et chaque avancée de l'un d'eux implique des progrès globaux.

On le voit, le travail effectué lors de ce stage, s'il est relativement nouveau dans le domaine de la linguistique informatique, ne représente en aucun cas une fin en soi, mais ouvre au contraire de nombreuses portes pour le développement d'une véritable théorie temporelle du discours. Malgré sa spécificité, cette problématique utilise toutes les connaissances de la linguistique et profite des avancées pratiques et théoriques les plus générales. Ce cadre était donc parfait pour la conduite d'un stage de DEA, alliant découverte d'un vaste domaine de recherche et participation modeste mais active à ces recherches.

# GLOSSAIRE

Certaines des définitions qui suivent sont empruntées au *Dictionnaire de linguistique et des sciences du langage* [Dubois, 1994].

**adverbial** : un adverbial est un élément (mot ou groupe de mots) ayant une fonction similaire à celle d'un adverbe ou d'un complément circonstanciel, c'est-à-dire qu'il modifie le verbe auquel il est rattaché.

**analyse morpho-syntaxique** : reconnaissance des mots et de leur catégorie grammaticale dans un texte.

**analyse syntaxique** : analyse visant à obtenir de façon correcte et non ambiguë la syntaxe\* d'un texte, au moyen de règles appropriées. voir la section 1.2.

**aspect** : La notion d'aspect rend compte de la façon dont l'action exprimée par le verbe est envisagée dans sa durée, son développement, son achèvement. . .

**corpus** : ensemble de textes existant que l'on soumet à une analyse.

**événement** : aspect\* particulier d'un verbe, représentant une action qui modifie les caractéristiques du monde et qui possède un début et une fin clairement reconnaissable.

**expression régulière** : une expression régulière est représentée par une suite de symboles qui expriment d'une manière formelle la structure d'une chaîne de caractères. voir l'annexe A.

**grammaire** : description totale d'une langue, spécification formelle des structures autorisées dans cette langue.

**focus temporel** : dans un texte, moment sur l'axe des temps dont il est question lorsque la phrase étudiée est énoncée.

**idiomatique** : une expression idiomatique est une forme grammaticale dont le sens ne peut être déduit de sa structure en mots et qui n'entre pas dans la constitution d'une forme plus large : *il y a, comment vas-tu ?* sont des expressions idiomatiques.

**intransitif** : un verbe intransitif interdit la présence d'un syntagme nominal\* pour le compléter. Par exemple, le verbe *partir* est intransitif.

**lemme** : racine d'un mot, dépouillée des marques d'accord, des préfixes et des suffixes.

**ontologie** : organisation d'un domaine déterminé en différentes classes d'objets.

**pragmatique** : étude de la signification des énoncés en lien avec le contexte (interlocuteurs, phrases précédentes, connaissance commune du monde, . . .).

**précision** : Pourcentage des éléments compris dans un ensemble testé appartenant également à un ensemble témoin.

**rappel** : Pourcentage des éléments compris dans un ensemble témoin appartenant également à l'ensemble testé.

**sémantique** : étude de la signification des énoncés, indépendamment de tout contexte.

**syntagme** : Suite d'éléments linguistiques formant un constituant d'une unité de rang supérieur.

**syntagme nominal** : Suite de mots formant un groupe ayant le rôle grammatical d'un nom. En Anglais : *noun phrase*.

**syntagme verbal** : Suite de mots formant un groupe ayant le rôle grammatical d'un verbe intransitif.  
En Anglais : *verbal phrase*.

**syntaxe** : partie de la grammaire décrivant les règles par lesquelles se combinent en phrases les unités significatives (mots).

**transitif** : un verbe transitif implique la présence d'un syntagme nominal\* pour le compléter. Par exemple, le verbe *aimer* (*quelqu'un*) est transitif.

**temps grammatical** : Marques linguistiques utilisées pour pour exprimer les notions dans le langage (imparfait, présent de l'indicatif, futur antérieur, etc. . . )

# BIBLIOGRAPHIE

- [Abeillé and Clément, 2003] Anne Abeillé and Lionel Clément. Annotation morpho-syntaxique, les mots simples, les mots composés. <http://www.llf.cnrs.fr/fr/Abeille/>, janvier 2003.
- [Abeillé, 1993] Anne Abeillé. *Les nouvelles syntaxes : grammaires d'unification et analyse du Français*. Collection Linguistique. Armand Colin, 1993.
- [Abney, 1996] Steven Abney. *Corpus-Based Methods in Language and Speech*, chapter Part-of-Speech Tagging and Partial Parsing. Kluwer Academic Publisher, 1996.
- [Abney, 1997] Steven Abney. *The SCOL Manual*, April 1997. <http://www.vinartus.net/spa/>.
- [Allen, 1983] James Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, pages 832–843, 1983.
- [Allen, 1994] James Allen. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc, 1994.
- [Berline, 2002] Chantal Berline. Cours de lambda-calcul, 2002. <http://www.pps.jussieu.fr/berline/Cours.html>.
- [Bras, 1990] Myriam Bras. *Calcul des Structures Temporelles du Discours*. PhD thesis, IRIT, 1990.
- [Casabianca, 2002] Michel Casabianca. Introduction au langage xml, 2002. <http://www.sdv.fr/pages/casa/html/intro-xml.html>.
- [Davis, 1991] Steven Davis, editor. *Pragmatics, a Reader*. Oxford University Press, 1991.
- [de Saussure, 1916] Ferdinand de Saussure. *Cours de Linguistique Générale*. Payot, 1916.
- [Dubois, 1994] Jean Dubois, editor. *Dictionnaire de linguistique et des sciences du langage*. Trésors du Français. Larousse, 1994.
- [Grover *et al.*, 1994] Claire Grover, Janet Hitzeman, and Marc Moens. The implementation of the temporal portion of the discourse grammar. Technical report, University of Edinburgh, Décembre 1994. LRE 61-062.
- [Grover *et al.*, 1995] Claire Grover, Janet Hitzeman, and Marc Moens. Algorithms for analysing the temporal structure of discourse. In *Sixth International Conference of the European Chapter of the Association for Computational Linguistics*. ACL, 1995.
- [Harper *et al.*, 2001] Lisa Harper, Inderjeet Mani, and Beth Sundheim, editors. *Temporal and Spatial Information Processing*, 39<sup>th</sup> Annual Meeting and 10<sup>th</sup> Conference of the European Chapter. Association for Computational Linguistics, Morgan Kaufman Publishers, Juillet 2001.

- [Katz and Arosio, 2001] Graham Katz and Fabrizio Arosio. The annotation of temporal information in natural language sentences. In Harper et al. [2001], pages 104–111.
- [Kreen and Samuelsson, 1997] Brigitte Kreen and Christer Samuelsson. The linguist’s guide to statistics, mai 1997. Don’t Panic.
- [Levinson, 1983] Stephen C. Levinson. *Pragmatics*. Cambridge University Press, 1983.
- [Marcu and Echihiabi, 2002] Daniel Marcu and Abdessamad Echihiabi. An unsupervised approach to recognizing discourse relations. In *40th Annual Meeting of the Association for Computational Linguistics*, Juillet 2002.
- [Pustejovsky et al., 2002] James Pustejovsky, Roser Saurí, Andrea Setzer, Rob Gaizauskas, and Bob Ingria. *TimeML Annotation Guidelines*. TERQAS Annotation Working Group, july 2002.
- [Schmid, 1994] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, September 1994.
- [Setzer, 2001] Andrea Setzer. *Temporal Information in Newswire Articles: an Annotation Scheme and Corpus Study*. PhD thesis, University of Sheffield, UK, September 2001.
- [Shilder and Habel, 2001] Franck Shilder and Christopher Habel. From temporal expressions to temporal information: Semantic tagging of news messages. In Harper et al. [2001], pages 65–72.
- [Steedman, 1997] Mark Steedman. Temporality. In J. Van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*. Elsevier Science B.V., 1997.
- [Vendler, 1967] Z. Vendler. *Linguistics in Philosophy*. Cornell University Press, 1967.
- [Vet, 1994] Co Vet. Petite grammaire de l’aktionsart et de l’aspect. *Les cahiers de grammaire*, 19, décembre 1994.

# Annexes



# ANNEXE A

## Quelques outils utilisés

### Les expressions régulières

Une expression régulière est représentée par une suite de symboles standardisés qui expriment d'une manière formelle la structure d'une chaîne de caractères.

*Exemples :*

- ★ `[aeiouyAEIOUY]` : toutes les voyelles
- ★ `[A-Z]\w\b` : tous les mots commençant par une majuscule.
- ★ `[A-Z]\w\b (\w*\b)* \.` : toutes les phrases (premier mot commençant par une majuscule, puis 0 ou plusieurs mots, puis un point).
- ★ `\w+ ( er | ir | dre ) \b` : tous les mots finissant par *er*, *ir* ou *dre* (approximativement, les verbes).
- ★ `"\."` : toutes les citations (caractères entre guillemets)

Table A.1: Quantificateurs reconnus

Quantificateur	Utilité
<code>[]</code>	fait référence à l'un ou l'autre des caractères compris entre les crochets
<code>[^ ]</code>	fait référence à tous les caractères sauf à ceux qui sont compris entre les crochets, après le <code>^</code>
<code>( )</code>	regroupe une séquence de symboles
<code>(   )</code>	fait référence à l'une ou l'autre des séquences (séparées par la barre verticale) qui sont comprises entre les parenthèses
<code>+</code>	ce qui précède le <code>+</code> est répété une fois ou plus
<code>?</code>	ce qui précède le <code>?</code> est optionel
<code>*</code>	ce qui précède le <code>*</code> est optionel mais peut aussi être répété une fois ou plus
<code>{n,}</code>	ce qui précède <code>{n,}</code> est répété n fois ou plus
<code>{n,m}</code>	ce qui précède <code>{n,m}</code> est répété entre n et m fois
<code>{,m}</code>	ce qui précède <code>{,m}</code> peut être répété jusqu'à m fois

Table A.2: Symboles reconnus

Symbole	Utilité
<code>^</code>	début du texte
<code>\$</code>	fin du texte
<code>.</code>	n'importe quel caractère
<code>\d</code>	les chiffres ; équivalent à <code>[0-9]</code>
<code>\D</code>	tout sauf les chiffres ; équivalent à <code>[^0-9]</code>
<code>\w</code>	les caractères de mot ; équivalent à <code>[a-zA-Z0-9_]</code>
<code>\W</code>	tout sauf les caractères de mots ; équivalent à <code>[^a-zA-Z0-9_]</code>
<code>\b</code>	les séparateurs de mot
<code>\B</code>	tout sauf les séparateurs de mot
<code>\n</code>	une nouvelle ligne
<code>\r</code>	un retour de chariot
<code>\t</code>	une tabulation (horizontale)
<code>\v</code>	une tabulation verticale
<code>\f</code>	un saut de page
<code>\s</code>	les caractères d'espacement; équivalent à <code>[\n\r\t\v\f]</code>
<code>\S</code>	tout sauf les caractères d'espacement; équivalent à <code>[\n\r\t\v\f]</code>
<code>\</code>	un <code>\</code> suivi d'un caractère annule la proprit� spéciale de celui-ci, soit ici. Par exemple: <code>\.</code> , <code>\\</code> , <code>\+</code> , <code>\ </code> , <code>\(</code> , <code>\)</code> , <code>\[</code> , <code>\]</code> , etc...

## XML

XML (eXtensible Markup Language) est né du constat que le HTML n'est pas adapté aux futures exigences de l'Internet. Depuis, XML s'impose non seulement dans les technologies du Web, mais également dans de nombreux domaines qui impliquent le stockage structuré de données.

Contrairement à HTML, qui proposent des balises prédéfinies (essentiellement dédiées au formatage des données), XML est un langage de balises définies par l'utilisateur (voir la figure A.1). Cela permet de dissocier le fond et la forme des documents ainsi traités. De plus, ce langage est extrêmement facile à apprendre ([Casabianca, 2002]).

### XML et Java

Deux API de Java permettent de gérer le XML de deux façons différentes :

SAX offre une analyse événementielle du XML. Il parcourt linéairement le document et génère un événement à chaque élément rencontré (ouverture et fermeture de balise, texte, etc...). Cette méthode est peu gourmande en mémoire, mais n'offre pas de vision globale du document (une fois un événement traité, il est impossible de revenir en arrière).

DOM construit une représentation arborescente globale du document XML, une balise étant un nœud pouvant contenir d'autres nœuds. Cette représentation, illustrée à la figure A.1, permet une utilisation globale et récursive de la structure du document XML, mais elle utilise plus de mémoire, puisque tout le document est analysé en même temps.

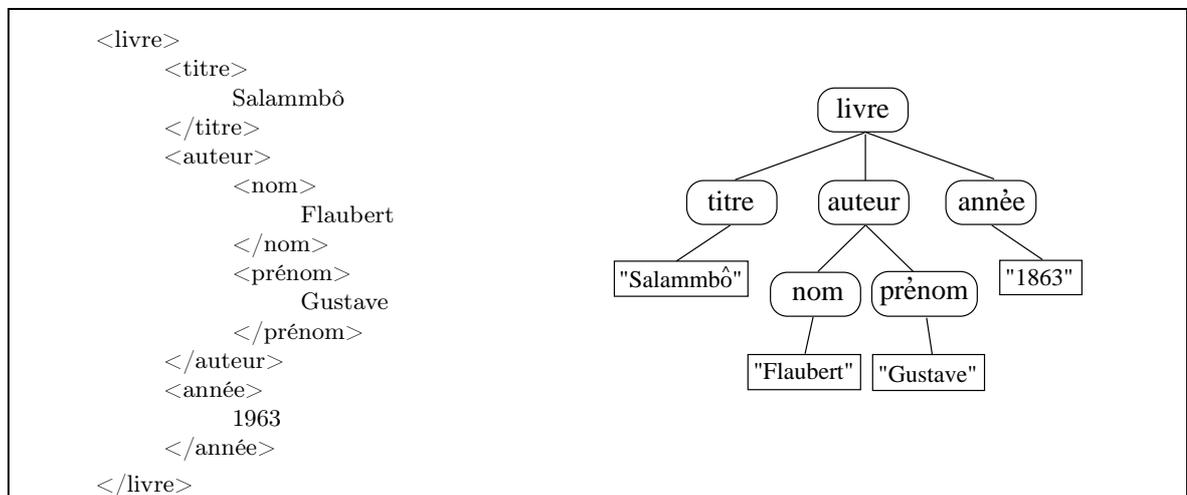


FIG. A.1: Structure arborescente de XML utilisée par l'API DOM



## ANNEXE B

# Analyse syntaxique : dictionnaire des données

### Éléments à la sortie de TreeTagger après modifications

<b>a</b> : le mot <i>a</i>	<b>debut</b> : le mot <i>début</i> .
<b>advspec</b> : adverbe de spécification ( <i>environ, exactement, ...</i> ).	<b>dta</b> : article.
<b>advtemp</b> : tous les adverbiaux temporels* reconnus.	<b>en</b> : le mot <i>en</i> .
<b>alors</b>	<b>en_ce_moment</b> : tous les synonymes de <i>en ce moment</i> .
<b>annee(s)</b> : le mot <i>année(s)</i>	<b>enfin</b>
<b>a_partir_de</b> : tous les synonymes de <i>à partir de</i> .	<b>et</b> : le mot <i>et</i> .
<b>après</b> : tous les synonymes de <i>après</i> .	<b>fin</b> : le mot <i>fin</i> .
<b>au_debut</b> : tous les synonymes de <i>au début</i> .	<b>fn</b> : prénom.
<b>aujourd_hui</b>	<b>heure(s)</b> : le mot <i>heure(s)</i>
<b>aux_imp</b> : un auxiliaire à l'imparfait.	<b>i</b> : une initiale.
<b>aux_pres</b> : un auxiliaire au présent.	<b>in</b> : préposition.
<b>aux_fut</b> : un auxiliaire au futur.	<b>jamais</b>
<b>aux_ps</b> : un auxiliaire au passé simple.	<b>jj?</b> : toutes sortes d'adjectifs.
<b>avant</b> : tous les synonymes de <i>avant</i> .	<b>jour(s)</b> : le mot <i>jour(s)</i>
<b>card</b> : cardinal, en lettres ou en chiffres.	<b>matin</b>
<b>day</b> : un jour de la semaine.	<b>md</b> : verbe modal.
<b>de</b> : le mot <i>de</i>	<b>minute(s)</b> : le mot <i>minute(s)</i>
<b>décennie(s)</b> : le mot <i>décennie(s)</i>	<b>modift</b> : modificateur temporel ( <i>plus de, moins de...</i> )
	<b>month</b> : un mot représentant un mois.

**name** : nom de personne.

**neg** : le mot *pas*.

**nuit**

**nn** : nom commun.

**nnp** : nom propre.

**onze**

**pas**

**pays** : un nom de pays.

**pendant** : tous les synonymes de *pendant*.

**peu**

**plus**

**plusieurs** : tous les synonymes de *plusieurs*.

**plus\_tard** : tous les synonymes de *plus tard*.

**plus\_tot** : tous les synonymes de *plus tôt*.

**pour\_1\_heure** : tous les synonymes de *maintenant*.

**puis**

**quelque(s)** : tous les synonymes de *quelque(s)*.

**pun\_cit** : guillemet.

**rb** : tous les adverbes qui ne sont pas temporellement spécifiques.

**rbr** : comparatif.

**rbs** : superlatif.

**relst** : modificateur temporel relatif au moment d'élocution (*dernier, prochain...*).

**reltf** : modificateur temporel relatif au focus temporel\* (*précédent, suivant...*).

**rien**

**saison** : un mot représentant une saison.

**se** : le mot *se*

**seconde(s)** : le mot *seconde(s)*

**semaine(s)** : le mot *semaine(s)*

**siecle(s)** : le mot *siècle(s)*

**soir**

**soudain** : tous les synonymes de *soudain*.

**sur** : le mot *sur*

**toujours**

**tout**

**tt** : identificateur de personne (Monsieur, Mme, etc...)

**un** : le mot *un*.

**ver\_cond** : verbe au conditionnel.

**ver\_fut** : verbe au futur.

**ver\_inf** : verbe à l'infinitif.

**ver\_imp** : verbe à l'imparfait.

**ver\_imper** : verbe à l'impératif.

**ver\_ppas** : participe passé.

**ver\_ppre** : participe présent.

**ver\_pres** : verbe au présent.

**ver\_ps** : verbe au passé simple.

**ville** : un nom de ville.

**virg** : virgule.

## Éléments générés par nos règles

**atome** : atome non temporel.

**ax** : groupe adjectival.

**ax\_sup** : groupe superlatif.

**c** : clause complète.

**ci** : clause interrogative complète.

**c0** : clause 1.

<b>c1</b> : clause 2.	<b>ng_verb(c)</b> : syntagme nominal formé avec un infinitif.
<b>ccd</b> : adverbial non spécifique.	<b>np</b> : syntagme nominal 2.
<b>ccl</b> : adverbial locatif.	<b>nx</b> : syntagme nominal 1.
<b>cct</b> : adverbial temporel.	<b>nombre</b> : un dérivé de <b>card</b> , incluant les nombres composés ( <i>soixante et un, ...</i> ).
<b>ci0</b> : clause interrogative 1.	<b>partpresent</b> : verbe au participe présent.
<b>citation</b> : citation.	<b>partpasse</b> : verbe au participe passé.
<b>clause*</b> : clauses diverses.	<b>person</b> : nom de personne.
<b>date(1/2)</b> : date non absolue.	<b>rx</b> : succession d'adverbes
<b>dateabs</b> : date absolue.	<b>subc0</b> : proposition subordonnée relative.
<b>datemix</b> : date ou durée contenue dans un NP ( <i>une heure de vol</i> ).	<b>subc0v</b> : proposition subordonnée complétive.
<b>datenom</b> : date suivie par un NP ( <i>trois jours après le vol</i> ).	<b>subc</b> : proposition subordonnée adverbiale.
<b>daterelST(1/2)</b> : date relative au moment d'élocution.	<b>subc1</b> : proposition incise.
<b>daterelTF(1/2/3)</b> : date relative au focus temporel*.	<b>subc2</b> : autres propositions subordonnées.
<b>datespecabs</b> : date absolue, de forme particulière.	<b>tatome</b> : atome de temps ( <i>trois jours, 4 ans, ...</i> ).
<b>datespecrel</b> : date relative, de forme particulière.	<b>time</b> : l'indication d'une heure.
<b>duree</b> : durée quelconque.	<b>titre</b> : titre
<b>duree(2/3)</b> : durée contenant deux dates ( <i>du 11 février au 27 octobre...</i> ).	<b>ver_fut_pas</b> : verbe au futur, voix passive.
<b>dureerelabs</b> : durée absolue.	<b>ver_imp_pas</b> : verbe à l'imparfait, voix passive.
<b>dureerelST</b> : durée relative au moment d'élocution.	<b>ver_pc</b> : verbe au passé composé.
<b>dureerelTF</b> : durée relative au focus temporel*.	<b>ver_pc_pas</b> : verbe au passé composé, voix passive.
<b>ind_sub</b> : indicateur spécifique de subordonnée (pour ... duquel...)	<b>ver_pqp</b> : verbe au plus-que-parfait.
<b>un_lieu</b> : lieu.	<b>ver_pqp_pas</b> : verbe au plus-que-parfait, voix passive.
<b>ng</b> : syntagme nominal 3.	<b>ver_ps_pas</b> : verbe au passé simple, voix passive.
<b>ng2</b> : syntagme nominal 4.	<b>vx</b> : verbe.
<b>ng_sup</b> : syntagme nominal avec superlatif.	<b>vxi</b> : verbe, forme interrogative.
	<b>vx_neg</b> : verbe négatif.
	<b>vx_pos</b> : verbe positif.



## ANNEXE C

# Propagation de contraintes d'Allen

La représentation de notre structure temporelle du discours est un graphe orienté. Les nœuds de ce graphe sont les événements du texte. Les arcs représentent les relations d'Allen possibles entre les nœuds. Si un arc existe entre les événements  $e1$  et  $e2$ , l'arc contenant les relations "inverses" au premier en est déduit entre  $e2$  et  $e1$ . Par exemple :

$$\forall i \forall j ((i \text{ b } j) \vee (i \text{ m } j)) \Leftrightarrow ((j \text{ bi } i) \vee (j \text{ mi } i))$$

C'est en tous cas ce que l'on considère dans l'algorithme qui suit. En pratique, pour économiser de la mémoire, on ne notera qu'un seul des deux arcs et on fera attention de ne pas oublier des éléments lors du choix des relations à propager.

*N.B.* : Par souci de simplicité, ce que l'on appellera par la suite une relation devra être compris comme une disjonction de relations d'Allen.

Pour compléter à chaque étape les relations obtenues par notre traitement, il est nécessaire de saturer ce graphe, c'est-à-dire de propager toutes les contraintes à travers les arcs par transitivité. Voici l'algorithme appliqué :

**Soient**  $\left\{ \begin{array}{l} A = \text{l'ensemble des arcs du graphe} \\ N = \text{l'ensemble des nœuds du graphe} \\ U = \text{la disjonction des treize relations d'Allen} \end{array} \right.$

1. *modifié* = 0
2. Pour chaque couple d'arcs  $(i, j)$  tel que :

$$\left\{ \begin{array}{l} \bullet i \in N, j \in N \\ \bullet \exists k \in N ((i, k) \in A \wedge (k, j) \in A) \\ \bullet \text{la relation entre deux nœuds } m \text{ et } n \text{ est notée } R_{m,n} \end{array} \right.$$

- (a)  $R1_{i,j} = \text{compose}(R_{i,k}, R_{k,j})$  (voir plus loin)
- (b) Si aucun arc (une relation  $R2_{i,j}$ ) n'existait déjà entre  $i$  et  $j$ ,  
alors  $R2_{i,j} = U$
- (c) faire l'intersection :  $R_{i,j} = R1_{i,j} \cap R2_{i,j}$
- (d) Si  $R_{i,j} = \emptyset$  (inconsistance, en théorie impossible, mais à ne pas propager)  
alors : afficher un message d'erreur  
 $R_{i,j} = R2_{i,j}$  (retour)
- (e) Si  $R_{i,j} = U$   
alors ne rien faire.  
(un arc comportant toutes les relations n'apporte aucune information)

sinon mettre l'arc à jour  
 $modifié = 1$

3. Si  $modifié = 1$ , recommencer en 1.

La fonction  $compose(R_{i,k}, R_{k,j})$  utilise les 169 relations de transitivité entre les relations d'Allen (voir section 2.1.4). Pour chaque couple  $(r_1, r_2)$  de relations binaires d'Allen tel que  $r_1 \in R_{i,k}$  et  $r_2 \in R_{k,j}$ , on applique la transition adaptée. Le résultat est la disjonction de toutes les relations ainsi obtenues.

Cet algorithme est appliqué en boucle jusqu'à ce qu'une itération ne change pas la structure du graphe ( $modifié = 0$ ). La propagation est alors terminée.

## ANNEXE D

# Liste des relations induites par les temps grammaticaux

S1	S2	Relation	Exemple
PS / PC	PS / PC	before after overlaps is_overlaped includes is_included	Marie poussa Pierre. Il tomba. ∅ Toute la classe rigola. Il rougit. ∅ J'assemblai mon bureau moi-même. Les tiroirs ne me prirent que dix minutes. ∅
IMP	PS / PC	before after overlaps is_overlaped includes is_included	∅ ∅ Marie était furieuse. Elle poussa Pierre. ∅ Jean n'était pas marié. Il fréquenta plusieurs femmes... ∅
PQP	PS / PC	before after overlaps is_overlaped includes is_included	Sam était arrivé à la maison. Il frappa la porte. ∅ Gringoire [...] était resté sur le pavé [...]. Il reprit ses sens. <i>Notre-Dame de Paris</i> ∅ ∅ ∅
PS / PC	IMP	before after overlaps is_overlaped includes is_included	Louis XI fit Paris asile en 1467. Le criminel y était sacré. ∅ Ils construisirent une cathédrale. C'était un bel édifice. Marie poussa Jean. Elle était en colère. Marie courut à travers les docks. C'était une course rapide et sans but. ∅

S1	S2	Relation	Exemple
IMP	IMP	before after overlaps  is_overlaped includes is_included	$\emptyset$ $\emptyset$ Franck était content. Il y avait une pizza en face de lui. $\emptyset$ $\emptyset$ $\emptyset$
PQP	IMP	before  after overlaps is_overlaped includes  is_included	Le taxi avait déposé Jean chez lui. Il était soulagé. $\emptyset$ L'avion avait atterri. Jean était à Boston. $\emptyset$ Il avait construit une cathédrale. C'était un travail difficile. $\emptyset$
PS / PC	PQP	before after overlaps is_overlaped includes is_included	$\emptyset$ Sam arriva chez lui. Il avait perdu ses clés. $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$
IMP	PQP	before after overlaps is_overlaped includes is_included	$\emptyset$ Jean était furieux. Julie s'était éloignée de lui. Julie était furieuse. Elle s'était éloignée de Jean. $\emptyset$ $\emptyset$ $\emptyset$
PQP	PQP	before after  overlaps  is_overlaped includes  is_included	Marie avait poussé Pierre. Il était tombé. Jean était parti vers Boston. Marie lui en avait donné l'idée. Marie avait regardé Pierre. Il lui avait rendu sa part de pizza. Pierre avait failli perdre le jeu. Il avait eu chaud ! Jean avait traversé la Sibérie. Il était parti de Minsk. $\emptyset$
Temps du passé <sup>1</sup>	Temps du présent <sup>2</sup>	before after overlaps is_overlaped includes is_included	Marie a poussé Pierre. Il tombe. $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$

<sup>1</sup>Imparfait, Passé Simple, Passé Composé, Plus-que-Parfait, Passé Antérieur, Futur Antérieur, Subjonctif Imparfait, Subjonctif Plus-que-Parfait, Conditionnel Passé 1 et 2

<sup>2</sup>Présents de l'indicatif, du subjonctif et du conditionnel

S1	S2	Relation	Exemple
Règle valable si aucune autre règle plus spécifique n'existe :			
Temps du passé <sup>1</sup>	Temps du passé <sup>1</sup>	before after overlaps is_overlaped includes is_included	OK $\emptyset$ OK $\emptyset$ OK $\emptyset$
Temps du passé <sup>1</sup>	Futur	before after overlaps is_overlaped includes is_included	Pierre a escaladé la falaise. Il recommencera. $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$
Temps du présent <sup>2</sup> / Temps du passé <sup>1</sup> : voir relation inverse			
Temps du présent <sup>2</sup>	Temps du présent <sup>2</sup>	before after overlaps is_overlaped includes is_included	Marie pousse Pierre. Il tombe. $\emptyset$ Marie regarde Pierre. Il lui rend sa part de pizza. $\emptyset$ Pierre escalade la falaise. Il entame un passage difficile. $\emptyset$
Temps du présent <sup>2</sup>	Futur	before after overlaps is_overlaped includes is_included	Pierre escalade la falaise. Il arrivera au sommet avant la nuit. $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$
Futur / Temps du présent <sup>2</sup> ou du passé <sup>1</sup> : voir relations inverses			



## ANNEXE E

# Guide de l'annotateur

Le guide fourni à l'annotateur est reproduit dans les deux pages qui suivent. C'est le seul outil qu'il a à sa disposition, en plus bien sûr du texte brut qu'il doit annoter.

# Guide pour l'annotation temporelle

Le seul élément fourni au sujet est le texte brut à annoter. L'annotation consiste à collecter les éléments suivants :

- Les événements (**verbes conjugués, participes passés n'ayant pas une valeur d'adjectif, participes présents**) qu'il reconnaît dans le texte. En pratique, on attribuera un numéro à chaque événement et on donnera l'infinitif de ce verbe. Les numéros seront attribués dans l'ordre de la lecture.
- Les dates auxquelles ont lieu les événements, *lorsqu'il est possible de le déterminer*. Il n'est pas nécessaire de calculer la date précise, une information comme "jeudi soir" suffira.
- Les relations qu'il juge pertinentes entre **deux événements**, ceci comme dans l'exemple qui suit. La liste des relations possibles est donnée à la page suivante.

## Exemple

Le 2 décembre 1804, Napoléon Bonaparte devint empereur des Français. Un an plus tard exactement, il gagna la bataille d'Austerlitz, que l'on surnommera *Bataille des trois empereurs*. Entre-temps, il avait été vaincu à Trafalgar, durant laquelle son ennemi Nelson avait trouvé la mort.

Un exemple d'annotation sur ce texte est la liste encadrée suivante :

```
--- EVENTS ---
1 : devenir
2 : gagner
3 : surnommer
4 : vaincre
5 : trouver

--- DATES ---
1 => le 2 décembre 1804
2 => le 2 décembre 1805

--- RELATIONS ---
1 2  avant
2 3  avant
4 2  avant
1 4  avant
5 4  pendant
```

*N.B. : Le texte qui suit est une illustration destinée à mieux faire comprendre l'exemple. L'annotateur ne doit pas fournir ce texte.*

Le 2 décembre 1804<sub>1</sub>, Napoléon Bonaparte devint<sub>1</sub> empereur des Français.

Un an plus tard exactement<sub>2</sub>, il gagna<sub>2</sub> la bataille d'Austerlitz, que l'on surnommera<sub>3</sub> *Bataille des trois empereurs*.

Entre-temps, il avait été vaincu<sub>4</sub> à Trafalgar, bataille durant laquelle son ennemi Nelson avait trouvé<sub>5</sub> la mort.

La liste des relations possibles est fournie sur la page suivante.

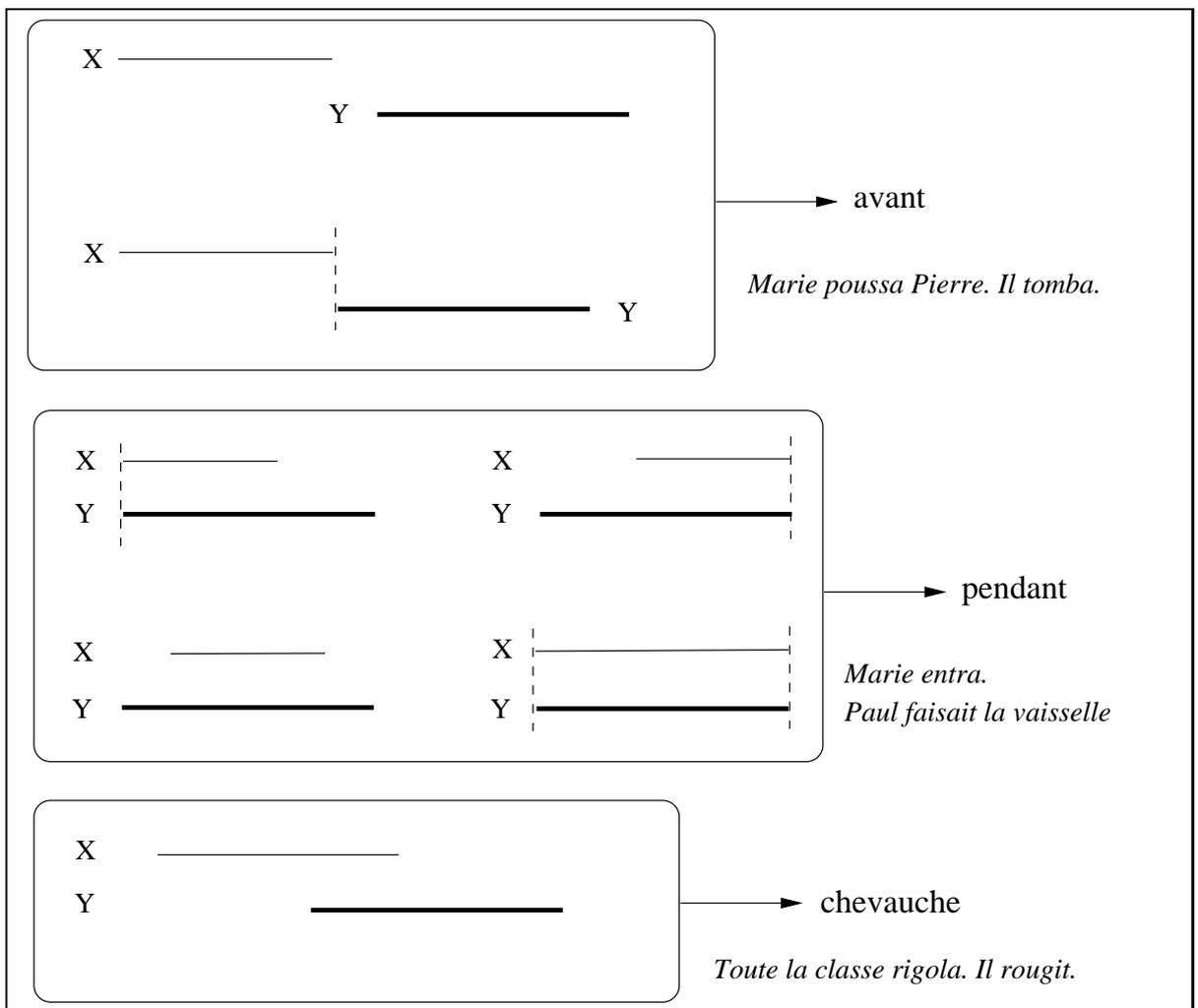


FIG. E.1: Les relations possibles entre les événements



# ANNEXE F

## Textes étudiés

### Les textes bruts

#### *Texte 1*

*Extrait d'une dépêche A.F.P. du 9 juillet 2003*

#### **Bush entame l'étape sud-africaine de sa tournée**

PRETORIA

Le président américain George W. Bush entame mercredi à Pretoria la deuxième étape de sa tournée africaine, par des entretiens avec le président sud-africain Thabo Mbeki, qui devraient porter sur les crises africaines, dont le Zimbabwe, l'aide, le commerce et le terrorisme.

M. Mbeki a officiellement accueilli mercredi matin George W. Bush à Pretoria. Après une brève cérémonie ponctuée des deux hymnes nationaux, les deux chefs d'Etat se sont ensuite dirigés immédiatement vers la résidence de la présidence sud-africaine, pour un entretien en tête-à-tête qui devait durer environ une heure.

Le président Bush, accompagné de son épouse Laura, est arrivé tard mardi soir à la base de Waterkloof, près de Pretoria, en provenance du Sénégal où il a débuté mardi une tournée africaine de cinq pays, sa première sur le continent, qui doit aussi le conduire au Botswana, en Ouganda et au Nigeria. Le président américain a été accueilli par la ministre sud-africaine des Affaires étrangères, Mme Nkosazana Dlamini-Zuma.

MM. Bush et Mbeki devraient ensuite déjeuner ensemble, occasion de nouvelles discussions dans un cadre "plus informel", selon Pretoria. Les conflits du continent africain, les négociations de l'Organisation mondiale du Commerce, l'allègement de la dette, l'aide à la stratégie africaine de développement NEPAD, ainsi que la lutte contre le terrorisme, devraient être discutés par les deux chefs d'Etat, selon des officiels sud-africains.

Parmi les crises africaines, le Zimbabwe devrait notamment figurer. Ces derniers jours, Colin Powell puis George W. Bush ont appelé sans détour Thabo Mbeki à "un rôle plus important" et à "insister" davantage pour la démocratie et la bonne gouvernance à Harare.

Plusieurs rues de Pretoria seront fermées pendant plusieurs jours, dans le cadre de ce que des officiels sud-africains ont qualifié "d'opération logistique massive" de sécurité pour la venue du président américain.

Plusieurs organisations, dont le Congrès National Africain (ANC), la centrale syndicale COSATU, le Parti communiste et une coalition pacifiste, ont prévu une manifestation mercredi à Pretoria pour protester contre la politique étrangère américaine en Irak, en Palestine et

contre Cuba.

Selon son programme disponible mardi, George W. Bush devait ensuite mercredi rencontrer, lors d'un dîner aux côtés de l'ambassadeur américain Cameron Hume, des industriels et hommes d'affaires sud-africains et américains. Jeudi matin, il quitte Pretoria pour une visite d'une journée au Botswana, avec un retour prévu jeudi soir, et un départ vendredi matin pour l'Ouganda.

*A.F.P.*

---

## *Texte 2*

*Extrait d'une dépêche A.F.P. du 30 mai 2003*

### **Séisme : les algérois dans la rue dans la crainte d'une forte réplique**

ALGER

Les Algérois étaient nombreux jeudi soir dans la rue et les jardins publics de la capitale algérienne dans la crainte d'une forte réplique sismique, comme celle qui les a brutalement réveillés la nuit dernière.

Secoués par une violente secousse d'une magnitude de 5,8 sur l'échelle de Richter jeudi à 03 h 15 (02 h 15 GMT), les Algérois sont descendus nombreux jeudi soir dans les rues d'Alger, a constaté l'AFP.

Cette réplique a été l'une des plus puissantes enregistrée dans Alger et sa région depuis le séisme d'une magnitude de 6,8 sur l'échelle de Richter qui a fait, le 21 mai, plus de 2.200 morts et quelque 10.000 blessés.

Mardi soir, la capitale avait été aussi secouée par une réplique de la même magnitude créant un fort mouvement de panique.

Au pied du palais du gouvernement et dans le jardin public proche des familles se sont installées pour la nuit.

Des voitures sont garées non loin avec des femmes et des enfants enroulés dans des couvertures baillant de sommeil à l'approche de minuit.

*A.F.P.*

---

## *Texte 3*

*Extrait d'une dépêche A.F.P. du 2 juin 2003*

### **Le représentant de l'Onu attendu, une administration dans 6 semaines**

BAGDAD

Le représentant de l'Onu pour l'Irak, Sergio Vieira de Mello, est attendu lundi soir à Bagdad alors que les nouveaux leaders politiques irakiens se concertent sur le projet de la coalition américano-britannique d'installer une administration intérimaire dans six semaines.

Dans la capitale, des centaines de militaires irakiens ont à nouveau manifesté pour protester contre la dissolution de l'ensemble de leurs unités que les Etats-Unis ont décidé de remplacer par une nouvelle armée de 40.000 à 50.000 hommes.

Les forces d'occupation ont annoncé par ailleurs leur intention de libérer prochainement la

plupart des prisonniers irakiens, à l'exception de 300 à 500 hauts cadres de l'armée, en raison de leur implication présumée dans des crimes contre leur peuple.

Dimanche, à l'issue d'une réunion tenue par l'administrateur américain Paul Bremer avec le Conseil de sept dirigeants de l'ex-opposition irakienne, la coalition a annoncé la mise en place dans un délai de six semaines d'une administration intérimaire irakienne, dirigée par un Conseil politique.

Composé de 25 à 30 membres, le Conseil politique aura pour tâche principale de faire des recommandations à l'administration occupante dans les domaines économique et politique. Il devra également nommer des conseillers dans les différents ministères, qui auront ensuite un statut de ministres intérimaires.

Ce Conseil, que la coalition compte mettre en place après des consultations élargies aux différents groupes irakiens, devra aussi préparer un référendum sur une nouvelle Constitution.

*A.F.P.*

# La sortie du système de calcul des relations temporelles

## Texte 1

#----- DATES -----

# 3 : ([2003, 7, 16, 0, 0, 0, 0], [2003, 7, 16, 23, 59, 59, 999])  
# 6 : ([2003, 7, 2, 5, 0, 0, 0], [2003, 7, 2, 11, 59, 59, 999])  
# 12 : ([2003, 7, 8, 0, 0, 0, 0], [2003, 7, 8, 23, 59, 59, 999])  
# 23 : ([2003, 7, 2, 0, 0, 0, 0], [2003, 7, 2, 23, 59, 59, 999])  
# 26 : ([2003, 7, 2, 0, 0, 0, 0], [2003, 7, 2, 23, 59, 59, 999])  
# 29 : ([2003, 7, 10, 5, 0, 0, 0], [2003, 7, 10, 11, 59, 59, 999])  
# 30 : ([2003, 7, 11, 5, 0, 0, 0], [2003, 7, 11, 11, 59, 59, 999])  
# 0 : ([2003, 7, 9, 0, 0, 0, 0], [2003, 7, 9, 23, 59, 59, 999])

#----- EVENTS -----

2 : entamer : ver\_pres  
4 : devoir porter : ver\_cond  
5 : avoir officiellement accueillir : ver\_pc  
7 : se être ensuite diriger : ver\_pc  
8 : devoir durer : ver\_imp  
9 : être arriver : ver\_pc  
10 : conduire : ver\_inf  
11 : avoir débiter : ver\_pc  
13 : devoir aussi : ver\_pres  
14 : avoir être accueillir par : ver\_pc\_pas  
15 : devoir ensuite déjeuner : ver\_cond  
16 : devoir être discuter : ver\_cond  
17 : devoir notamment figurer : ver\_cond  
18 : avoir appeler : ver\_pc  
19 : insister : ver\_inf  
20 : être fermer : ver\_fut\_pas  
21 : avoir qualifier : ver\_pc  
22 : avoir prévoir : ver\_pc  
24 : protester : ver\_inf  
25 : devoir ensuite : ver\_imp  
27 : rencontrer : ver\_inf  
28 : quitter : ver\_pres  
1 : begin : ver\_pres

#----- RELATIONS -----

1, 2 : [trel\_before]  
1, 4 : [trel\_beforei, trel\_during, trel\_duringi, trel\_duringi, trel\_equals, trel\_finishesi, trel\_meetsi, trel\_overlapsi, trel\_startsi]  
1, 5 : [trel\_beforei]  
1, 7 : [trel\_beforei]  
1, 8 : [trel\_beforei]  
1, 9 : [trel\_beforei]  
1, 11 : [trel\_beforei]  
1, 13 : [trel\_before]  
1, 14 : [trel\_beforei]  
1, 15 : [trel\_beforei, trel\_during, trel\_duringi, trel\_duringi, trel\_equals, trel\_finishesi, trel\_meetsi, trel\_overlapsi, trel\_startsi]  
1, 16 : [trel\_beforei, trel\_during, trel\_duringi, trel\_duringi, trel\_equals, trel\_finishesi, trel\_meetsi, trel\_overlapsi, trel\_startsi]  
1, 17 : [trel\_beforei, trel\_during, trel\_duringi, trel\_duringi, trel\_equals, trel\_finishesi, trel\_meetsi, trel\_overlapsi, trel\_startsi]  
1, 18 : [trel\_beforei]  
1, 21 : [trel\_beforei]  
1, 22 : [trel\_beforei]  
1, 25 : [trel\_beforei]  
1, 28 : [trel\_before]  
2, 4 : [trel\_beforei, trel\_during, trel\_meetsi, trel\_overlapsi]  
2, 5 : [trel\_beforei]  
2, 7 : [trel\_beforei]

2, 8 : [trel\_beforei]  
 2, 9 : [trel\_beforei]  
 2, 11 : [trel\_beforei]  
 2, 13 : [trel\_before]  
 2, 14 : [trel\_beforei]  
 2, 15 : [trel\_beforei, trel\_during, trel\_meetsi, trel\_overlapsi]  
 2, 16 : [trel\_beforei, trel\_during, trel\_meetsi, trel\_overlapsi]  
 2, 17 : [trel\_beforei, trel\_during, trel\_meetsi, trel\_overlapsi]  
 2, 18 : [trel\_beforei]  
 2, 21 : [trel\_beforei]  
 2, 22 : [trel\_beforei]  
 2, 25 : [trel\_beforei]  
 2, 28 : [trel\_beforei]  
 4, 5 : [trel\_beforei]  
 4, 7 : [trel\_beforei]  
 4, 8 : [trel\_beforei]  
 4, 9 : [trel\_beforei]  
 4, 11 : [trel\_beforei]  
 4, 13 : [trel\_before, trel\_duringi, trel\_meets, trel\_overlaps]  
 4, 14 : [trel\_beforei, trel\_meetsi]  
 4, 18 : [trel\_beforei]  
 4, 21 : [trel\_beforei]  
 4, 22 : [trel\_beforei]  
 4, 25 : [trel\_beforei]  
 4, 28 : [trel\_before, trel\_duringi, trel\_meets, trel\_overlaps]  
 5, 11 : [trel\_before]  
 5, 13 : [trel\_before]  
 5, 15 : [trel\_before]  
 5, 16 : [trel\_before]  
 5, 17 : [trel\_before]  
 5, 20 : [trel\_before]  
 5, 28 : [trel\_before]  
 7, 13 : [trel\_before]  
 7, 15 : [trel\_before, trel\_meets]  
 7, 16 : [trel\_before, trel\_meets]  
 7, 17 : [trel\_before, trel\_meets]  
 7, 20 : [trel\_before, trel\_meets]  
 7, 28 : [trel\_before]  
 8, 11 : [trel\_before, trel\_duringi, trel\_finishesi, trel\_meets, trel\_overlaps]  
 8, 13 : [trel\_before]  
 8, 15 : [trel\_before, trel\_meets]  
 8, 16 : [trel\_before, trel\_meets]  
 8, 17 : [trel\_before, trel\_meets]  
 8, 20 : [trel\_before, trel\_meets]  
 8, 25 : [trel\_overlaps, trel\_equals, trel\_starts, trel\_during, trel\_finishes,  
 trel\_startsi, trel\_duringi, trel\_finishesi]  
 8, 28 : [trel\_before]  
 9, 13 : [trel\_before]  
 9, 15 : [trel\_before, trel\_meets]  
 9, 16 : [trel\_before, trel\_meets]  
 9, 17 : [trel\_before, trel\_meets]  
 9, 20 : [trel\_before, trel\_meets]  
 9, 28 : [trel\_before]  
 11, 13 : [trel\_before]  
 11, 15 : [trel\_before, trel\_meets]  
 11, 16 : [trel\_before, trel\_meets]  
 11, 17 : [trel\_before, trel\_meets]  
 11, 20 : [trel\_before, trel\_meets]  
 11, 22 : [trel\_beforei]  
 11, 25 : [trel\_beforei]  
 11, 28 : [trel\_beforei]  
 13, 14 : [trel\_beforei]  
 13, 15 : [trel\_beforei, trel\_during, trel\_meetsi, trel\_overlapsi]  
 13, 16 : [trel\_beforei, trel\_during, trel\_meetsi, trel\_overlapsi]

13, 17 : [trel\_beforei , trel\_during , trel\_meetsi , trel\_overlaps]  
13, 18 : [trel\_beforei]  
13, 21 : [trel\_beforei]  
13, 22 : [trel\_beforei]  
13, 25 : [trel\_beforei]  
13, 28 : [trel\_before]  
14, 15 : [trel\_before , trel\_meets]  
14, 16 : [trel\_before , trel\_meets]  
14, 17 : [trel\_before , trel\_meets]  
14, 20 : [trel\_before , trel\_meets]  
14, 28 : [trel\_before]  
15, 18 : [trel\_beforei]  
15, 21 : [trel\_beforei]  
15, 22 : [trel\_beforei]  
15, 25 : [trel\_beforei]  
15, 28 : [trel\_before , trel\_duringi , trel\_meets , trel\_overlaps]  
16, 18 : [trel\_beforei]  
16, 21 : [trel\_beforei]  
16, 22 : [trel\_beforei]  
16, 25 : [trel\_beforei]  
16, 28 : [trel\_before , trel\_duringi , trel\_meets , trel\_overlaps]  
17, 18 : [trel\_beforei]  
17, 21 : [trel\_beforei]  
17, 22 : [trel\_beforei]  
17, 25 : [trel\_beforei]  
17, 28 : [trel\_before , trel\_duringi , trel\_meets , trel\_overlaps]  
18, 20 : [trel\_before , trel\_meets]  
18, 28 : [trel\_before]  
20, 21 : [trel\_beforei]  
20, 22 : [trel\_beforei]  
20, 25 : [trel\_beforei]  
21, 28 : [trel\_before]  
22, 28 : [trel\_before]  
25, 28 : [trel\_before]

## Texte 2

```
#----- DATES-----
# 0 : ([2003, 5, 30, 0, 0, 0, 0], [2003, 5, 30, 23, 59, 59, 999])
# 3 : ([2003, 5, 29, 17, 0, 0, 0], [2003, 5, 29, 23, 59, 59, 999])
# 6 : ([i5, i5, i5, 18, 0, 0, 0], [i5, i5, i5+1, 8, 59, 59, 999])
# 8 : ([i5, i5, i5, 3, 0, 15, 0], [i5, i5, i5+1, 3, 0, 15, 999])
# 10 : ([2003, 5, 29, 17, 0, 0, 0], [2003, 5, 29, 23, 59, 59, 999])
# 15 : ([i15, i15, i15, 0, 0, 0, 0], [i15, i15, i15, 23, 59, 59, 999])
# 16 : ([i16, 5, 21, 0, 0, 0, 0], [i16, 5, 21, 23, 59, 59, 999])
# 18 : ([2003, 5, 27, 17, 0, 0, 0], [2003, 5, 27, 23, 59, 59, 999])
# 24 : ([i24, i21, i21, 18, 0, 0, 0], [i24, i21, i21+1, 8, 59, 59, 999])
# 23 : ([i23, i23, i23, 0, 0, 0, 0], [i23, i23, i23, 23, 59, 59, 999])

#----- EVENTS-----
2 : être : ver_imp
4 : avoir brutalement réveiller : ver_pc
9 : être descendre : ver_pc
11 : avoir constater : ver_pc
12 : avoir être : ver_pc
13 : avoir faire : ver_pc
17 : avoir être de : ver_pqp
19 : paniquer : ver_imper
20 : se être installer : ver_pc
25 : être garer : ver_pc
1 : begin : ver_pres

#----- RELATIONS-----
1, 2 : [trel_beforei]
1, 4 : [trel_beforei]
1, 9 : [trel_beforei]
1, 11 : [trel_beforei]
1, 12 : [trel_beforei]
1, 13 : [trel_beforei]
1, 17 : [trel_beforei]
1, 20 : [trel_beforei]
1, 25 : [trel_beforei]
2, 17 : [trel_beforei]
9, 17 : [trel_beforei]
12, 13 : [trel_beforei]
```

### Texte 3

#----- DATES-----

# 3 : ([2003, 5, 26, 17, 0, 0, 0], [2003, 5, 26, 23, 59, 59, 999])  
# 7 : ([2003, 7, 14, 0, 0, 0, 0], [2003, 7, 14, 23, 59, 59, 999])  
# 0 : ([2003, 6, 2, 0, 0, 0, 0], [2003, 6, 2, 23, 59, 59, 999])

#----- EVENTS-----

2 : être attendre : ver\_pc  
4 : se concerter : ver\_pres  
5 : installer : ver\_inf  
8 : avoir encore manifester : ver\_pc  
9 : protester : ver\_inf  
10 : avoir décider de : ver\_pc  
11 : remplacer : ver\_inf  
12 : avoir annoncer : ver\_pc  
13 : libérer : ver\_inf  
14 : avoir annoncer : ver\_pc  
15 : avoir : ver\_fut  
16 : faire : ver\_inf  
17 : devoir également nommer de : ver\_fut  
18 : avoir : ver\_fut  
19 : compter : ver\_pres  
20 : mettre : ver\_inf  
21 : devoir aussi préparer : ver\_fut  
1 : begin : ver\_pres

#----- RELATIONS-----

1, 2 : [trel\_beforei]  
1, 4 : [trel\_before, trel\_duringi, trel\_equals, trel\_finishesi, trel\_meets, trel\_overlaps, trel\_startsi]  
1, 8 : [trel\_beforei]  
1, 10 : [trel\_beforei]  
1, 12 : [trel\_beforei]  
1, 14 : [trel\_beforei]  
1, 19 : [trel\_before, trel\_duringi, trel\_equals, trel\_finishesi, trel\_meets, trel\_overlaps, trel\_startsi]  
2, 4 : [trel\_before]  
2, 15 : [trel\_before, trel\_meets]  
2, 17 : [trel\_before, trel\_meets]  
2, 18 : [trel\_before, trel\_meets]  
2, 19 : [trel\_before]  
2, 21 : [trel\_before, trel\_meets]  
4, 8 : [trel\_beforei]  
4, 10 : [trel\_beforei]  
4, 12 : [trel\_beforei]  
4, 14 : [trel\_beforei]  
4, 19 : [trel\_before, trel\_duringi, trel\_equals, trel\_finishesi, trel\_meets, trel\_overlaps, trel\_startsi]  
8, 15 : [trel\_before, trel\_meets]  
8, 17 : [trel\_before, trel\_meets]  
8, 18 : [trel\_before, trel\_meets]  
8, 19 : [trel\_before]  
8, 21 : [trel\_before, trel\_meets]  
10, 15 : [trel\_before, trel\_meets]  
10, 17 : [trel\_before, trel\_meets]  
10, 18 : [trel\_before, trel\_meets]  
10, 19 : [trel\_before]  
10, 21 : [trel\_before, trel\_meets]  
12, 15 : [trel\_before, trel\_meets]  
12, 17 : [trel\_before, trel\_meets]  
12, 18 : [trel\_before, trel\_meets]  
12, 19 : [trel\_before]  
12, 21 : [trel\_before, trel\_meets]

14, 15 : [trel\_before, trel\_meets]  
14, 17 : [trel\_before, trel\_meets]  
14, 18 : [trel\_before, trel\_meets]  
14, 19 : [trel\_before]  
14, 21 : [trel\_before, trel\_meets]