

ELU, un environnement d'expérimentation pour la TA

Dominique Estival

Volume 37, Number 4, décembre 1992

Études et recherches en traductique / Studies and Researches in
Machine Translation

URI: <https://id.erudit.org/iderudit/002711ar>

DOI: <https://doi.org/10.7202/002711ar>

[See table of contents](#)

Publisher(s)

Les Presses de l'Université de Montréal

ISSN

0026-0452 (print)

1492-1421 (digital)

[Explore this journal](#)

Cite this article

Estival, D. (1992). ELU, un environnement d'expérimentation pour la TA. *Meta*, 37(4), 693–708. <https://doi.org/10.7202/002711ar>

Article abstract

ELU gives researchers in natural language processing (NLP) a unification-based linguistic development environment, whose theoretically validated formalism permits the experimentation with various linguistic theories and analyses for the description of natural languages, and more specifically it gives researchers in machine translation (MT) the possibility to experiment with various approaches to translation. The unification grammar paradigm is attractive for NLP because its formal properties are clear and well-established. If the resolutely theory-neutral perspective of the environment converges with some concerns we also find in linguistic theory, ELU is mainly designed for building MT prototype systems. Not only is ELU based on unification but, in each of its components, it takes advantage of the reversibility which is an inherent property of unification-based systems. Beyond grammar reversibility, i.e. in the monolingual descriptions, we can make use of reversibility in the transfer component, i.e. in the bilingual descriptions. The formalism thus allows testing the hypothesis that a MT system could, and thus maybe should, be bidirectional.

ELU, UN ENVIRONNEMENT D'EXPÉRIMENTATION POUR LA TA

DOMINIQUE ESTIVAL
Université de Genève, Genève, Suisse

Résumé

ELU donne aux chercheurs en traitement automatique du langage (TAL) un environnement de génie linguistique basé sur l'unification, dont le formalisme théoriquement validé permet d'expérimenter différentes analyses et théories linguistiques pour les descriptions des langues, et il donne plus particulièrement aux chercheurs en traduction automatique (TA) la possibilité d'expérimenter différentes approches de la traduction.

Le paradigme des grammaires d'unification est un formalisme attrayant pour le TAL parce que ses propriétés formelles sont claires et bien établies. Si l'optique théoriquement neutre de l'environnement rejoint des préoccupations que l'on retrouve chez certains chercheurs en linguistique théorique, ELU est principalement destiné à la construction de systèmes prototypés de TA.

Non seulement ELU est basé sur l'unification, mais il exploite dans chacune de ses composantes la réversibilité des grammaires, celle des descriptions linguistiques monolingues, on peut en effet pratiquer la réversibilité des descriptions bilingues, dans la composante de transfert. Le formalisme adopté permet donc de tester l'hypothèse qu'un système de TA peut, et donc devrait peut-être, être bidirectionnel.

Abstract

ELU gives researchers in natural language processing (NLP) a unification-based linguistic development environment, whose theoretically validated formalism permits the experimentation with various linguistic theories and analyses for the description of natural languages, and more specifically it gives researchers in machine translation (MT) the possibility to experiment with various approaches to translation.

The unification grammar paradigm is attractive for NLP because its formal properties are clear and well-established. If the resolutely theory-neutral perspective of the environment converges with some concerns we also find in linguistic theory, ELU is mainly designed for building MT prototype systems.

Not only is ELU based on unification but, in each of its components, it takes advantage of the reversibility which is an inherent property of unification-based systems. Beyond grammar reversibility, i.e. in the monolingual descriptions, we can make use of reversibility in the transfer component, i.e. in the bilingual descriptions. The formalism thus allows testing the hypothesis that a MT system could, and thus maybe should, be bidirectional.

1. INTRODUCTION

Cet article décrit un environnement de génie linguistique destiné à la construction de systèmes prototypes de traduction automatique. ELU (environnement linguistique d'unification) a pour but de fournir aux linguistes et aux chercheurs en traitement automatique du langage (TAL) un environnement qui offre un formalisme théoriquement validé permettant d'expérimenter différentes analyses et théories linguistiques pour les descriptions des langues, et plus particulièrement aux chercheurs en traduction automatique (TA) la possibilité d'expérimenter différentes approches de la traduction. Élaboré à l'ISSCO, il y a déjà permis la construction d'un système de traduction français-allemand (*cf.* Bouillon et Boesefeldt 1991, et le présent numéro de *Meta*).

ELU présente deux caractéristiques notables : il est basé sur l'**unification** et il exploite dans chacune de ses composantes la **réversibilité** inhérente aux systèmes basés sur l'unification.

L'originalité de ce système ne consiste pas dans l'adoption de l'unification comme processus de base, car bien d'autres systèmes de TAL l'utilisent, mais dans l'optique résolument théoriquement neutre de l'environnement, qui permet l'expérimentation de différentes théories. En ceci, il rejoint des préoccupations que l'on retrouve chez certains chercheurs en linguistique théorique : la convergence des grandes familles de théories linguistiques, et leur représentation possible dans un formalisme commun.

En tant que système basé sur l'unification, ELU peut être décrit comme un PATR-II (Shieber 1986) amélioré. En ce qui concerne l'idée d'utiliser des systèmes basés sur l'unification pour la TA, elle remonte à Kay 1984, avec le système FUG (voir également les contributions de R. Sharp et O. Streiter, et de D. Arnold et L. Sadler, dans ce numéro). Ce formalisme est attrayant parce que ses propriétés formelles sont claires et bien établies. De plus, les travaux faits dans le cadre des grammaires d'unification ces dernières années ont montré qu'il se prêtait particulièrement bien au mode d'analyse déclaratif, permettant donc d'envisager des grammaires qui soient neutres par rapport au processus qui les emploie, analyseur ou générateur, c'est-à-dire réversibles. Cette neutralité par rapport au processus offre des avantages pratiques et théoriques considérables, même si elle impose des contraintes qui peuvent modifier les analyses linguistiques et mener à une approche différente pour traiter certains problèmes (Estival 1990a).

Au-delà de la réversibilité des grammaires, celle des descriptions linguistiques monolingues, on peut aussi pratiquer la réversibilité de la composante de transfert, celle des descriptions bilingues¹. Comme base de formalisme, l'unification permet donc la réversibilité complète de chacune des composantes du système. Mais avant d'entrer en détail dans le formalisme, il convient de présenter une vue plus générale des buts de ces travaux.

2. LA TA COMME DESCRIPTION DES CONNAISSANCES

Les travaux menant à ELU ont été guidés dès le départ par une certaine conception de la TA. Dans une approche relevant de la recherche en sciences cognitives, un des buts est de «modéliser le savoir du traducteur». Ce savoir peut se caractériser comme comprenant :

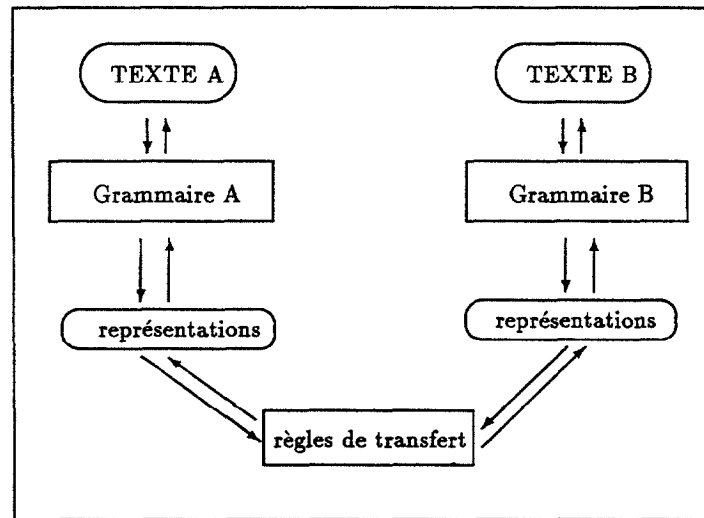
- des connaissances séparées des langues source et cible,
- un savoir-faire spécifique à la traduction.

Dans un système de TA développé dans cette optique, le premier point implique qu'il faut définir des grammaires monolingues indépendamment les unes des autres, et non des grammaires élaborées en parallèle dans le but de minimiser les efforts à fournir pour construire le module de transfert (comme dans le système des avalanches présenté dans l'article de P. Bouillon et K. Boesefeldt).

Le terme **grammaire** s'entend au sens large : la description de toutes les connaissances linguistiques nécessaires à la compréhension et à la production d'une langue. Une grammaire recouvre donc l'ensemble des informations — que ce soit des règles, contraintes, paramètres ou principes généraux ou des spécifications lexicales particulières — requises pour la description adéquate d'une langue tant aux niveaux sémantique, lexical, morphologique, phonologique, etc. que syntaxique. Cette description est représentée par un ensemble de données qui, selon la théorie linguistique adoptée, peuvent être toutes mises au même niveau ou être organisées en modules définissant des niveaux d'abstraction correspondants. Pour être utilisable aussi bien pour la compréhension (l'analyse) que pour la production (la synthèse), une description doit être réversible à tous les niveaux.

Le deuxième point implique que, pour la modélisation du savoir nécessaire à la traduction, les questions spécifiques à la traduction doivent être envisagées parallèlement au développement des grammaires monolingues, mais de manière indépendante. La traduction entre deux langues est considérée comme une mise en correspondance des représentations de ces langues. Le processus de traduction peut être représenté par le schéma classique de la Figure 1, où des représentations élaborées par des grammaires indépendantes sont mises en correspondance par un module de transfert, lui-même indépendant des grammaires. Celui-ci représente le savoir d'un traducteur entre les deux langues A et B.

Figure 1 : Le processus de traduction



Chaque grammaire produit des représentations qui font abstraction des formes spécifiques de la langue et, selon les grammaires utilisées, ces représentations peuvent se trouver à des niveaux d'abstraction différents. La définition du meilleur niveau d'abstraction, ou du type de représentation adéquat, pour effectuer le transfert est l'un des paramètres de la recherche en TA que le système permet d'explorer¹.

Tandis qu'une grammaire représente les connaissances linguistiques de tout locuteur de la langue qu'elle décrit (sa compétence), les règles de transfert entre deux langues représentent le savoir-faire du traducteur et ses connaissances sur l'équivalence des expressions d'une langue avec celles de l'autre langue.

3. LE FORMALISME D'UNIFICATION

L'unification est devenu un concept central pour un certain nombre de théories/ outils informatiques: PATR-II, FUG, GDE (cf. Shieber 1986, Kay 1983, Carroll *et al.* 1988), et de théories linguistiques: GPSG, LFG, HPSG, TAG (cf. Sag *et al.* 1986, Vijay-Shanker et Joshi 1988). Son intérêt pour la linguistique réside dans le fait que c'est un mécanisme pour combiner l'information. On peut donc s'en servir pour stipuler des corrélations entre une forme linguistique et son interprétation, entre une interprétation et sa réalisation morphologique ou phonologique, ou entre les interprétations de différentes formes linguistiques. Ce n'est cependant qu'un mécanisme informatique sous-jacent aux grammaires, et non une théorie linguistique, cognitive, ou autre.

Sa propriété principale est d'être **déclaratif**, et donc réversible en principe, mais parmi ses autres qualités nous pouvons citer l'**expressivité**: le formalisme est un métalangage qui permet d'exprimer aussi bien différents types d'analyse que des analyses à différents niveaux d'abstraction; l'**uniformité**: pour toutes les analyses, et pour toutes les composantes d'une description linguistique; la **neutralité théorique**: l'unification n'est pas une théorie linguistique, mais permet d'implémenter différentes théories syntaxiques (LFG, GPSG, HPSG, ou même GB).

3.1. UNIFICATION : LES CONCEPTS DE BASE

Tout formalisme requiert la définition des objets du domaine et celle des relations et opérations admises dans le domaine. Pour le formalisme d'unification, les **structures de traits** sont les objets que l'on manipule, et cette manipulation se fait par comparaison, grâce à la relation de **subsomption**, et par combinaison, grâce à l'opération d'**unification** imposée par des **contraintes**³.

3.1.1. STRUCTURES DE TRAITS : LES ÉLÉMENTS D'INFORMATION

Une structure de traits est un ensemble de paires <attribut valeur>, où dans chacune de ces paires (ou **traits**) la **valeur** est l'information apportée par l'attribut.

Les structures de traits servent tout aussi bien pour la représentation des éléments de base d'un système — les entrées lexicales, par exemple — que pour le niveau de représentation le plus abstrait que ce système utilise — les représentations construites par le parseur et prises comme point de départ par le générateur.

La structure de (1) est un exemple simple, représentant le nom *Pierre* avec quelques informations sur sa catégorie (*cat*) et les traits d'accord (*accord* dont la valeur est elle-même une structure de traits).

(1)	<table style="border-collapse: collapse; margin: 0 auto;"> <tr> <td style="padding: 2px 10px;"><i>tête</i></td> <td style="padding: 2px 10px;"><i>pred : Pierre</i></td> </tr> <tr> <td style="padding: 2px 10px;"><i>cat</i></td> <td style="padding: 2px 10px;"><i>n</i></td> </tr> <tr> <td style="padding: 2px 10px;"><i>accord</i></td> <td style="border-left: 1px solid black; border-right: 1px solid black; padding: 2px 10px;"> <table style="border-collapse: collapse; margin: 0 auto;"> <tr> <td style="padding: 2px 10px;"><i>genre</i></td> <td style="padding: 2px 10px;"><i>masc</i></td> </tr> <tr> <td style="padding: 2px 10px;"><i>nombre</i></td> <td style="padding: 2px 10px;"><i>sg</i></td> </tr> <tr> <td style="padding: 2px 10px;"><i>personne</i></td> <td style="padding: 2px 10px;"><i>3</i></td> </tr> </table> </td> </tr> </table>	<i>tête</i>	<i>pred : Pierre</i>	<i>cat</i>	<i>n</i>	<i>accord</i>	<table style="border-collapse: collapse; margin: 0 auto;"> <tr> <td style="padding: 2px 10px;"><i>genre</i></td> <td style="padding: 2px 10px;"><i>masc</i></td> </tr> <tr> <td style="padding: 2px 10px;"><i>nombre</i></td> <td style="padding: 2px 10px;"><i>sg</i></td> </tr> <tr> <td style="padding: 2px 10px;"><i>personne</i></td> <td style="padding: 2px 10px;"><i>3</i></td> </tr> </table>	<i>genre</i>	<i>masc</i>	<i>nombre</i>	<i>sg</i>	<i>personne</i>	<i>3</i>
<i>tête</i>	<i>pred : Pierre</i>												
<i>cat</i>	<i>n</i>												
<i>accord</i>	<table style="border-collapse: collapse; margin: 0 auto;"> <tr> <td style="padding: 2px 10px;"><i>genre</i></td> <td style="padding: 2px 10px;"><i>masc</i></td> </tr> <tr> <td style="padding: 2px 10px;"><i>nombre</i></td> <td style="padding: 2px 10px;"><i>sg</i></td> </tr> <tr> <td style="padding: 2px 10px;"><i>personne</i></td> <td style="padding: 2px 10px;"><i>3</i></td> </tr> </table>	<i>genre</i>	<i>masc</i>	<i>nombre</i>	<i>sg</i>	<i>personne</i>	<i>3</i>						
<i>genre</i>	<i>masc</i>												
<i>nombre</i>	<i>sg</i>												
<i>personne</i>	<i>3</i>												

3.1.2. UNIFICATION : OPÉRATION COMBINANT L'INFORMATION

Pour combiner les informations contenues dans plusieurs structures de traits, l'opération d'unification se base sur la relation de subsomption. Celle-ci mesure le degré d'informativité d'une structure par rapport à une autre: on dit qu'une structure de traits en subsume une autre si elle contient moins d'informations, mais que toutes leurs informations sont compatibles. Le résultat de l'opération d'unification est une nouvelle structure de traits qui contient tous les traits des structures de départ. Si les structures de départ contiennent des informations incompatibles, l'opération échoue. L'unification de deux structures constitue le mécanisme de base pour résoudre les contraintes dans les descriptions et créer les structures d'arrivée⁴.

3.1.3. CONTRAINTES : CONDITIONS DE BONNE FORMATION

Une contrainte est la description d'un chemin de paires <attribut valeur> dans une structure de traits. Résoudre une contrainte, c'est vérifier que cette description est compatible (s'unifie) avec la structure de traits considérée.

Par exemple, la contrainte (2a.) requiert que l'objet <SUJET> ait la valeur <nominatif> pour son attribut <cas> et la contrainte (2b.) requiert que les descriptions des deux objets <DET> et <NOM> coïncident en ce qui concerne leur attribut <accord>.

- (2) a. <SUJET cas> = nominatif
 b. <DET accord> = <NOM accord>

Les contraintes sont le seul moyen non seulement de spécifier l'information, mais aussi de spécifier comment utiliser les informations contenues dans les structures de traits. Pour obtenir un résultat, toutes les contraintes applicables doivent avoir été résolues et l'unification ne doit pas avoir échoué. Il n'y a pas d'ordre entre les contraintes, seul le résultat final compte, et c'est ce qui rend le système déclaratif.

3.1.4. GRAMMAIRE D'UNIFICATION

Une grammaire basée sur l'unification est une description dont les objets sont des mots ou des syntagmes, ou des structures de traits décrivant des ensembles de propriétés linguistiques. Dans un formalisme d'unification, une règle est un ensemble de contraintes s'appliquant à des structures de traits. Toutes les contraintes dans les descriptions linguistiques sont du même type, que ce soit dans les entrées lexicales, les données morphologiques, les règles syntaxiques ou les règles de transfert.

3.2. UNIFICATION : PROBLÈMES ET EXTENSIONS

Tout formalisme basé sur l'unification pose deux problèmes classiques, à savoir la **complétude** et la **cohérence** des structures de traits (*cf.* Kaplan et Bresnan 1982). Ces questions sont toujours relatives à une certaine grammaire et à l'utilisation que l'on veut faire des structures d'arrivée ; en particulier, les algorithmes d'analyse, de synthèse et de transfert ne garantissent pas tous la cohérence et la complétude des structures de traits qu'ils créent. Aussi, malgré la déclarativité qui permet de séparer les données des processus, l'utilisateur d'un système basé sur l'unification doit prêter attention à certaines propriétés des structures de traits d'arrivée en écrivant sa grammaire.

La complétude demande qu'une structure d'arrivée n'omette pas d'information contenue dans une structure de départ. La cohérence demande qu'une structure d'arrivée ne contienne pas d'information supplémentaire non validée par une règle. Pour préserver la complétude et la cohérence, on doit utiliser des structures pré-contraintes, telles que **listes**, **arbres**, ou **types**, et si on utilise des variables, comme dans les exemples (2a.) et (2b.), celles-ci doivent être instanciées.

Les structures pré-contraintes que l'on utilise pour résoudre les problèmes de complétude et de cohérence sont une des extensions classiques du formalisme d'unification pur. Deux autres extensions, qui se sont révélées nécessaires pour que les systèmes construits dans ce formalisme soient plus facilement utilisables, sont la possibilité d'exprimer la **négation** («~») et la **disjonction** («/») dans les descriptions de structures de traits. Toutes deux permettent en effet une syntaxe plus souple qui permet d'exprimer de manière plus succincte des contraintes qui deviennent rapidement très complexes. Elles introduisent cependant un certain degré d'incertitude, et donc de non-déterminisme. Pour éviter de trop grands problèmes de complexité, la négation dans ELU ne peut être qu'atomique et ne porte donc que sur une valeur en fin de chemin d'attributs. De même, la disjonction, que l'on saurait pourtant manipuler pour des chemins plus complexes, est limitée aux valeurs atomiques.

4. LES COMPOSANTES DU SYSTÈME ELU

Comme il est plus aisé d'introduire le formalisme en décrivant les grammaires monolingues que les descriptions bilingues, cette section décrit les modules d'analyse et de synthèse utilisés par ELU avant de présenter la composante de transfert.

Un parseur est un algorithme pour appliquer une grammaire à des données (un texte et une grammaire) et obtenir en sortie une (ou plusieurs) représentation(s) du texte en fonction de la grammaire. De même, un générateur est un algorithme pour appliquer une grammaire à une représentation et obtenir en sortie la (ou les) réalisation(s) de cette représentation en fonction de cette grammaire. Ces deux algorithmes, d'analyse et de synthèse, doivent être clairement séparés de la grammaire avec laquelle ils opèrent.

4.1. DESCRIPTION DU PARSEUR

ELU utilise un parseur non déterministe, qui produit toutes les analyses de la phrase d'entrée validées par la grammaire. Nous verrons qu'une grammaire d'unification peut être divisée en deux parties distinctes, un squelette non contextuel et un ensemble de contraintes, et le parseur reflète cette division en opérant en deux passes⁵.

Le squelette non contextuel exprime les relations de précédence linéaire et de dominance immédiate entre constituants, et la première passe se base sur l'algorithme de Earley pour construire l'arbre de dérivation. L'ensemble de contraintes peut exprimer n'importe quel type d'informations (y compris des relations entre les constituants) et la seconde passe résout les contraintes des règles utilisées pour construire cet arbre. La résolution de ces contraintes peut demander de résoudre d'autres contraintes invoquées par récursion dans d'autres parties de la grammaire. Toute la grammaire peut ainsi être impliquée au niveau de cette deuxième passe, sans ordre imposé entre les différentes contraintes.

4.2. DESCRIPTION DU GÉNÉRATEUR

Tandis que les techniques d'analyse ont fait l'objet de recherches intensives pendant plusieurs années et qu'il existe déjà un certain nombre de parseurs pour les systèmes basés sur l'unification, la synthèse est un sujet de recherche plus récent dans ce formalisme, même en restant au niveau purement syntaxique et sans entrer dans les problèmes de discours. Tous les algorithmes de génération élaborés jusqu'à présent (*cf.* Dymetman et Isabelle 1988, Russell *et al.* 1990, Shieber *et al.* 1989, Saint-Dizier 1989) imposent des contraintes spécifiques sur les grammaires, de même que certains algorithmes d'analyse peuvent imposer des contraintes spécifiques sur les types de grammaires pour lesquels ils conviennent.

L'algorithme de génération de ELU (basé sur celui de Shieber *et al.* 1989) peut procéder de manière ascendante à travers les «têtes» de règles «chaînées». La notion de «tête sémantique» d'une règle est cruciale car c'est elle qui détermine si deux règles sont chaînées (si elles ont la même tête), ce qui permet un traitement ascendant plutôt que descendant. C'est aussi la «tête sémantique» de la règle qui définit l'information partagée automatiquement entre les nœuds de la mère et de la fille aînée. Bien que la distinction entre les types d'information soit toujours arbitraire dans un tel système, le plus grand soin est nécessaire pour déterminer quels attributs appartiennent au domaine de la syntaxe ou à celui de la sémantique.

4.3. FORMALISME POUR LA GRAMMAIRE

Nous pouvons maintenant spécifier la forme d'une grammaire et montrer comment des principes linguistiques peuvent être instanciés dans une description déclarative. En (3), nous avons un exemple d'une règle de grammaire simple, où la première ligne est la règle de réécriture non contextuelle du nœud mère «p» (phrase) en ses deux nœuds filles «sn» (syntagme nominal) et «sv» (syntagme verbal), et où les lignes suivantes sont les contraintes sur ces trois syntagmes (en particulier celle qui fait du SV la tête de la règle).

- (3) p → sn sv
 <p tete> = <sv tete>
 <p accord> = <sn accord> = <sv accord>
 <p sujet> = <sn>
 <sv tete forme> = tensee
 <p subcat> = <sv subcat> — sn = []

Les contraintes en (3) sont stipulées de manière spécifique pour les éléments de cette règle. Dans une grammaire réelle, les règles et les représentations doivent être plus générales, et on a besoin de mécanismes pour construire des grammaires plus concises et plus expressives. En particulier, les «macros paramétrées» constituent un moyen élégant et efficace d'exprimer des généralisations, car elles permettent de raccourcir l'expression d'ensembles de contraintes. Elles représentent un mécanisme beaucoup plus puissant que de simples abréviations puisque non seulement elles peuvent prendre des arguments, mais que ces arguments peuvent être des variables. De plus, dans ELU, les macros paramétrées admettent des définitions récursives (dans lesquelles d'autres macros peuvent être invoquées), et des définitions multiples (exprimant un choix au niveau de la généralisation à appliquer). Même si des définitions multiples et récursives introduisent un degré d'indéterminisme qui peut être difficile à maîtriser en génération (Russell *et al.* 1990), les macros paramétrées permettent d'imposer des restrictions linguistiques (lexicales, morphologiques ou syntaxiques) de manière élégante et concise tout en conservant la déclarativité et la monotonie du formalisme⁶. En utilisant certains de ces mécanismes et extensions, la simple règle de (3) peut être transformée en (4).

- (4) * → SN H
 <H cat> = sv
 <SN cat> = sn
 !Tete (H)
 !Accord (SN, H)
 !Sujet (SN)
 !Tensee
 <* subcat> = <H subcat> — SN = []
 !Gap(SN, H)

En (4), les trois syntagmes de la partie non contextuelle sont représentés par des variables qui peuvent servir d'argument à des macros (les variables commencent par une majuscule et «!» précède l'appel à une macro)⁷. Celles-ci expriment des généralisations qui évitent de répéter les mêmes groupes d'équations de contraintes dans différentes règles de grammaire. Par exemple, la macro «Tête (H)» permet de spécifier dans n'importe quelle règle la tête sémantique de cette règle.

De même la macro paramétrée «Accord (X,Y)», définie comme en (5), remplacera toute une série de contraintes du modèle de (2b.), puisque les variables X et Y peuvent être instanciées par DET et NOM comme dans (2b.), ou par SN et H comme dans (4). Cette macro aura toujours l'effet de contraindre deux éléments à avoir la même valeur pour leur attribut <accord>

- (5) !Accord (X, Y)
 <X accord> = <Y accord>

Les macros peuvent aussi représenter des faits de langue idiosyncratiques qui encombreraient les règles s'ils n'étaient isolés du reste de la grammaire, par exemple l'alternance *qui-que* pour les complémenteurs interrogatifs (opposition humain/non humain) qui n'est pas la même que celle pour les pronoms relatifs (sujet/objet).

Comme nous l'avons vu, toute contrainte, quelle qu'elle soit, peut être invoquée en cours d'analyse, et rien n'empêche qu'une contrainte dite morphologique ou sémantique intervienne pour bloquer l'application d'une règle dite syntaxique. En d'autres termes, c'est au linguiste écrivant sa grammaire qu'il incombe de s'assurer qu'une division théorique de celle-ci en modules distincts soit bien respectée dans la pratique. Pour maintenir cette modularité, il est bon par exemple que les règles non contextuelles, qui construisent l'arbre de dérivation et peuvent être considérées comme purement syntaxiques, ne fassent appel dans leurs contraintes qu'à des macros qui ont été préalablement définies comme syntaxiques et concernent les relations entre les constituants syntaxiques qui sont concaténés. De même, les entrées lexicales ne devraient faire appel qu'à des macros concernant les propriétés syntaxiques et sémantiques des entrées lexicales. Ces propriétés peuvent être soit dues à des processus généraux (règles de redondance lexicale pour la sous-catégorisation, la diathèse, les propriétés de contrôle des verbes, etc.), soit idiosyncratiques (par exemple, les formes contractées des prépositions / articles : *du, des, au, aux*)⁸.

4.4. RÉVERSIBILITÉ DE LA GRAMMAIRE

Une «grammaire générative» est une grammaire considérée formellement du point de vue des phrases engendrées, mais les analyses syntaxiques se placent le plus souvent du point de vue de la reconnaissance et de l'analyse des phrases, et on ne considère la synthèse que comme un test vérifiant la validité de l'analyse. Des mécanismes spéciaux tels que filtres, contraintes trans-dérivationnelles, etc. sont alors nécessaires pour éviter la surgénération (production de phrases qui ne sont pas analysables par la grammaire). Nous nous plaçons ici dans une optique de réversibilité complète où la grammaire est utilisable indifféremment, sans contraintes supplémentaires, pour l'analyse et pour la synthèse⁹.

Comme l'utilisation de la grammaire par l'algorithme de génération impose des contraintes, certains compromis doivent être consentis pour assurer à la fois cette réversibilité et une certaine efficacité. En particulier, l'obligation de choisir un des éléments dominés (la «fille aînée») dans une règle de grammaire comme étant la «tête» de cette règle peut forcer à réviser le traitement de telle ou telle construction. Néanmoins, la réversibilité apporte des avantages considérables lors de l'élaboration et de l'utilisation d'une grammaire, non seulement en ce qui concerne la facilité de développement et de maintenance, comme on le fait généralement remarquer, mais aussi parce qu'elle impose une certaine rigueur de l'analyse et une cohérence formelle dans les traitements de phénomènes linguistiques.

Il faut souligner à ce propos que si la notion de «tête sémantique» d'une règle est cruciale dans l'utilisation de la grammaire en synthèse, une notion de tête se trouve par ailleurs être extrêmement importante dans nombre d'analyses syntaxiques (en HPSG, TAG, GB, entre autres) pour des raisons purement linguistiques. Un mécanisme formel est donc indépendamment validé par la théorie.

4.5. DESCRIPTION DU TRANSFERT

Conformément au schéma de la Figure 1, la traduction avec ELU est effectuée par un module de transfert mettant en correspondance des descriptions linguistiques. Les modules de transfert n'ont que récemment commencé à être décrits de manière précise dans la littérature sur la TA. En effet, les problèmes de traduction ont longtemps été envisagés en termes de procédures utilisées pour les résoudre et non en termes de principes mis en cause. Les formalismes basés sur l'unification ont permis de décomposer les problèmes de manière déclarative et d'établir une séparation claire entre les différents types de problèmes et les solutions à y apporter¹⁰. Le module de transfert de ELU est une extension du formalisme de l'unification, car les descriptions linguistiques sur lesquelles

opèrent les règles de transfert sont des structures de traits et la relation sur laquelle est basée l'opération de transfert est la subsomption.

4.5.1. TRANSFERT DE STRUCTURES DE TRAITS

Chaque règle de transfert décrit une mise en correspondance de deux structures de traits, qui sont des représentations dans deux langues différentes. Si l'une est le résultat de l'analyse dans une des langues, l'autre peut servir de données pour la synthèse dans l'autre langue. On définit ainsi le transfert comme une relation déclarative et bidirectionnelle entre deux structures de traits. Un exemple relativement simple est illustré par les phrases de (6) qui présentent un des problèmes de traduction classiques, où la relation tête-modificateur dans une langue est inversée dans l'autre.

- (6) a. *Maria schwimmt gern.*
- b. *Maria aime nager.*

Le modificateur du verbe allemand *gern* de (6a.) se traduit par le verbe principal français *aimer* en (6b.), tandis que le verbe principal allemand se traduit par le complément infinitif en français. Les structures de traits (7a.) et (7b.) sont des représentations (simplifiées) de (6) et (6b.).

- (7) a.

[<i>tete</i>	<i>sem</i>	<i>pred</i>	<i>schwimmen</i>]
			<i>args</i>	[<1> <i>tete sem pred Maria</i>]	
			<i>mod</i>	[<i>gern</i>]	
- b.

[<i>tete</i>	<i>sem</i>	<i>pred</i>	<i>aimer</i>]
				[<1> <i>tete sem pred Maria</i>	
			<i>arg</i>	<2> <i>tete sem pred nager</i>	
				<i>args</i> [# 1]	

Dans chacune de ces structures, les chemins d'attributs décrivent la tête du prédicat principal de la phrase, la liste (notée par «[...]») de ses arguments, et pour (7a.) la liste des modificateurs. Dans (7b.), l'élément «# 1» dénote la variable qui reprend le premier argument du prédicat principal introduit par <1> (ici *Maria*).

Avec ces représentations, la traduction de ces phrases peut s'effectuer grâce aux règles de transfert données en (8), dont nous examinerons plus particulièrement la règle *gern-aimer* dans la section suivante. Cet exemple donne un ensemble de règles de transfert complet qui pourrait apparaître en tant que tel dans un programme ELU.

- (8) # Transfer german french
- :PATH1: <* cat>
- <* tete>
- :PATH2: <* cat>
- <* tete>
- :T: cat
- :L1: <* cat> = Cg
- :L2: <* cat> = Cf
- :X: Cg <=> Cf
- :T: general
- :L1: <* tete sem pred> = Rg
- <* tete sem args> = Lg
- :L2: <* tete sem pred> = Rf
- <* tete sem args> = Lf
- :X: Rg <=> Rf
- Lg <=> Lf

```

:T: gern-aimer
:L1: <* tete sem pred> = Rg
      <* tete sem args> = [AglTg]
      <* tete sem mod> = gern
:L2: <* tete sem pred> = aimer
      <* tete sem args> = [Af,Vf]
      <Vf tete sem pred> = Rf
      <Vf tete sem args> = [AflTf]
:X: Rg <=> Rf
     Ag <=> Af
     Tg <=> Tf

:TA: p p
:TA: Paul Paul
:TA: Maria Maria
:TA: lieben aimer
:TA: schwimmen nager

```

4.5.2. FORMALISME POUR LE TRANSFERT

Comme les grammaires monolingues, l'ensemble des règles de transfert obéit lui aussi à un formalisme, qui est d'ailleurs presque identique à celui des grammaires. Parmi les différences, notons tout d'abord que puisque tous les attributs d'une représentation ne sont pas à traduire (par exemple, on ne traduit pas les informations morphologiques qui découlent de la grammaire utilisée en synthèse), l'utilisateur spécifie quels chemins d'attributs doivent être pris en considération dans les contraintes (PATHS1 et PATHS2).

Ensuite, il convient de séparer les règles atomiques, indiquées par «:TA», des règles structurales, indiquées par «:T:». Les premières mettent en correspondance des atomes, c'est-à-dire des valeurs terminales dans les structures de traits, les secondes mettent en correspondance des structures qui peuvent être complexes. Cette division ne correspond cependant pas à la division classique entre «transfert lexical» et «transfert structural»¹¹. En effet, si une règle atomique telle que «:TA: schwimmen nager» dans (8) représente bien une mise en correspondance de deux éléments lexicaux simples, la règle atomique «:TA: p p» met en correspondance des éléments de la description linguistique (ici la catégorie syntaxique de la phrase) sans réalisation lexicale, tandis que la règle «:T: gern-aimer» représenterait l'entrée d'un dictionnaire bilingue. La règle *gern-aimer* n'est cependant pas une règle atomique, car elle donne non seulement une équivalence entre les éléments lexicaux *gern* et *aimer*, mais elle donne aussi des contraintes spécifiant les conditions sous lesquelles cette équivalence est valable, et des équations pour le transfert d'autres éléments dans les structures de traits considérées.

Comme toute règle dans un formalisme basé sur l'unification, une règle de transfert contient des contraintes: ici, deux ensembles de contraintes portant sur les descriptions monolingues (L1 et L2), chacun décrivant en partie soit la structure de départ, soit la structure d'arrivée; le troisième ensemble (X) contient les équations de correspondance pour les variables mentionnées en L1 ou en L2. Ces trois ensembles de contraintes correspondent aux trois composantes du système de TA de la Figure 1: un ensemble pour la description dans une langue, un ensemble pour la description dans une autre langue, et un ensemble de correspondances de traduction entre ces deux descriptions.

La règle *gern-aimer* s'interprète de la manière suivante:

- les chemins d'attributs donnés en L1 spécifient que la tête de la structure en allemand est un élément qui a au moins un argument, et que cette structure contient un élément modificateur, réalisé par l'élément lexical *gern*;

- les chemins d'attributs donnés en L2 spécifient que la tête de la structure en français est l'élément lexical *aimer*, que celui-ci a deux arguments, et que le premier argument de *aimer* (son sujet) est identique au premier argument de son deuxième argument (contrôle du sujet du complément à l'infinitif)¹²;
- les équations sous X indiquent, de manière récursive, qu'il existe une relation de transfert entre l'élément tête de la structure allemande (son verbe) et la tête du deuxième argument de la tête de la structure française (le complément à l'infinitif), ainsi qu'entre leurs arguments respectifs.

La règle ne peut s'appliquer avec succès que si toutes les contraintes, y compris celles requises par les équations sous X, sont satisfaites.

4.5.3. RÉVERSIBILITÉ DU TRANSFERT

Comme une règle de grammaire monolingue, une règle de transfert est bidirectionnelle. En effet, aussi bien L1 que L2 peut être prise comme langue de départ. Pour (8), L1 est la langue de départ en traduisant de l'allemand vers le français, tandis que c'est la langue d'arrivée en traduisant du français vers l'allemand. Quand les règles de transfert sont compilées, L1 est la langue de départ dans une direction, L2 dans l'autre.

Puisque chaque règle de transfert est birectionnelle, l'ensemble des règles de transfert est lui-même bidirectionnel. Comme pour les grammaires cependant, la réversibilité du transfert n'est pas garantie par le formalisme, elle est seulement rendue possible. De même que l'on peut écrire une grammaire qui, malgré le formalisme, ne soit utilisable qu'en analyse ou qu'en synthèse, de même on peut écrire un module de règles de transfert qui ne soit utilisable que dans l'une des deux directions. La réversibilité reste l'entière responsabilité de l'utilisateur.

Pour une règle donnée, être réversible implique pouvoir être appliquée dans l'une ou l'autre direction. Pour un ensemble de règles, la réversibilité s'interprète de la manière suivante : si on opère le transfert dans une direction à partir d'une structure de traits S_i , et qu'on opère ensuite le transfert dans l'autre direction à partir de n'importe quelle structure de traits d'arrivée S_j , la structure de départ S_i sera nécessairement l'une des structures de traits d'arrivée de ce deuxième transfert à partir de S_j ¹³.

Il faut noter que la notion de réversibilité n'est applicable qu'au niveau de l'ensemble des règles non compilées : par définition, la compilation des règles de transfert crée deux programmes séparés qui sont chacun organisés en fonction de la direction. Nous examinons dans la section suivante quelques-unes des propriétés de la relation de transfert, telle qu'elle est définie pour le module de transfert d'ELU.

5. LA RELATION DE TRANSFERT

La relation de transfert entre deux structures de traits est définie tant par l'ensemble des règles de transfert entre les deux langues pour lesquelles ces structures sont des représentations valides, que par la manière dont ces règles sont appliquées. Si les règles elles-mêmes sont déclaratives et peuvent être interprétées indépendamment du processus qui les applique, la composante de transfert présuppose une stratégie de contrôle. Pour ELU, cette stratégie est l'application récursive des règles à partir de la racine de la structure de départ. Elle utilise : (a) la relation de subsomption entre la description donnée par une règle et la structure de départ, pour choisir à quelle partie de la structure appliquer cette règle ; (b) un ordre partiel des règles de transfert, pour choisir quelles règles appliquer. Tandis que (a) fait partie du formalisme sur lequel ELU se base, puisque la subsomption est la relation fondamentale de l'unification, (b) n'est pas requis par le formalisme et résulte d'un choix théorique. Avant de revenir sur les conséquences de ce choix, en particulier son effet sur la réversibilité, précisons d'abord le mécanisme utilisé.

Pour une structure de départ S_j , toutes les règles applicables sont essayées en parallèle. Concrètement, cela veut dire que pour chaque règle, on vérifie la relation de subsomption entre la description de départ dans la règle et S_j . Si cette relation est vérifiée, alors on applique la règle en liant les variables de cette description à des éléments de S_j .

Puisque le transfert crée une structure de traits, on retrouve dans cette composante les problèmes de la cohérence et de la complétude de la structure de traits d'arrivée, et celui de la terminaison du processus.

Ainsi, les règles doivent garantir la cohérence : si aucune règle de transfert n'introduit d'information superflue dans sa description d'arrivée, la structure d'arrivée produite par le transfert sera cohérente par rapport à la structure de départ, car elle doit être subsumée par toutes les descriptions d'arrivée des règles qui se sont appliquées à la structure de départ. De plus, pour qu'une règle de transfert soit applicable, sa description de départ doit subsumer la structure de départ.

La complétude de la structure d'arrivée par rapport à la structure de départ est vérifiée si la relation de subsomption est obtenue entre chaque élément à traduire dans la structure de départ et la structure d'arrivée, car alors toute l'information traduisible a bien été traduite.

En résumé, une règle de transfert s'applique avec succès si sa description de départ subsume la structure de départ produite par l'analyse du texte de départ, et si sa description d'arrivée s'unifie avec la structure d'arrivée qui sera la donnée du processus de synthèse. La terminaison du processus est garantie s'il n'y a pas de règle induisant de récursion infinie, et si seules des structures de traits acycliques sont utilisées.

5.1. RÉ-ENTRÉE

La ré-entrée des valeurs dans une structure de traits est un mécanisme très puissant dans les grammaires d'unification (*cf.* Sadler et Arnold dans le présent numéro). On trouve une ré-entrée dans une structure de traits quand deux attributs partagent la même valeur. Généralement, on se sert des ré-entrées pour exprimer des phénomènes linguistiques tels que l'accord ou le contrôle, pour lesquels il n'est pas suffisant que les valeurs de deux attributs soient semblables, mais où il est nécessaire que ces valeurs lient les mêmes objets. Par exemple, la variable <1> dans l'exemple (7b.) représente une ré-entrée : la valeur du sujet de l'infinitif est celle du sujet du verbe principal.

Les variables ré-entrées demandent une attention spéciale pour le contrôle du transfert et la mise en correspondance de ces objets. L'approche adoptée pour ELU est de ne traiter comme telles que les ré-entrées qui sont explicitement mentionnées dans les règles de transfert.

Par exemple, la règle de transfert *wollen-vouloir* en (9) requiert spécifiquement une ré-entrée : de chaque côté de la règle, dans L1 et L2, deux attributs partagent une variable («Ag» et «Af» respectivement), et ces variables sont mises en correspondance par une équation dans X.

- (9) :T: *wollen-vouloir*
 :L1: <* tete sem pred> = *wollen*
 <* tete sem args> = [Ag,Vg]
 <Vf tete sem pred> = Rg
 <Vf tete sem args> = [Ag|Tg]
 :L2: <* tete sem pred> = *vouloir*
 <* tete sem args> = [Af,Vf]
 <Vf tete sem pred> = Rf
 <Vf tete sem args> = [A|Tf]
 :X: Rg <=> Rf
 Ag <=> Af
 Tg <=> Tf

La ré-entrée est ainsi préservée durant le transfert entre (10a.) et (10b.)¹⁴.

- (10) a. *Maria will schwimmen gehen.*
 b. Maria veut aller nager.

Dans la règle *gern-aimer*, où il n'y a une ré-entrée que d'un côté de la règle (L2), cette ré-entrée sera créée dans la structure d'arrivée durant le transfert entre l'allemand et le français, mais seulement vérifiée dans la structure de départ lors du transfert entre le français et l'allemand.

Quand une valeur ré-entrée n'est pas explicitement mentionnée comme telle dans une règle de transfert, cela équivaut au déploiement de la structure de départ et à celui de la structure d'arrivée : la valeur n'y est pas ré-entrée et les deux instances de l'attribut ne lient pas la même variable. Néanmoins, les deux instances de l'attribut dans la structure d'arrivée peuvent recevoir la même traduction à condition que les mêmes règles de transfert s'appliquent aux deux variables qui les instancient. La ré-entrée peut aussi être reconstruite durant la phase de synthèse, si la grammaire de la langue d'arrivée peut déterminer à partir de l'information présente dans la structure d'arrivée que deux variables sont liées par le même élément.

Ainsi l'utilisateur a-t-il la possibilité de contrôler les ré-entrées. Ce mécanisme permet de traiter des cas où les phénomènes de contrôle et d'anaphore sont différents entre les langues, et il ne semble ni nécessaire ni souhaitable de conserver automatiquement les ré-entrées pour la traduction¹⁵.

5.2. SPÉCIFICITÉ

Tandis que le problème de la ré-entrée ne se présente, en ces termes du moins, que dans les systèmes basés sur l'unification, la question de la préséance de certaines règles de traduction sur d'autres se pose dans tout système de TA. Cependant, dans un système où par définition il n'y a pas d'ordre entre les règles, cette question doit être résolue par un mécanisme extérieur au formalisme.

Sans définir ici de manière formelle la relation de **spécificité** qui ordonne les règles de transfert dans ELU (le lecteur intéressé peut en trouver une description détaillée dans Estival *et al.* 1990), il est bon d'en expliquer la motivation principale, car en ceci le système se comporte différemment pour l'analyse/synthèse que pour le transfert. En effet, les stratégies d'analyse et de synthèse n'empêchent pas l'application d'une règle en cas de succès d'une autre règle. Si une règle peut être appliquée, elle l'est, et de même, toutes les structures de traits qui peuvent être construites grâce à des règles applicables le sont (c'est pourquoi le parseur est non déterministe).

Pour le transfert, ce n'est pas le cas, et toutes les structures qui seraient compatibles avec des règles de transfert applicables ne sont pas nécessairement produites : les règles les plus spécifiques empêchent l'application des règles moins spécifiques. La stratégie de contrôle basée sur la relation de spécificité définie entre les règles de transfert représente donc une hypothèse assez forte pour la théorie de la traduction.

Celle-ci s'est d'ailleurs avérée prometteuse, comme en témoigne le système de traduction des bulletins d'avalanches déjà cité. En effet, il est souvent préférable de pouvoir spécifier une traduction plus appropriée à certains contextes et de bloquer ainsi les traductions plus générales. Un exemple classique dans la littérature sur le sujet est l'équivalence lexicale *cheval* — *Pferd* qui doit céder le pas à *cheval blanc* — *Schimmel*. De même, toutes les expressions idiomatiques, qui doivent être analysées de manière non compositionnelle, telle *pomme de terre* — *Kartoffel* où *pomme* n'est pas traduit par *Apfel*.

La stratégie basée sur la relation de spécificité permet de bloquer de manière élégante et conséquente toutes les règles plus générales en présence du contexte requis

par une règle spécifique. L'ordre partiel établi entre les règles de transfert étant intégré dans la composante de transfert, l'utilisateur n'a pas à se préoccuper de spécifier, pour chaque règle particulière, quelles règles plus générales doivent être bloquées, ni sous quelles conditions ; ces conditions font partie de la règle spécifique elle-même. De plus, la relation entre les règles de transfert étant définie formellement à partir de la relation de subsomption, l'uniformité du formalisme dans le système est assurée¹⁶.

Cette stratégie permet au processus de transfert d'opérer de manière plus efficace en limitant l'espace de recherche. Cependant, si le blocage automatique des règles s'avérait une contrainte trop forte, l'hypothèse d'un ordre partiel entre les règles de transfert pourrait être maintenue et la relation de spécificité garder une utilité. ELU est un outil de développement, et l'ordre entre les règles qui définit formellement le début et la fin du processus de traduction permet à l'utilisateur d'en suivre le déroulement. De plus, un ordre entre les règles définit un ordre entre le résultat de leur application : si plusieurs résultats sont obtenus, cet ordre peut révéler des généralisations possibles et permettre un affinement des règles elles-mêmes.

Si l'hypothèse même d'un ordre entre les règles était abandonnée, l'ensemble des règles de transfert déjà établies pour une paire de langues données pourrait demeurer tel quel. Seule la stratégie de contrôle du module de transfert de ELU devrait être modifiée pour laisser s'appliquer toutes les règles applicables et produire toutes les structures d'arrivée compatibles, même les moins plausibles, dans n'importe quel ordre¹⁷.

6. CONCLUSION

L'unification offre aux chercheurs un formalisme avec des propriétés simples et mathématiquement bien définies. En particulier, la localité des règles et leur déclarativité présentent des avantages évidents pour le TAL. La bidirectionalité des grammaires qui en découle représente pour la TA un gain de temps et une facilité de maintenance considérables, alliés à une rigueur et à une logique des analyses linguistiques, qui en font un outil de recherche précieux.

La grammaire du français élaborée pour le prototype nous a permis d'avancer dans la recherche sur la construction de grammaires réversibles, et nous continuons nos travaux dans cette direction. Un autre aspect du projet concerne le lexique et la spécification des informations syntaxiques qui doivent y être présentes. Bien que ce ne soit qu'un système de recherche, ELU permet de travailler avec des dictionnaires de taille réelle, et nous sommes en train d'effectuer la conversion des entrées d'un gros lexique pour pouvoir l'utiliser avec la grammaire du français.

Par ailleurs, en développant un prototype de système de TA, nous avons défini de manière précise le concept de transfert en tant que mise en correspondance entre des structures de traits, et cela nous a permis de démontrer que le formalisme d'unification peut être étendu à la composante de transfert sans perdre ses qualités principales.

Les propriétés d'une composante de transfert réversible sont elles-mêmes beaucoup mieux comprises maintenant, ainsi que ses conséquences et implications. Par exemple, la définition de la réversibilité, qui est la même pour des règles de grammaire monolingues et pour des règles de transfert bilingues, ne peut être identique au niveau de la grammaire et de la composante de transfert. Une autre direction de recherche à poursuivre concerne les limites de l'ordre entre les règles de transfert.

En résumé, avec sa composante de transfert basé sur l'unification, ELU offre aux chercheurs en TA un outil et un environnement dont les propriétés formelles permettent d'expérimenter différentes théories linguistiques et traductives.

Notes

1. Nous verrons que les règles de transfert sont des règles de correspondance qui décrivent des relations entre descriptions monolingues. Si ces descriptions sont à un niveau d'abstraction suffisamment élevé, les relations décrites par les règles de transfert peuvent être considérées comme relevant d'une interlangue.
2. Si l'on suit l'hypothèse que le niveau de représentation le plus abstrait présente le plus d'uniformité d'une langue à une autre, et que c'est donc celui auquel devrait se faire la mise en correspondance pour le processus de traduction, on choisira, selon la théorie linguistique adoptée, la représentation sémantique, la forme logique, le niveau de structure fonctionnelle, etc.
3. Une version plus détaillée de cette partie de l'article se trouve dans Estival à paraître.
4. Voir la contribution de Sadler et Arnold, dans le présent volume, pour une description plus détaillée de ce mécanisme.
5. Il n'est pas nécessaire qu'un parseur reflète cette division, car l'information exprimée dans la partie non contextuelle des règles pourrait aussi être exprimée avec des contraintes. De même que l'on peut avoir une grammaire qui n'accorde pas un statut spécial aux relations de dominance immédiate ou de précedence linéaire, de même pourrait-on choisir un analyseur qui prenne simultanément en considération les deux types d'information.
6. Par exemple Estival 1991 montre comment implémenter l'esprit de la théorie GB — contraindre l'application de principes généraux plutôt que l'application de règles particulières — en représentant le système de principes qui constituent cette théorie avec des macros paramétrées.
7. L'utilisation de variables nous permet maintenant de mettre en œuvre, avec la macro «Gap», le mécanisme dit de «passage de trous» pour traiter des dépendances non bornées, que nous n'illustrons pas ici.
8. Le lexique fait bien aussi appel à des macros morphologiques pour les formes fléchies et composées.
9. Si «l'hypothèse zéro», selon laquelle les processus de compréhension et de production du langage seraient exactement l'inverse l'un de l'autre, est largement démentie par l'évidence psycho-linguistique, il n'a jamais été démontré que les données accessibles à ces processus doivent être séparées, et il n'y a pas d'arguments théoriques convaincants pour une approche dans laquelle les grammaires de synthèse et d'analyse seraient distinctes.
10. Cf. en particulier, les Actes de la *Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages* (Austin TX, June 90), qui contiennent des contributions montrant une convergence avec nos propres travaux.
11. D'un point de vue linguistique, le transfert lexical est la mise en correspondance de deux éléments lexicaux, telle qu'on peut la trouver dans des dictionnaires bilingues et qu'elle est pratiquée dans la traduction mot à mot, tandis que le transfert structural manipule des structures à un niveau d'analyse plus élevé pour rendre compte de différences d'organisation syntaxique.
12. Les listes d'arguments qui sont spécifiées dans la règle *gern-aimer* ne le sont pas dans la règle *general*, valable dans le cas général pour le transfert de l'attribut du prédicat de la tête d'une structure. En effet, les variables «Lg» et «Lf» peuvent représenter ces listes d'arguments, la relation de transfert étant définie aussi bien pour les listes, arbres, et types, que pour les simples traits <attribut valeur>.
13. Le résultat du transfert est en effet un ensemble de structures de traits, car le transfert est une relation. Ceci correspond à la notion de «traduction possible», plutôt qu'à celle de «meilleure traduction» qui ne peut être obtenue par une fonction réversible (cf. Landsbergen 1987).
14. Cet exemple ne prétend pas être une analyse des verbes modaux en allemand et en français, mais illustre la possibilité d'une analyse nécessitant une ré-entrée dans les deux langues qui soit préservée durant le transfert.
15. Cette approche contraste avec celle que l'on trouve par exemple dans le système MiMo2 (van Noord *et al.* 1990), où les ré-entrées de la structure de départ sont toujours préservées dans la structure d'arrivée.
16. La relation de spécificité est définie entre des règles unidirectionnelles et est donc déterminée quand l'ensemble bidirectionnel des règles de transfert est compilé en deux ensembles unidirectionnels partiellement ordonnés.
Les règles les plus générales du système sont les règles de transfert atomique ; pour les autres règles qui ont la même racine, soit dans L1 soit dans L2, on compare les structures mentionnées dans L1 et L2 d'après la relation de subsomption et le nombre de variables dans ces structures.
17. Dans ce cas, le module de transfert ne serait très probablement pas efficace, mais pourrait être réversible, même après compilation.

BIBLIOGRAPHIE

- BOUILLON, P. et K. BOESEFELDT (1991) : «Applying an Experimental MT System to a Realistic Problem», *Proceedings of Machine Translation Summit III*, Washington.
- CARROLL, J., BOGURAEV, B., GROVER, C. et T. BRISCOE (1988) : *A Development Environment for Large Natural Language Grammars*, Technical Report 127, Computer Laboratory, University of Cambridge.
- DYMETMAN, M. et P. ISABELLE (1988) : «Reversible Logic Grammars for Machine Translation», *Proceedings of the 2nd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Pittsburgh.
- ESTIVAL, D. (1990a) : «Generating French with a Reversible Unification Grammar», *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*, Helsinki.
- ESTIVAL, D. (1990b) : *ELU User Manual*, Technical Report, ISSCO.
- ESTIVAL, D. (1991) : «Declarativeness and Linguistic Theory», *Knowledge Modeling & Expertise Transfer*, D. Hérin-Aimé, R. Dieng, J. P. Regourd, J. P. Angoujard (Eds.), Amsterdam, IOS Press.
- ESTIVAL, D. (à paraître) : «Grammaires d'unification et traduction automatique», *Manuel de traduction automatique* (titre provisoire), P. Bouillon (Éd.).
- ESTIVAL, D., BALLIM, A., RUSSELL, G. et S. WARWICK (1990) : «A Syntax and Semantics for Feature Structure Transfer», *Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Austin.
- KAPLAN, R. et J. BRESNAN (1982) : «Lexical-functional Grammar: A Formal System for Grammatical Representation», *The Mental Representation of Grammatical Relations*, J. Bresnan (Ed.), Cambridge, MIT Press.
- KAY, M. (1983) : *Unification Grammar*, Technical Report, Xerox Palo Alto Research Center.
- KAY, M. (1984) : «Functional Unification Grammar: A Formalism for Machine Translation», *Proceedings of the 10th International Conference on Computational Linguistics (COLING)*, Stanford.
- LANDSBERGEN, J. (1987) : «Montague Grammar and Machine Translation», *Linguistic Theory and Computer Applications*, P. Whitelock, M. M. Wood, H.L. Somers, R. Johnson et P. Bennett (Eds.), London, Academic Press Limited.
- RUSSELL, G., WARWICK, S. et J. CARROLL (1990) : «Asymmetry in Parsing and Generating with Unification Grammars», *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh.
- SAG, I., KAPLAN, R., KARTTUNEN, L., KAY, M., POLLARD, C., SHIEBER, C. et A. ZAENEN (1986) : «Unification and Grammatical Theory», *Proceedings of the West Coast Conference on Formal Linguistics*, vol. 5, University of Washington, Stanford Linguistics Association.
- SAINT-DIZIER, P. (1989) : «A Generation Method Based on Principles of Government-Binding Theory», *Proceedings of the 2nd European Natural Language Generation Workshop*, Edinburgh.
- SHIEBER, S. (1986) : *An Introduction to Unification-based Approaches to Grammar*, CSLI Lecture Notes, 4., CSLI, Stanford California.
- SHIEBER, S., VAN NOORD, G., MOORE, R. C. et F. C. N. PEREIRA (1989) : «A Semantic-Head-Driven Generation Algorithm for Unification-Based Formalisms», *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver.
- VAN NOORD, G., DORREPAAL, J., VAN DER EIJK, P., FLORENZA, M. et L. DES TOMBE (1990) : «The MiMo2 Research System», *Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Austin.
- VIJAY-SHANKER, K. et A. K. JOSHI (1988) : «Feature Structure Based Tree Adjoining Grammars», *Proceedings of the 12th International Conference on Computational Linguistics (COLING)*, Budapest.