

TRAVAUX DE LINGUISTIQUE JAPONAISE (1991): *Description systématique de la grammaire japonaise*, Université Diderot (Paris 7), vol . IX, Édition L'Asiathèque, Paris

André WLODARCZYK

APPLICATION DE LA GRAMMAIRE DE MIZUTANI AU TRAITEMENT INFORMATIQUE

La description formelle de la grammaire japonaise proposée par Mizutani Shizuo (dorénavant appelée GM), en plus de ses qualités théoriques incontestables, a déjà donné lieu à son application au traitement informatique. Elle a notamment été exploitée lors de la rédaction d'une partie du dictionnaire expérimental de la langue japonaise par Yokoyama Shôichi et al., et a servi à la définition d'un langage de programmation appelé Shushin dont la syntaxe se fonde sur celle de la langue japonaise (Mizutani Shizuo et Kurokawa Toshiaki). Cela n'est pas surprenant car le linguiste japonais a pu profiter des expériences antérieures de description des phrases simples en japonais ayant pour objectif le traitement informatique. Cependant, il a pensé sa théorie grammaticale dans l'abstrait en allant au-delà des possibilités informatiques de l'époque. Ce qui constitue l'originalité de GM, ce sont moins les méthodes mathématiques employées que le retour aux sources conceptuelles de la théorie grammaticale telle qu'elle a été élaborée par les Japonais dans le passé. Pour cette raison, la différence entre GM et les grammaires formelles des langues occidentales réside surtout dans les partis pris linguistiques. En effet, Mizutani, par souci d'éviter les interprétations qui ne seraient que des calques des

modèles syntaxiques construits originellement pour les langues occidentales, tente sciemment de mettre en oeuvre les concepts de la grammaire japonaise traditionnelle.

Du point de vue linguistique, GM présente les caractéristiques suivantes:

- 1) distinction entre l'énonciation et la prédication
- 2) introduction des signes zéro
- 3) classification originale du lexique en "parties du discours"
- 4) utilisation d'un corpus d'énoncés attestés (sans recours à la compétence des sujets parlants)
- 5) conviction de la relative spécificité de la langue japonaise
- 6) intention d'élaborer une représentation qui dépasserait les structures arborescentes en signalant qu'il serait plus avantageux de les remplacer par des structures de treillis

Du point de vue de la théorie des langages, Yokoyama Shôichi a, pour sa part, mis l'accent sur les caractéristiques suivantes de GM:

- 1) elle est rédigée en forme de grammaire hors-contexte augmentée de conditions
- 2) ses règles représentent les niveaux: a) égal ou supérieur au syntagme et b) égal ou inférieur à la phrase (exceptionnellement celui de la morphologie)
- 3) la terminologie utilisée est spécifique (propre à celles des grammairiens tels que Yamada et Tokieda)
- 4) pour la plupart, les règles définissent des récursivités "à droite"
- 5) plusieurs phénomènes non traités par la grammaire sont expliqués de façon informelle.

Malgré l'ensemble important des règles qu'elle énumère, GM n'est pas une grammaire complète de la langue japonaise. Si tel était le cas, les recherches en linguistique japonaise n'auraient plus d'objectifs à atteindre. Les règles de production sont présentées par paquets. Elles se répartissent

donc en huit catégories, chacune se subdivisant encore en sous-catégories (la transcription latine utilisée est connue sous le nom de "Nihon-shiki"):

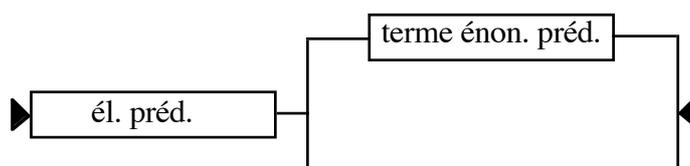
- [1] B (0-5) - Bun («phrase»)
- [2] K (0-3) - Kantai-ku («proposition exclamative»)
- [3] Z (0-34) - Zyuttai-ku («proposition prédicative»)
- [4] T (0-12 et 30-39) - Tai-rengo («syntagme nominal»)
- [5] N (0-3) - Tenka zyutsugo («prédicats transformés»)
- [6] S (0-3) - Sooren-go («syntagme qualificatif»)
- [7] Y (0-7 et 30-31) - Yooren-go («syntagme verbal»)
- [8] X (0-13) - Jookyoo-go («complément circonstanciel»)

La forme des représentations utilisées dans GM est celle des diagrammes syntaxiques. L'information que ces diagrammes contiennent peut être exprimée au moyen d'un symbolisme appelé norme de Backus-Naur (ou grammaire indépendante du contexte de Chomsky). Les conditions qui sont posées aux règles de la grammaire le sont en pseudo-Algol N.

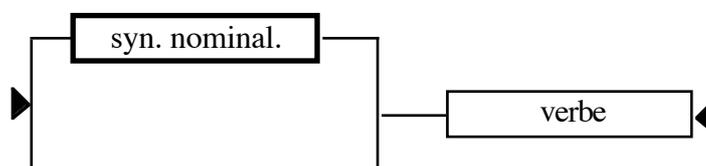
A. Un exemple d'analyseur syntaxique selon GM en PROLOG

Pour pouvoir juger de la validité de GM ainsi que pour mieux comprendre les applications de la description dont il sera question ici, nous tenterons tout d'abord de construire un petit analyseur de quelques phrases japonaises et nous montrerons comment l'exprimer au moyen du langage Prolog. Nous procéderons en reconstituant le diagramme syntaxique de l'analyseur rédigé par Maruyama N. (1989) en langage Shushin et publié par Mizutani (1989) lui-même comme exemple de sa grammaire de la langue japonaise.

Phrase:

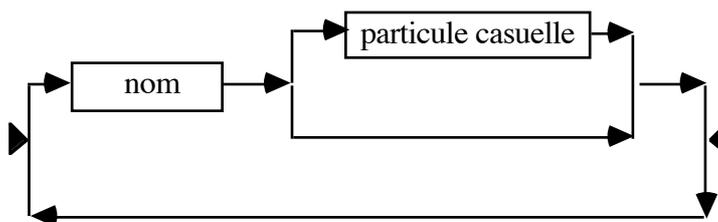


Élément prédicatif:



Le syntagme nominal est défini de manière récursive. Pour cette raison, nous sommes obligé d'utiliser les chemins fléchés (graphe orienté) plutôt que de suivre les conventions de représentation graphique de GM (ceci explique le cadre épais autour du syntagme nominal dans notre diagramme définissant l'élément prédicatif).

Syntagme nominal:



Le terme énonciatif prédicatif, le verbe, le nom et les particules casuelles sont définis comme terminaux renvoyant ainsi directement au dictionnaire.

Ainsi que l'ont démontré Pereira et Warren (1980), "tout réseau de transition est équivalent à un ensemble de simples réseaux reliés par des **push** et des **pop**" ce qu'ils appellent une *décomposition*. En décomposant les diagrammes syntaxiques des figures ci-dessus, nous obtenons une grammaire indépendante du contexte vérifiable automatiquement par une grammaire de clauses définies (Definite Clause Grammars, Pereira et Warren, 1980). Etant donné que cette dernière a été formulée dans le cadre de la programmation logique, nous avons décomposé les diagrammes

syntaxiques issus de GM en règles de grammaire de clauses définies exprimées en langage Prolog:

```
% Petit analyseur syntaxique selon la Grammaire de Mizutani (v. 0.2)
% Implémentation en Delphia-Prolog par A. Wlodarczyk (16/10/1989)
```

```
%%% REGLES SYNTAXIQUES:
```

```
ph(phrase(J1,J2)) --> el_pred(J1), t_enon_pred(J2).
ph(phrase(J1,'@E')) --> el_pred(J1).
el_pred(el_pred(D)) --> verbe(D).
el_pred(el_pred(M,'part_cas',J)) --> nom(M), el_pred(J).
el_pred(el_pred(M,K,J)) --> nom(M), part_cas(K), el_pred(J).
```

```
%%% REGLES DU LEXIQUE:
```

```
t_enon_pred(t_enon_pred(X)) --> [X], {dict(t_enon_pred,X)}.
verbe(verbe(X)) --> [X], {dict(verbe,X)}.
nom(nom(X)) --> [X], {dict(nom,X)}.
part_cas(part_cas(X)) --> [X], {dict(part_cas,X)}.
```

```
dict(t_enon_pred,rashii).
dict(verbe,chiru).
dict(verbe,saku).
dict(verbe,niou).
dict(verbe,fuku).
dict(nom,hana).
dict(nom,wakagi).
dict(nom,kaze).
dict(part_cas,ga).
dict(part_cas,ni).
```

Si nous soumettons à ces règles la phrase *Kaze ni hana ga chiru rashii*. (Il paraît que les fleurs tombent au vent), nous obtenons l'analyse suivante matérialisée par un arbre syntaxique indenté:

```
kaze ni hana ga chiru rashii.
phrase(el_pred(nom(kaze),
              part_cas(ni),
              el_pred(nom(hana),
                    part_cas(ga),
                    el_pred(verbe(chiru)))),
      t_enon_pred(rashii))
```

Dans le cas où le terme énonciatif prédicatif est un signe zéro, l'analyseur fournit l'analyse suivante. Notons que cette fois nous soumettons à l'analyseur la phrase *Kaze ni hana ga chiru*. (Les fleurs tombent au vent).

```

kaze ni hana ga chiru.
phrase(el_pred(nom(kaze),
               part_cas(ni),
               el_pred(nom(hana),
                       part_cas(ga),
                       el_pred(verbe(chiru))))),
      @E)

```

Là où GM suit bien la tradition grammaticale japonaise, c'est le fait qu'elle ne privilégie pas le sujet par rapport aux autres actants syntaxiques. Pour cette raison, la deuxième règle ci-dessus ne fait appel qu'au verbe seul. Ainsi, en analysant la phrase *Chiru.* ([ça] tombe.), le terme énonciatif prédicatif est un signe zéro, comme il fallait s'y attendre, mais il n'est pas nécessaire de poser un sujet zéro.

```

chiru.
phrase(el_pred(verbe(chiru))),
      @E)

```

Les deux dernières règles de notre mini-analyseur permettent de reconnaître la structure syntaxique des phrases composées aussi bien d'un prédicat (avec un terme énonciatif prédicatif zéro) que d'un actant (suivi ou non d'une particule).

```

hana chiru.
phrase(el_pred(nom(hana),
               part_cas,
               el_pred(verbe(chiru))),
      @E)

```

```

hana ga chiru.
phrase(el_pred(nom(hana),
               part_cas(ga),
               el_pred(verbe(chiru))),
      @E)

```

```

kaze ni chiru.
phrase(el_pred(nom(kaze),
               part_cas(ni),
               el_pred(verbe(chiru))),
      @E)

```

Le petit analyseur ci-dessus, quoique conforme à celui de Maruyama, présente quelques inadéquations par rapport aux règles de GM quant au traitement de certains signes zéro liés à l'analyse morphologique (conjugaison) ainsi qu'à la profondeur de la structure des arbres dérivés

(trait caractéristique important de GM face aux grammaires occidentales). Rappelons cependant que la théorie syntaxique de GM commence par définir l'énoncé en tant qu'un constituant prédicatif (élément prédicatif - jutsu-so) suivi d'un constituant énonciatif (terme énonciatif prédicatif - jut-tai-ji) sans que la structure SN + SV ou Sujet + Prédicat soit posée à la base de la grammaire.

B. Implémentation de GM en LINGOL

GM a été implantée dans sa totalité en langage Lingol par Yokoyama Shôichi et al. en 1987 à l'Institut Général de Technologies Electroniques de Tsukuba. Dans cette implémentation, GM constitue le noyau grammatical d'un système de traitement de la langue japonaise qui utilise un dictionnaire de grande taille. Afin de rendre un tel système efficace, ses concepteurs distinguent entre deux niveaux de mémoire: mémoire primaire (immédiatement accessible - mémoire vive) et mémoire secondaire (accessible par tables d'adressage dispersé (hash tables) - mémoire sur support magnétique). Actuellement, le système n'utilise que le premier niveau du dictionnaire.

La raison pour laquelle GM a été choisie comme pivot de ce système est la suivante: étant donné que la structure des arbres dérivés au cours de l'analyse est plus profonde que dans d'autres descriptions grammaticales du japonais, l'ambiguïté syntaxique y est moins importante. C'est notamment cette caractéristique de GM que nous allons expliquer. Soit, par exemple, l'énoncé (sans sujet) suivant: *Ikanai.* ([qn] n'ira pas.). L'analyse syntaxique classique aboutirait probablement à la structure arborescente (Fig. 1) où chaque noeud (à l'exception évidemment des noeuds pré-terminaux) serait binaire:

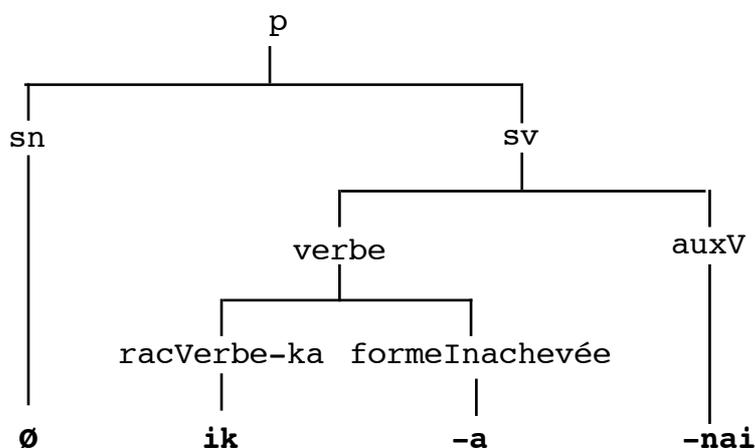


Fig. 1. Exemple d'arbre dérivationnel "classique".

L'analyse du même énoncé par les règles de GM permet cependant de produire un arbre dérivationnel "plus profond" (Fig. 2) qui, de ce fait, est plus riche en information syntaxique.

En effet, en admettant un grand nombre de noeuds qui ne possèdent chacun qu'un seul descendant, les règles syntaxiques produisent un nombre restreint d'arbres dérivationnels, ce qui évite de poser de multiples restrictions sémantiques pour empêcher de générer un grand nombre d'interprétations ambiguës. Ainsi, la perte de l'efficacité au niveau syntaxique n'est probablement qu'apparente.

L'implémentation de GM par Yokoyama et al. a contribué à améliorer la description de la grammaire japonaise. Nous avons été témoin d'une démonstration de cette implémentation au cours de laquelle le système arrivait à tracer les arbres dérivationnels corrects d'énoncés parfois même assez complexes. De plus, le dictionnaire total contient actuellement 130.000 mots dont certains (comme, par exemple, les particules, les formes grammaticales et les verbes) sont accessibles directement par l'analyseur, mais la grande majorité des entrées est logée dans la mémoire secondaire (tel est le cas des substantifs dont la quantité atteint près de 80% du vocabulaire). Il faut ajouter qu'aussi bien les règles grammaticales que les dictionnaires sont implantés directement en *kanji*, ce qui rend

possible le traitement des textes japonais tels quels et facilite grandement la maintenance du système.

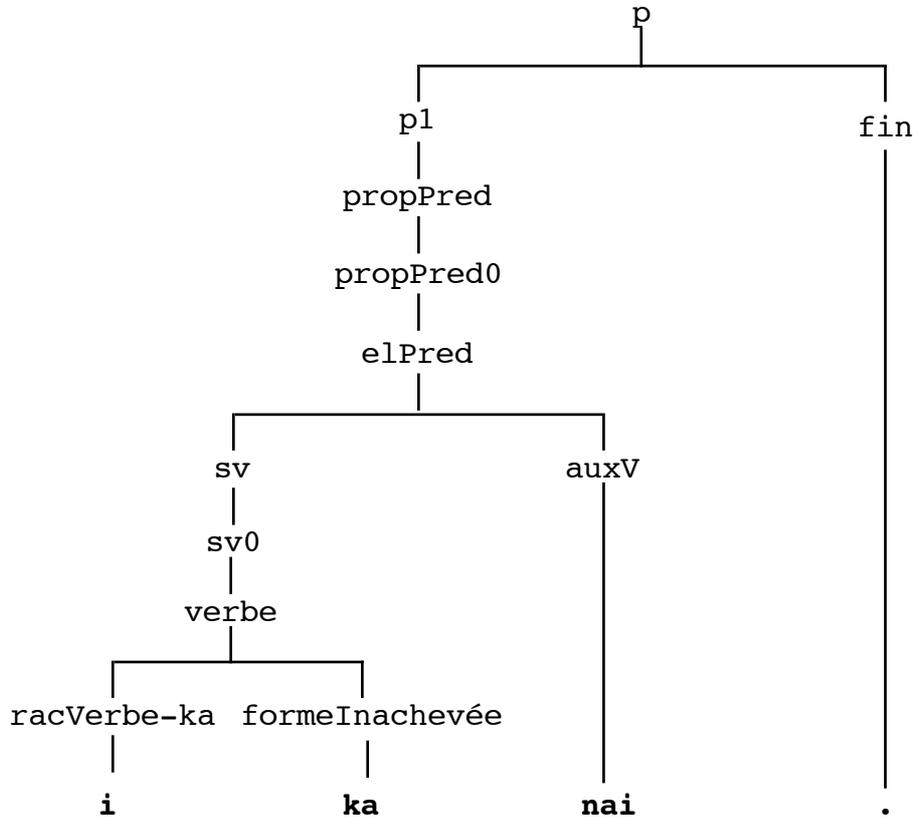


Fig. 2. Exemple d'arbre dérivationnel selon les règles de GM.

Cependant, les auteurs de l'implémentation de GM sur ordinateur ont signalé également quelques difficultés qu'ils ont rencontrées au cours de leur travail. Il s'agit tout particulièrement du traitement des sept signes zéro (difficile à programmer en Lingol) et de certaines parties conditionnelles de la grammaire (causes des répétitions à l'infini - "boucles sans fin"). D'autre part, le système rencontre quelques problèmes au niveau du dictionnaire, et notamment:

- 1) les variantes orthographiques (dues à "mazegaki" - écriture mélangeant les caractères kanji et kana), très nombreuses en japonais, n'y sont pas spécifiées

- 2) la partie secondaire du vocabulaire occasionne quelques difficultés au cours de l'analyse (certains mots fréquemment rencontrés doivent être transférés dans la mémoire vive)
- 3) étant donné que tous les mots du dictionnaire secondaire sont considérés comme des noms, il faut modifier celui-ci en précisant davantage leurs qualités en tant que parties du discours

On constate également que l'analyse qui se fonde sur un double dictionnaire pose des problèmes insurmontables tant il est vrai que le japonais se sert d'une écriture continue (sans délimiteurs de mots) et que le système utilise la méthode de reconnaissance "des plus longues chaînes". Ceci empêche tout retour en arrière (réapplication des mêmes règles) au moment où on doit accéder à la mémoire secondaire. Pour cette raison, une méthode spécialement conçue pour le traitement du japonais (Motoyoshi Fumio et al.) sera utilisée à l'avenir.

C. Application de GM à la «conversion» des honorifiques

Tanaka Sachiko [1983] a présenté une expérience computationnelle de «conversion honorifique». Son expérience de formalisation des honorifiques japonais s'appuie sur la théorie que Mizutani Shizuo [1983] a développée pendant ses cours à l'Université de Jeunes Filles de Tokyo (Tôkyô Jôshi Daigaku) comme extension de GM. Selon cette théorie, l'acte langagier peut être vu tantôt du côté du locuteur, il s'agira alors de l'acte d'expression, tantôt du côté de l'allocutaire, dans ce dernier cas, on parlera de l'acte de réception. Ainsi, tout acte langagier s'appuie sur trois supports que, en se plaçant du point de vue de l'acte d'expression, on peut définir de la façon suivante:

- sujet parlant (celui qui effectue l'acte d'expression) - **I**
- interlocuteur (celui vers qui l'acte d'expression est dirigé) - **I'**
- autrui (celui à qui le sujet parlant se réfère) - **P,Q**

Ensuite la théorie de Mizutani distingue entre les trois niveaux honorifiques suivants:

- le **niveau 1** ne concerne que les rapports entre I et I' qu'expriment les éléments énonciatifs (chinjutsu-so), par exemple:

Yoi tenki desu .

(Il fait beau.)

- le **niveau 2** concerne les rapports de I avec P et Q. Ces rapports sont exprimés aussi bien par les éléments prédicatifs (jutsu-so) que par les termes renvoyant directement à P et/ou à Q, par exemple:

Tarô wa Hanako-sama ni môshi-ageta.

(Tarô l'a dit à Madame Hanako)

ou bien

Tarô-sama ga Hanako ni hon wo yonde kudasatta.

(Monsieur Tarô a lu un livre à Hanako)

- le **niveau 3** ne concerne que les rapports entre P et Q. Ces rapports sont exprimés par les éléments prédicatifs (jutsu-so), par exemple:

Tarô wa Hanako ni hon wo ageta.

(Tarô a lu un livre à Hanako)

Les rapports que les quatre termes (I, I', P et Q) sont susceptibles d'entretenir entre eux s'organisent, avant tout, de façon hiérarchique. Par voie d'hypothèse, cette hiérarchie distingue trois relations:

$a > b$: a est supérieur à b

$a < b$: a est inférieur à b

$a = b$: a est égal à b

Tanaka S. considère qu'afin de générer un énoncé contenant des éléments d'honorification, il est nécessaire d'effectuer une série d'opérations appelées "conversion des termes honorifiques". Parmi les opérations théoriquement possibles ($3^6 = 729$) seules 75 ont été retenues. Une convention supplémentaire a été adoptée selon laquelle les termes soulignés entre cro-

chets signifient que leur relation hiérarchique avec le terme mis en rapport grâce aux signes $<$, $>$ et $=$ n'est pas pertinente pour la conversion donnée. Par exemple: Dans la notation $[a, b, c] < d$, seul compte le rapport d'infériorité entre c et d .

Les tableaux 1, 2, et 3 illustrent les correspondances entre les conditions à appliquer au cours du traitement et les expressions linguistiques possibles:

Condition	Expression possible
$I > I'$	-ta
$I \leq I'$	-mashita

Tab. 1: Règles de répartition des formes énonciatives

Condition	Expression possible
$[P, Q, \underline{I}] < I$ ou $P=Q=I=I'$	tsurete iku
$[I, I', Q] < P$ sauf $I < I' < P=Q$	tsurete irassharu
$[\underline{I}, P] < P$ ou $[P, \underline{I}] < Q=I$	o-tsure suru
$[I, I'] < P < Q$ sauf $I=I' < P=Q$ et $I' < I < P=Q$	o-tsure ni naru
$[I, P, Q] < I'$	tsurete mairu
$[I, P] < I' > Q$ mais $P < I < Q=I'$	tsurete itasu

Tab. 2: Règles de répartition des éléments prédicatifs

Condition	Expression possible
$\alpha < I$	---
$I < \alpha < I'$ ou $I = \alpha$	-san
$[I, \underline{I}] < \alpha$	-sama

Tab. 3: Règles de répartition des formes relatives aux "objets"

Le symbole α représente les "objets" humains du discours (P ou Q). De manière générale, "on appelle par son nom tout court l'objet α qui est égal au locuteur" (c.-à.-d.: quand $I=\alpha$). Selon Tanaka, cette observation ne tiendrait pas dès lors que le sujet parlant serait une femme et α n'appartiendrait pas au même groupe social qu'elle car, dans ce cas, elle devrait utiliser les termes à suffixe -san ou -sama. Au cas où $P=I$, le locu-

teur a le choix entre *watakushi* (quand $I < I'$) et *boku* (quand $I \geq I'$) pour s'indiquer soi-même, tandis que si $Q=I'$, le locuteur a le choix entre *anata* (quand $I < I'$) et *kimi* (quand $I \geq I'$) pour indiquer son interlocuteur.

Afin d'introduire l'information relative aux conditions de conversion (c.-à.-d.: l'information concernant les rapports hiérarchiques entre les actants de l'énoncé et/ou les participants de l'énonciation), l'auteur de l'expérience a défini le quadruplet de valeurs $\{I, I', P, Q\}$ où P et Q représentent en plus l'information concernant le sujet et l'objet de l'énoncé respectivement.

L'ordinateur détermine le rapport hiérarchique entre deux éléments facultatifs selon l'opération de soustraction sur les valeurs numériques qui ont été attribuées à ces derniers. Ainsi,

- si $I' - I > 0$, alors I' est supérieur à I
- si $I' - I = 0$, alors I' est égal à I
- si $I' - I < 0$, alors I' est inférieur à I

Ces rapports hiérarchiques sont donc déterminés de façon relative suivant l'ordre de grandeur des quatre valeurs fournies. Les nombres qui occupent la troisième et la quatrième positions (P et Q) se réfèrent aux deutéragonistes de l'énoncé: le sujet (P) et l'objet (Q) de la phrase respectivement. Etant donné qu'aucun de ces éléments n'est obligatoire dans la phrase japonaise, dans le premier cas, par exemple: *gakkô ni itta* (<[qn] est allé à l'école>), on introduira la valeur du rapport hiérarchique du sujet non exprimé à la position de P, tandis qu'on entrera "N" dans le cas d'un énoncé sans objet.

En vue d'obtenir la conversion honorifique des énoncés neutres, Tanaka a été obligée de préciser les rapports de succession entre les classes de mots et les diverses formes morphologiques du japonais. Le traitement morphologique lui-même ne serait pas possible sans l'analyse syntaxique dont le rôle principal est de déterminer l'appartenance de chaque mot à l'une des catégories suivantes (toutes ces catégories font partie de GM):

- élément énonciatif - cet élément concerne surtout le premier niveau du traitement honorifique
- élément prédicatif - cet élément peut être composé d'une suite de n éléments prédicatifs (par exemple: morphème \emptyset , *ni*, *te*, *de*, *tari*, *dari*, *dattari* ainsi que leurs combinaisons)
- "objets": S (pour sujet) et O (pour complément d'objet direct)
- "modificateur": tout terme qui joue le rôle d'adverbe ou "modifie" le verbe, par exemple: *kinô* (hier).

Les résultats de l'analyse syntaxique sont à leur tour utilisés par les procédures de conversion honorifique. Ainsi, par exemple, selon le jeu de conditions, les énoncés générés par l'algorithme de Tanaka sont les suivants:

- 1,0,0,0 - Hanako wa yasashii kao wo shite iru.
- 1,0,1,1 - Hanako-san wa yasashii kao wo shite iru.
- 1,1,1,1 - Hanako-san wa yasashii kao wo shite imasu.
- 1,2,1,1 - Hanako-san wa yasashii kao wo shite orimasu.
- 1,1,2,2 - Hanako-san wa yasashii o-kao wo shite irasshaimasu.

(Hanako fait bonne mine.)

- 1,1,1,1 - Tarô wa Hanako wo kinô eiga ni tsurete ikimashita.
- 2,1,2,0 - Tarô-san wa Hanako wo kinô eiga ni tsurete itta.
- 2,1,3,2 - Tarô-sama wa Hanako-san wo kinô eiga ni tsurete irasshatta.
- 2,1,1,3 - Tarô wa Hanako-sama wo kinô eiga ni o-tsure shita.
- 2,1,1,2 - Tarô wa Hanako-san wo kinô eiga ni o-tsure shita.
- 1,1,2,3 - Tarô-sama wa Hanako-sama wo kinô eiga ni o-tsure ni narimashita.
- 1,3,2,2 - Tarô-san wa Hanako-san wo kinô eiga ni tsurete mairishita.
- 1,2,1,3 - Tarô-sama wa Hanako-sama wo kinô eiga ni o-tsure itashimashita.
- 2,3,1,3 - Tarô wa Hanako-sama wo kinô eiga ni tsurete mairishita.
- 2,1,3,3 - Tarô-sama wa Hanako-sama wo kinô eiga ni tsurete irasshatta.

(Hier, Tarô a accompagné Hanako au cinéma.)

Du point de vue théorique, il n'est pas sûr, selon nous, qu'il faille poser le problème de la hiérarchie sociale (supérieur/inférieur) au coeur du système honorifique. Cela en raison des difficultés à établir des correspondances univoques entre les formes honorifiques et les critères sociaux. En réalité, à notre avis, si le système de Tanaka est opérationnel tel quel, c'est

parce qu'il est possible de réinterpréter la distinction de hiérarchie sociale par le trait sémique \pm Honorifique. Il va sans dire que le sujet parlant doit du respect, entre autres, aux personnes qui lui sont hiérarchiquement supérieures, mais cela n'est pas la seule raison. On pourrait énumérer plusieurs autres critères référentiels (tels que l'âge ou le sexe), pragmatiques (la volonté de se conformer à certaines maximes: tact, générosité, approbation, modestie, consentement et sympathie cf. Leech - 1983) ou même rhétoriques (par exemple, l'ironie). Il est indéniable que la prise en compte de tels phénomènes contribuerait à une meilleure compréhension de l'honorification mais cela nécessiterait d'intégrer, dans les programmes, des modules capables de résoudre les problèmes qui relèvent de la pragmatique et de la connaissance commune de l'univers (cf. Włodarczyk - 1988). Malgré cela, les résultats obtenus présentent beaucoup de points positifs, d'autant plus que l'auteur de l'expérience a utilisé un langage de programmation (BASIC) qui ne lui permettait pas de raffiner son analyse. Il est indéniable que le travail aurait gagné en clarté si Tanaka avait utilisé un langage de programmation plus performant. Néanmoins, d'une manière générale, l'utilisation de GM par Tanaka montre que les capacités d'extension de cette grammaire sont loin d'être épuisées.

D. SHUSHIN - un langage de programmation en japonais

En utilisant sa description formelle de la syntaxe japonaise, Mizutani Shizuo a pu définir un langage de programmation procédural dont la particularité consiste avant tout en ce que son lexique aussi bien que sa syntaxe sont japonais. La nécessité d'une meilleure description des règles grammaticales de la langue japonaise s'imposait tant il est vrai que les tentatives précédentes de définition des langages de programmation utilisant le lexique et la syntaxe de cette langue en violaient trop souvent les règles syntaxiques produisant ainsi une impression quelque peu singulière. Par exemple, l'expression de la condition en langage MIND commençait par le prédicat (copulatif) «naraba» (lit.: "si [cela est que]"), alors que la syntaxe japonaise réserve pour le prédicat la place finale dans la phrase. Ainsi, la

forme complète de la condition prenait la forme "naraba X=Y" au lieu de "X=Y naraba".

La première version du langage de Mizutani appelée Sho-Shushin, fut écrite en 1984, en collaboration avec Kurokawa Toshiaki. Sa version définitive est connue sous le nom Shushin et a vu le jour en juillet 1989. Shushin a été spécialement conçu pour le traitement des listes. Mais au vu seul de la documentation, le langage Shushin ne semble pas apporter de solutions réellement novatrices du point de vue informatique. Cependant, en plus de sa syntaxe bien japonaise, ce langage réunit les deux caractéristiques suivantes:

- (1) le traitement des listes (de symboles) hérité des langages SNOBOL et LISP
- (2) le traitement des mélanges de caractères (kanji, kana et alphabet latin)

La subroutine ci-dessous qui est tirée de l'analyseur syntaxique proposé par Maruyama permettra d'apprécier la qualité de la langue japonaise utilisée dans ce langage. Le rôle de cette subroutine est de vérifier si l'unité linguistique traitée est un terminal (mot), dans quel cas, on fait précéder cette unité du caractère @.

```
*shitauke [Match] de @Awase:
  shiwake
  .. SakiYomi = Match nara /\_Match wo Kôzô no sue ni utsuse,
     Bun wo #SakiYomi\_ \_#Bun ni wakate;
     hoka wa [/\$/\_Match\_/\$\\\_/ga mistukaranai\] de @tsûhō .:
o:α
```

En traduction française, cette subroutine prendrait la forme suivante:

```
*subroutine @Comparer avec [Filtre]:
  condition
  ..si Tête = Filtre alors mettre " @" et Filtre en fin de
     Structure,
     décomposer Phrase en #Tête et #Phrase;
     sinon @répondre avec [Filtre et espace et "est
     introuvable"] .:
fin : retour
```

La traduction de cette procédure en un langage de programmation existant en Occident (tel que HyperTalk, par exemple) est quasiment

immédiate. Cela s'explique parce que le traitement de structures de données textuelles intégré dans Shushin est suffisamment complet et que, de par sa définition syntaxique, HyperTalk également est un langage artificiel dont les structures sont proches de celles d'une langue naturelle - l'anglais.

```
on Compare
  Global Match, Head, Sentence, Structure
  if Head is Match then
    put " @" & Match after Structure
    put first word of Sentence into Head
    delete first word of Sentence
  else
    put Match & space & "cannot be found." into message window
  end if
end Compare
```

Il est clair que l'intérêt essentiel du langage Shushin est d'être fondé sur la syntaxe du japonais alors que les autres langages de programmation (même traduits en japonais) se fondent sur la syntaxe de l'anglais. Rappelons que le petit analyseur de Maruyama a été réalisé afin de montrer comment on peut utiliser le langage Shushin pour écrire des analyseurs syntaxiques du japonais. Du point de vue informatique cependant, malgré ses qualités linguistiques incontestables, Shushin est un langage procédural et se prête moins bien à l'écriture de grammaires que Prolog (langage déclaratif). Ce dernier, en effet, permet de formuler des grammaires-programmes capables à la fois d'analyse et de synthèse. Mentionnons tout de même en passant que, dans l'exemple de l'analyseur de Maruyama, le traitement des rapports casuels est récursif et que notre traduction en Prolog ne lui procure pas non plus la qualité d'être également un générateur. A ce propos, il convient d'observer qu'en présentant sa grammaire, Mizutani (cf. Hypothèse 5) a surtout développé une théorie de la **formation** des phrases (bun-sosei-ron, cf. la récursivité - ireko ou atama-yama) mais non une théorie de la **composition** des phrases (bun-seiritsu-ron, cf. la co-occurrence - kakari-musubi). Cependant, il est à noter que, pour rendre compte des rapports casuels, la récursivité au sens étroit n'est probablement pas nécessaire tant il est vrai que les prédicats des langues naturelles ne peuvent régir qu'un nombre restreint d'actants (dépassant rarement quatre ou cinq).

*
* *

L'intérêt de présenter une grammaire de façon formelle ne réside cependant pas seulement dans la possibilité de l'exprimer au moyen de différents langages artificiels (d'autant plus que de nombreux problèmes se posent, bien évidemment, à propos de l'adéquation pratique des langages artificiels choisis aux questions traitées). Le fait de pouvoir écrire la même grammaire au moyen de tel ou tel langage artificiel importe peu ici. Ce que nous aimerions souligner, ce sont les options retenues par le grammairien dans la description d'une langue donnée car, a priori, il ne nous paraît pas totalement impossible de prendre pour axiome la définition de la phrase donnée par Mizutani pour décrire le français, mais la grammaire ainsi obtenue risquerait d'être exagérément complexe dès le départ.

Au terme des recherches menées dans le domaine de l'intelligence artificielle, les langages de programmation se rapprocheront des langues naturelles. Cependant, les difficultés que les Japonais ont éprouvé à définir leur propre langage informatique témoignent de ce que les recherches en linguistique formelle du japonais ne viennent que de débiter, et cela malgré une longue tradition grammaticale qui a été pendant trop longtemps ensevelie sous les apports de la linguistique occidentale. Il a fallu trois décennies de recherches intenses en informatique pour que les Japonais se donnent les moyens d'une meilleure utilisation de leur propre langue pour construire des langages de programmation. Aux yeux du linguiste, cet événement est d'autant plus important que la langue japonaise présente une structure ("superficielle") très différente de celle des langues indo-européennes.

La tentative de description globale de la grammaire japonaise traditionnelle du point de vue formel réalisée par Mizutani est un événement sans précédent dans l'histoire de la linguistique générale et, en

tant que tel, nous ouvre des horizons sur les univers linguistiques dits exotiques. Dans ce contexte, même si l'on admet que la distinction entre les langues naturelles (modèles formels) et les langues humaines (systèmes de communication) risque de ne jamais s'effacer, la voie ouverte mérite d'être poursuivie.

André WŁODARCZYK

REFERENCES BIBLIOGRAPHIQUES

- [1] APPLE COMPUTER France, Guide du langage HyperTalk™, Paris, 1988.
- [2] LEECH N. Geoffrey: Principles of Pragmatics, Longman, London and New York, 1983.
- [3] MARUYAMA N.: TDP (top down parser) dans «Kokugofû shidai-date gengo - shushin no tebiki» (Manuel du langage de programmation en japonais Shushin), p. 74, Tôkyô-Joshi-Daigaku Nihon-Bungakka, Tokyo 1989
- [4] MIZUTANI Shizuo: Koku-bumpô sobyô (Description systématique de la grammaire japonaise), dans «Asakura Nihongo Shin-Kôza», vol. 3, (ouvrage collectif) Asakura Shoten, Tokyo, 1983.
- [5] MIZUTANI Shizuo: Kantai-bun Shakugihô (Grammaire des phrases exclamatives), dans «Mathematical Linguistics», Vol 16, n° 7, pp. 281-312, Tokyo 1988
- [6] MIZUTANI Shizuo: «Kokugofû shidai-date gengo - shushin no tebiki» (Manuel du langage de programmation en japonais Shushin), Tôkyô-Joshi-Daigaku Nihon-Bungakka, Tokyo 1989
- [7] MIZUTANI Shizuo: Taigû-hyôgen no shi-kumi (Les Mécanismes des expressions honorifiques), dans «Asakura Nihongo Shin-Kôza», vol. 5, (ouvrage collectif) Asakura Shoten, Tokyo 1983.
- [8] MOTOYOSHI Fumio, ISAHARA Hitoshi et ISHIZAKI Shun: Nihongo-yô kanzen yoko-gata shinsaku kôbun kaiseki-hô, dans «Dai 32-kai Jôhō shori gakkai zenkoku taikai rombunshû» (Techniques pour l'analyse complète "en largeur d'abord" du japonais), 4 S-3, 1986

- [9] NISHIMURA Hirohiko, MIZUTANI Shizuo, ONOE Keisuke et ONO Mieko: Nihongo kihon bumpô - fukubun-hen (Grammaire fondamentale du japonais - la phrase complexe), dans «Researches of the Electronical Laboratory», n° 784, Tokyo 1978
- [10] NISHIMURA Hirohiko, MIZUTANI Shizuo, ONOE Keisuke et TANAKA Sachiko: Nihongo kihon bumpô - tambun-hen (Grammaire fondamentale du japonais - la phrase simple), dans «Researches of the Electronical Laboratory», n° 783, Tokyo 1978
- [11] PEREIRA F. C. N. et WARREN D. H. D.: Definite Clause Grammars for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks, dans «Artificial Intelligence» 13 (1980), pp. 231-278, North-Holland Publishing Company, 1980
- [12] TANAKA Sachiko: Keigo wo totonoeru (Traitement des formes honorifiques), dans «Asakura Nihongo Shin-Kôza», vol. 5, (ouvrage collectif) Asakura Shoten, Tokyo, 1983.
- [13] WŁODARCZYK André: Les traits pertinents du système honorifique japonais - une tentative d'implémentation en Prolog, communication au 5e Congrès de l'EASJ, Durham - 1988, dans «European Studies in Japanese Linguistics 1988-90», Lone Publications, London, 1991 (pp. 127-150).
- [14] YOKOYAMA Shôichi, MOTOYOSHI Fumio & ISAHARA Hitoshi: A Natural Language Processing System with a Large Vocabulary in Secondary Storage, dans «Bulletin of the Electro-Technical Laboratory», Vol. 53, No. 5, 1989, (pp. 52-70)
- [15] YOKOYAMA Shôichi, MOTOYOSHI Fumio & ISAHARA Hitoshi: Niji-Kiyokujô no dai-kibo go'i wo mochiiru shizen-gengo shori shisutemu (Un système de traitement du langage naturel muni

d'un lexique de grande taille enregistré en mémoire secondaire), dans «JôHô-Shori Gakkai Rombunshi», Vol. 29, No. 6, 1988

- [16] YOKOYAMA Shôichi: Kokugo-jiten kara no dai-kibo go'i wo haikai to shita kôbun-kaiseki shisutemu (Un système d'analyse syntaxique fondé sur un lexique de grande taille extrait du dictionnaire de la langue japonaise), dans «Denshika-jisho ni motozuku nihongo no keisanki gengogaku-teki kenkyû», Dai 881 gô, 1987