Recherche et identification automatiques des morphèmes en japonais

Hubert Dauphin

N° d'ordre :

THÈSE

présentée à

LA FACULTÉ DES SCIENCES DE L'UNIVERSITÉ DE GRENOBLE

pour obtenir

LE TITRE DE DOCTEUR DE TROISIÈME CYCLE

Mathématiques Appliquées

---

par

Hubert DAUPHIN

Licencié ès Sciences

RECHERCHE ET IDENTIFICATION AUTOMATIQUES DES MORPHÈMES EN JAPONAIS

Thèse soutenue le 10 Novembre 1962, devant la Commission d'examen :

Monsieur J. KUNTZMANN, Président

Messieurs B. VAUQUOIS, Examinateurs

N. GASTINEL

T H E S E

présentée à

LA FACULTE DES SCIENCES DE L'UNIVERSITE DE GRENOBLE

pour obtenir

le titre de Docteur de Troisième cycle

MATHEMATIQUES APPLIQUEES

par

Hubert DAUPHIN

Licencié-ès-sciences

RECHERCHE ET IDENTIFICATION AUTOMATIQUES DES MORPHEMES EN JAPONAIS

Thèse soutenue le 10 novembre 1962 devant la commission d'examen

MM. J. KUNTZMANN Président
B. VAUQUOIS
N. GASTINEL Examinateurs.

FACULTE DES SCIENCES DE L'UNIVERSITE DE GRENOBLE

Doyens honoraires :

M. FORTRAT P.

M. MORET L. , Membre de l'Institut

Doyen :

M. WEIL L.

Professeurs :

MM. NEEL L. , Membre de l'Institut - Physique expérimentale
MOREL L. Géologie et minéralogie
WOLFERS F. - Physique
DORIER A. - Zoologie
HEILMANN R. - Chimie organique
KRAVTCHENKO J. - Mécanique rationnelle
PARDE M. - Potamologie
BENOIT J. - Radioélectricité
CHENE M. - Chimie papetière
NOBECOURT P. -Micrographie papetière
BESSON J- Chimie
WEIL L. - Thermodynamique
FELICI N. - Electrostatique
KUNTZMANN J. - Mathématique Appliquées
BARBIER R. - Géologie appliquée
SANTON L. - Mécanique des fluides
CHABAUTY C. - Calcul différentiel et intégral
OZENDA P. - Botanique
FALLOT M. - Physique industrielle
GALVANI O. - Mathématiques
MOUSSA A. - Chimie nucléaire et radioactivité
TRAYNARD P. - Chimie générale
CRAYA A. - Hydrodynamique
SOUTIF M. - Physique générale
REEB G. - [illegible] Mathématiques
REULOS R. - Théorie des champs
AYANT Y. - Physique approfondie
GALLISSOT F. - Mathématiques pures
Melle LUTZ E. - Mathématiques générales
MM. BLAMBERT M. Mathématiques
BOUCHEZ R. - Physique nucléaire
LLIBOUTRY L. - Géophysique
MICHEL R. - Géologie et minéralogie
BONNIER E. - Electrochimie
VAUQUOIS B. - Calcul Electronique

Professeurs sans chaire :

MM. SILBER R. - Mécanique des fluides
DESSAUX G.- Physiologie animale
MOUSSIEGT J. - Electronique
PILLET E. - Electrotechnique
BARBIER J.C.- Physique
BUYLE-BODIN M. - Electronique
PAUTHENET R. - Electrotechnique
Mme KOFLER L. - Botanique

Maitres de Conférences :

MM. VAILLANT F. - Zoologie et hydrobiologie
DREYFUS B. - Thermodynamique
Melle NAIM L. - Mathématiques
MM. PERRET R. - Servomécanisme
ARNAUD P. - Chimie
Mme BARBIER M.J. - Electrochimie
MM. BRISSONNEAU P. - Physique
COHEN J. - Physique
DEBELMAS J. - Géologie et minéralogie
Mme SOUTIF J. - Physique

MM. DEPASSEL R. - Mécanique des fluides
GERBER R. Mathématiques
ROBERT A. - Chimie papetière
ANGLES D'AURIAC - Mécanique des fluides
BIAREZ - Mécanique physique
COUMES A. - Electronique
DODU J. - Mécanique des fluides
DUCROS P. - Minéralogie et cristallographie
GIDON P. - Géologie et minéralogie
GLENAT R. - Chimie
HACQUES G. - Calcul numérique
LANCIA R. - Physique automatique
PEBAY-PERCULA- Physique
GASTINEL - Chargé d'enseignement - Mathématiques appliquées
LACAZE A. - Chargé d'enseignement - Thermodynamique.

Je tiens à exprimer ma profonde reconnaissance

à Monsieur le Professeur KUNTZMANN, Directeur du Service de Mathématiques Appliquées de Grenoble qui a bien voulu me faire l'honneur de présider le Jury

à Monsieur le Professeur VAUQUOIS, pour les conseils judicieux qu'il m'a donnés tout au long de mon travail et pour le temps qu'il y a consacré

à Monsieur GASTINEL, Maître de Conférences qui a bien voulu faire partie du Jury.

Mes remerciements s'adressent également aux membres du Centre d'Etudes pour la Traduction Automatique de Grenoble et notamment

à Madame YAMADA pour l'aide et les nombreuses indications qu'elle m'a données sur la langue japonaise

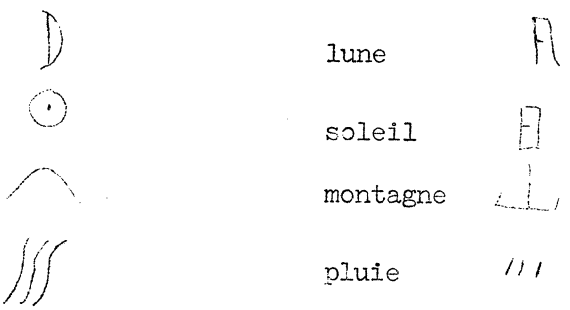
à Mademoiselle M. BOUVIER pour la réalisation matérielle de cette thèse.

INTRODUCTION A LA MORPHOLOGIE DU JAPONAIS

-I- HISTORIQUE

Avant le III ème siècle, les japonais n'avaient pas de système d'écriture propre. La littérature était uniquement de tradition orale. Ils utilisèrent alors les caractères chinois appelés "KANJIS" ou "IDEOGRAMMES". Chaque idéogramme représentait alors un mot chinois différent.

Comme dans beaucoup de langues dans l'antiquité, l'idéogramme représentait une action ou une pensée. Les chinois avaient commencé à décrire leur pensée d'abord par des images primitives qu'ils stylisèrent ensuite avec le temps



puis en combinant des caractères simples ils obtinrent un choix innombrables d'idéogrammes

Il va de soi qu'il n'était pas facile d'utiliser le système d'écriture chinois pour une langue aussi différente que le japonais, d'une part parce que tous les mots japonais n'avaient pas leur équivalent en chinois, d'autre part parce que le japonais est une langue fléchie à l'encontre du chinois.

Les idéogrammes empruntés étaient utilisés tantôt selon leur signification, tantôt suivant leur son et bien souvent c'était dans la phrase écrite une cause d'ambiguité car on ne savait pas si l'idéogramme conservait son sens.

Les japonais éprouvèrent la nécessité de créer des caractères à transcription phonétique monosyllabique : " LES KANAS " permettant de mieux épouser leur langue et leur façon de penser.

-II- ALPHABET

Actuellement l'alphabet japonais est composé de trois sortes de caractères:

- LES IDEOGRAMMES : d'origine chinoise (1800 environ)
- LES HIRAGANAS : créés par les Japonais (50 environ)
- LES KATAKANAS : créés également par les japonais (50 environ) utilisés uniquement pour les mots d'origine étrangère.

Les deux derniers sont appelés "KANAS" et ont une transposition phonétique monosyllabique. En écriture romaine le kana est représenté par deux lettres. Une voyelle et une consonne en général.

-III- LA PHRASE

D'une façon générale désignons par $\overset{l}{\underset{i=1}{T}}(S_i)$ la concaténation de l symboles :

$$S_1, S_2 \ldots\ldots\ldots\ldots S_{l-1}\, S_l$$

Si pouvant être une caractère simple, c'est-à-dire une lettre, mais pouvant aussi désigner un mot japonais, c'est-à-dire une suite de caractères simples, ou pouvant avoir toute autre signification que l'on voudra bien lui donner.

En représentant l'idéogramme par le symbole I
le kana " " K_n
L'hiragana " " H
le katakana " " K

tout signe de ponctuation ne marquant pas la fin d'une phrase par le symbole $\mathcal{T}$

et en appelant :

$\langle$suite primaire$\rangle$ ::= T(H)/T(I)T(H)/T(K)T(H)

la phrase à l'aspect suivant

$\langle$Phrase$\rangle$ ::= $\langle$T (suite primaire)$\rangle$ $\langle$signe de ponctuation de fin de phrase$\rangle$T(suite primaire)$\rangle$ $\langle \mathcal{T} \rangle$ $\langle$Phrase.$\rangle$

Autrement dit une phrase en japonais se présente sous la forme d'une suite alternée d'idéogrammes et de kanas, séparée le cas échéant par des signes de ponctuation.

Entre les mots les japonais laissent le même espace qu'entre les caractères, le résultat est le même que si tous les mots étaient attachés les uns à la suite des autres.

Le but du découpage est de substituer à la segmentation de la phrase sous l'aspect précédent, tous les découpages sous la forme suivante

$\langle$phrase$\rangle$ ::= $\langle$T(terme primaire)$\rangle$

ou le

$\langle$terme primaire$\rangle$ ::= $\langle$base$\rangle$ $\langle$affixe$\rangle$

Celui de l'analyse morphologique est de calculer à partir des renseignements apportés par la segmentation les attributs lexicaux de chaque terme primaire.

En fait, ces deux phases ne sont pas successives, elles sont imbriquées l'une dans l'autre.

-IV - LE TERME PRIMAIRE

Un terme primaire qui a toujours un affixe nul est un mot invariable.

Il est variable s'il existe au moins un affixe non nul qui peut le suivre.

La base est écrite :

- soit en katakanas $B_1 = T(K)$
- soit en idéogrammes $B_2 = T(I)$
- soit en hiraganas $B_3 = T(H)$
- soit en idéogrammes et hiraganas $B_4 = T(I)\ T(H)$

90 °/₀ des bases du type B_2 ont moins de trois idéogrammes.

Parmi les bases du type $B_4 = T(I)\ T(H)$ il y a celles dont :

$$T(I) \in \{B_2\}$$

et celles dont :

$$T(I) \notin \{B_2\}$$

T(H) sera alors appelé terme complémentaire "tc" . Une base du type B_2 ou B_4 peut toujours s'écrire entièrement en hiraganas, mais une base du type B_2 ne peut jamais se mettre sous la forme T(I) T(H)

Certains mots écrits jadis aves des idéogrammes sont écrits maintenant avec des hiraganas car il a été promulgué une loi limitant l'emploi des idéogrammes dans la langue japonaise.

- V- EXEMPLES DE PHRASE JAPONAISE

Les différentes suites primaires sont les suivantes:

S_1 = フランス (K) で (H)

S_2 = 糸合 (I) がきの (H)

S_3 = 三木 (I) さんは (H)

S_4 = 会 (I) にまいにち (H)

会	A	フ	HU
い	I	ラ	RA
ま	MA	ン	N
し	SI	ス	SU
た	TA	で	DE
が	GA	糸合	E
°		か	KA
		き	KI
		の	NO
		三	MI
		木	KI
		さ	SA
		ん	N
		に	NI

Le découpage de chacune de ces suites en terme primaire est :

S1 { t_1 = フランス
HURAN SU
FRANCE

t_2 = Z"
DE (Particule)
DANS

S2 { t_3 = 来会 カ'キ
E KAKI
ARTISTE

t_4 = の
NO
(Particule)

S3 { t_5 = 三木
MIKI
MIKI

t_6 = に
NI
(Particule)

t_6 = さん
SA N
MONSIEUR

S4 { t_7 = 会いました
AI MASITA
RENCONTRER

t_8 = カ'
KA
(point d'interrogation).

Avez-vous rencontré Monsieur l'artiste MIKI en France ?

CONSTITUTION DU DICTIONNAIRE EN VUE DU

DECOUPAGE

-I- PRELIMINAIRE

Considérons une phrase formée par une suite de symboles

$$\alpha_1 \alpha_2 \alpha_3 \ldots\ldots\ldots \alpha_{p-1} \alpha_p$$

Le découpage a pour but principal de déterminer dans une telle chaine les différents morphèmes successifs.

$$M_1 M_2 \ldots\ldots\ldots M_{k-1} M_k \qquad (k \leq p)$$

qui la constitue.

Dans une langue naturelle les morphèmes sont agglutinés les uns aux autres et séparés de temps en temps par des séparateurs "S".

$$M_1 M_3 M_3 M_4 \quad S \quad M_5 M_6 \quad S \ldots\ldots\ldots S M_k$$

En japonais le séparateur blanc // n'existe pas. On distingue les séparateurs réels qui comprennent tous les signes de ponctuation, et les séparateurs virtuels qu'il convient de placer virtuellement entre deux symboles successifs dont le premier est un hiragana et le second un idéogramme ou un katakana.

Les séparateurs isolent les suites primaires de la phrase japonaise.

Lorsqu'une suite primaire est formée par l'agglutination de plusieurs morphèmes M_1 M_2 .....M_k, on se trouve en présence d'un code dit "indécodable dans le cas où il existe au moins un $j < k$ ~~tel que la~~ dans lequel ou les premières lettres de M_j constituent aussi un morphème, également dans le cas où M_j peut former un morphème avec les symboles qui le suivent.

Plusieurs chaines de morphèmes possibles sont ainsi obtenues et au niveau morphologique la seule chaine exacte ne peut pas être déterminée Il faut faire appel à des critères syntaxiques et sémantiques.

Il est évident que du point de vue machine on est dans l'impossibilité de rechercher toutes les solutions morphologiquement valables. On a été amené à faire un compromis.

Lorsque l'on a commencé l'étude du japonais, on s'est donné à priori une méthode grossière de découpage méthode ~~de~~ "longest match" pour entreprendre l'étude linguistique en vue de la traduction automatique. Mais bientôt l'étude linguistique montrait que l'on laissait de côté un nombre important de solutions morphologiquement valables et surtout que les solutions trouvées étaient fausses sur le plan syntaxique et sémantique.

Il fallu modifier légèrement le découpage, lequel avait une repercussion sur l'étude linguistique, finalement on arrivait à un compromis compte tenu de toutes les contraintes que l'on s'était fixées.

Les exemples qui suivent illustrent ces difficultés.

1); KARE HAKESA GATUKO HE ITUTA
IL [illegible]

La méthode de la base la plus longue donne la sulution

KARE HAKESA etc...

alors que la solutions est

KARE	A	KASA	GATUKO	HE	ITUTA
il	marque le sujet	ce matin	école	à	est allé.

2) Considérons le verbe

YOMEBA — Je lis

I H H

On peut adopter les quatre découpages ci-dessous :

YO	(Base)	ME	(Suffixe)	BA	(Désinence)
YO	(Base)			MEBA	(Désinence)
YOM	(Base)			EBA	(Désinence)
YOME	(Base)			BA	(Désinence)

Il est clair que le dictionnaire des bases et la liste des affixes seront différents selon que l'on adopte l'un ou l'autre de ces découpages, en outre les informations associées à chaque Base ne seront pas les mêmes lorsqu'elle pourra être suivie d'un suffixe ou directement d'une désinence.

En principe pour une forme donnée c'est la base la plus fine possible qui sera mise dans le dictionnaire pour en réduire le volume, ce qui augmente en revanche les cas d'homographie.

Ainsi, dans l'exemple précédent, YO sera la base du verbe lire.

Le compromis adopté est le suivant :

-Etant donné une suite primaire de la forme T(I) T(H) on recherche parmi la suite d'idéogrammes la base la plus longue.

- Si elle n'existe pas, le terme complémentaire le plus long qu'exige T(I) pour former une base est recherché, et cette base est enregistrée en mémoire.

- Si elle existe, on l'enregistre et si elle admet un terme complémentaire le plus long est identifié et la nouvelle base est enregistrée également en mémoire. Dons si dans une suite primaire, la Base la plus longue T(I) T(H) est enregistrée, la solution T(I) l'est également si T(I) $\in \{B_2\}$

Etant donné une suite primaire de la forme T(H) ou T(K) T(H) la base la plus longue sera seule identifiée, excepté pour quelques bases formées d'un seul kana qu'il ne faut absolument pas laisser échapper

HAKESA HA particule
HAKESA Brosse.

STRATEGIE GENERALE

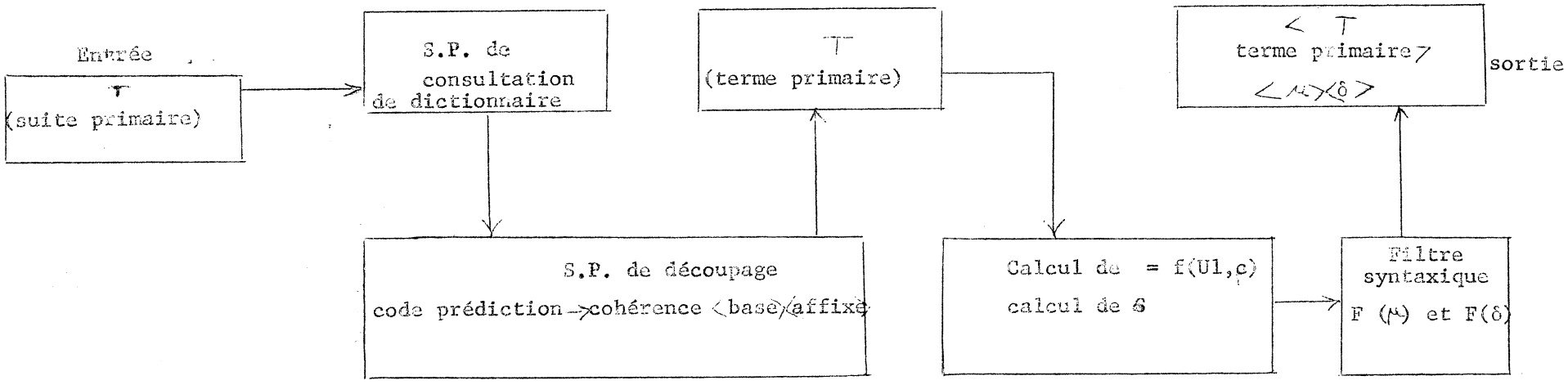
1.) Le texte entre sous la forme d'une série de suites primaires

2.) Le sous programme de consultation du dictionnaire identifie une base [illegible] du découpage les affixes possibles à l'aide des informations associées à la base, et [illegible] de signaler à la fin de la phrase.

3-) On recommence pour les bases laissées de côté, et l'on obtient une ou plusieurs séries de termes primaires.

4-) Le ou les codes morphologiques "codes μ " viennent remplir la mémoire correspondant à chaque base. Il en est de même pour les codes désinences "codes δ".

5-) Les filtres $F(\mu)$, $F(\delta)$, $F(\mu,\delta)$ rejettent une partie des solutions syntaxiquement incorrecte.



STRATEGIE GLOBALE

-III- LA CODIFICATION

1) niveau du caractère alphabétique

12 bits sont utilisés par caractère

N°	idéogramme		0	Idéogramme
N°	hiragana	1	1	Hiragana
N°	katakana	0	1	Katakana

Le premier bit à droite indique s'il s'agit d'un idéogramme ou d'un kana. Dans ce dernier cas le second bit marque la distinction entre katakana et hiragana.

En écriture romaine la kana est représenté par deux lettres (une voyelle et un consonne en général).

On pourrait réserver les 10 bits restants pour coder séparément la voyelle et la consonne. Ceci permettrait de créer des bases virtuelles d'un mot bien qu'il ne forme qu'un seul caractère. Le procédé a été mis en pratique par Monsieur KUNO à l'Université de Harvard, mais ne sera pas utilisé ici.

2) Niveau de la Base

- Pour réaliser le découpage, on a besoin de renseignements morphologiques pour savoir quels affixes il faut rechercher après avoir identifié une base, mais de découpage apporte lui-même des informations morphologiques

- Les codes prédictions "code p" associés à chaque base contiennent les informations nécessaires au découpage et permettent de déterminer la cohérence ⟨Base⟩ ⟨affixe⟩

- Les codes affixes et le code morphologique " code μ" sont déterminés par le découpage. Le code est fonction de la base et du code prédiction considéré.

Comme la base est repérée par son unité lexicale Ul

$$\mu = f(Ul, p)$$

-IV- CONSTITUTION DU DICTIONNAIREE

Il y a trois sortes de dictionnaire

1) Un dictionnaire pour les bases écrites en katakanas $\in \{B_1\}$
2) Un dictionnaire pour les bases écrites en hiraganas $\in \{B_3\}$

il est assez réduit car il comprend tous les mots qui s'écrivent seulement en hiraganas (une quarantaine de particules , quelques verbes et adjectifs auxiliaires etc...) plus quelques bases qui se mettent indifféramment en idéogrammes ou kanas.

Certes en principe toutes les bases $\in \{B_2\} \cup \{B_4\}$ peuvent se transposer en hiraganas, mais dans la réalité il y en a un nombre restreint parce que pour les japonais eux mêmes il y aurait de la difficulté à saisir le sens des mots parmi tous les homographes possibles dans une longue suite d'hiraganas.

3) Un dictionnaire pour les bases $\in \{B_2\} \cup \{B_4\}$

Il sera construit sous la forme d'arbre telle qu'elle a été préconisée par LAMB et étudiée par VEILLON dans sa thèse.

Le principe d'identification d'une base commençant par un ou plusieurs idéogrammes est défini de la façon récurrente suivante :

Supposons que l'on ait identifié d'une manière ou d'une autre la suite $\prod_{i=1}^{l} (I_i)$ des l premiers idéogrammes d'une base.

Soit S_{l+1} le caractère suivant.

La suite $\prod_{i=1}^{l} (I_i)\, S_{l+1}$ constitue-t-elle les (l+1) premiers casactères d'une base ? On peut répondre à cette question si l'on a dressé au préalable la liste $\left\{ I_{l+1}^{k} \right\}$ des k idéogrammes qui peuvent s'associer à $\prod_{i=1}^{l} (Ii)$ pour former les (l+1) premières lettres d'une base, ainsi que la liste des $\left\{ H_{l+1}^{j} \right\}$ des hiraganas qui s'associent à $\prod_{i=1}^{l} (Ii)$

Une simple consultation de liste révèle alors si :

$$S_{l+1} \in \left\{ H_{l+1}^{j} \right\} \cup \left\{ I_{l+1}^{k} \right\}$$

On a défini une structure d'arbre qui est évidemment récurrente puisqu'il suffit de répéter le même procédé pour le caractère $S_{l+2}, \ldots$

$$\left\{ H_{l+1}^{j} \right\} \cup \left\{ I_{l+1}^{k} \right\} = f\left(\prod_{i=1}^{l} (Ii) \right)$$

Il faut indiquer si $\prod_{i=1}^{l} (Ii)$ peut former une base à elle toute seule c'est-à-dire si $\prod_{i=1}^{l} (Ii) \in \left\{ B_2 \right\}$

Dans les trois cas ci-dessous on évite une consultation de table pour S_{l+1}

$\left\{I^k_{l+1}\right\} \cup \left\{H^j_{l+1}\right\} = \Phi \Longrightarrow$

$\left\{I^k_{l+1}\right\} = \Phi$ et $S_{l+1} \in \left\{I\right\} \Rightarrow i_1^l(I_1)$ est la base la plus longue

$\left\{H^j_{l+1}\right\} = \Phi$ et $S_{l+1} \left\{H\right\} \Rightarrow$

Sur 1800 idéogrammes, 1500 environ peuvent former une base à eux seuls. A chacun d'eux considéré comme première lettre est associée une mémoire dont l'adresse A

A = N (numéro de codification de l'idéogramme)

Cette mémoire contient l'adresse A_D de la mémoire qui donne les codes de prédiction associé à la base ainsi que l'adresse de la première mémoire du bloc $\left\{H^j_2\right\} \cup \left\{I^k_2\right\}$

$A_D = 0 \Longrightarrow I_1^\rho$ ne peut former une base à lui tout seul

$A_D = 1 \Longrightarrow$ la base est uniquement celle d'un terme primaire invariable.

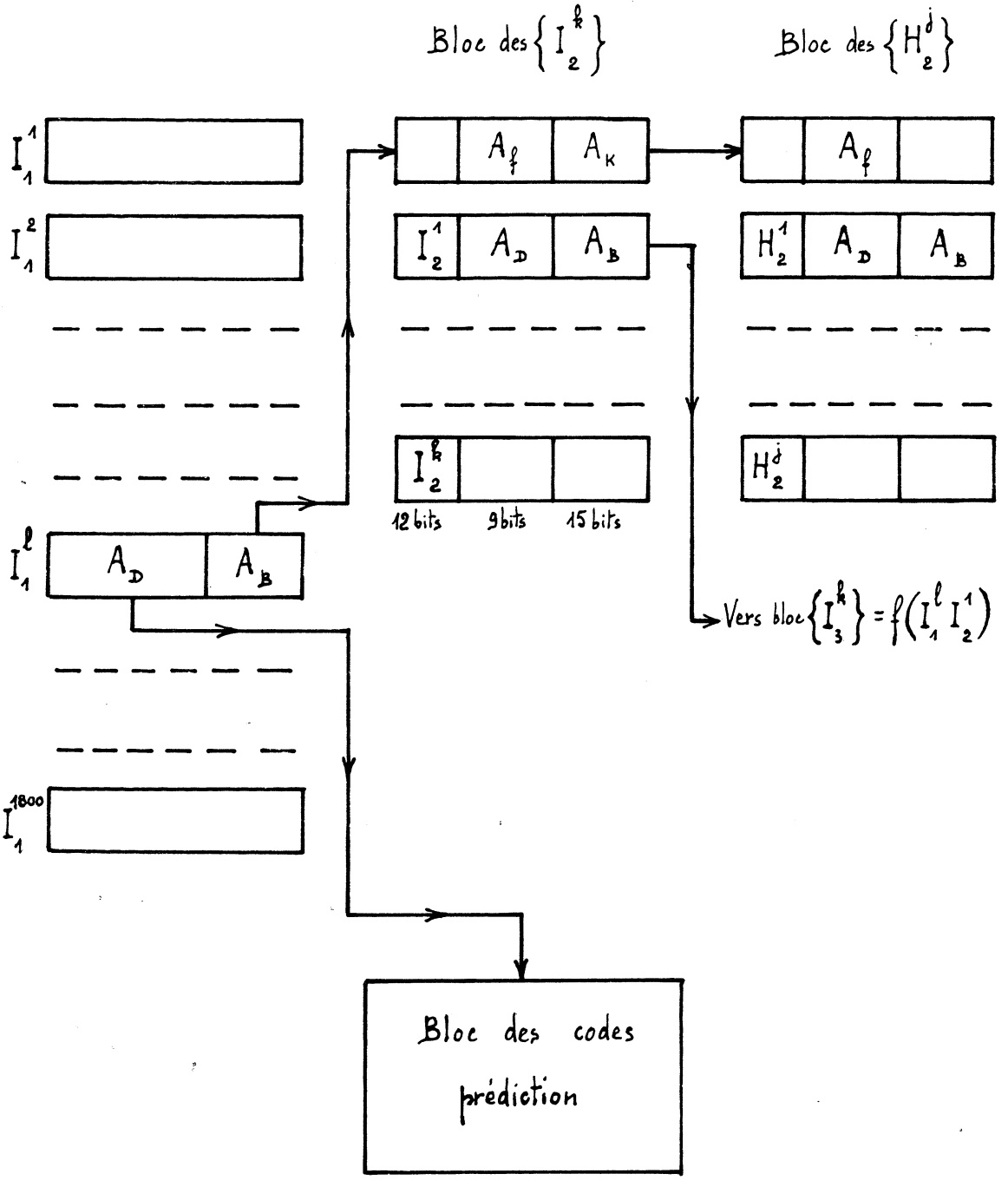
Ceci afin d'accélérer le découpage et d'identifier aussitôt une nouvelle base, sans avoir à consulter la mémoire d'adresse A_D

A_B donne l'adresse de la première mémoire du bloc $\left\{I^k_2\right\} \cup \left\{H^j_2\right\}$

$A_B = 0 \Rightarrow \left\{I^k\right\} \cup \left\{H^i_2\right\} = \Phi$

A_k adresse du sous bloc $\left\{H^j_2\right\}$

$A_k = 0 \Rightarrow \left\{H^i_2\right\} = \Phi$


Bloc des $\{I_2^k\}$
Bloc des $\{H_2^j\}$
I_1^1
I_1^2
I_1^l
I_1^{1800}
A_D
A_B
A_f
A_K
I_2^1
I_2^k
H_2^1
H_2^j
12 bits
9 bits
15 bits
Vers bloc $\{I_3^k\} = f(I_1^l I_2^1)$
Bloc des codes prédiction

A_f l'adresse du dernier idéogramme des $\{I_2 i\}$ ceux-ci sont placés dans les mémoires successives par ordre croissant de N° de codification, le role du code A_f est de permettre une consultation de liste par dichotomie des I_2^1 connaissant l'adresse de $A_B + 1$ du 1er idéogramme, et l'adresse A_f du dernier.

Un tel dictionnaire offre la possiblilité de lire toutes les bases contenues dant une suite de caractères de la plus courte à la plus longue. On a vu précédemment que du point de vue machine il n'était pas possible d'identifier toutes les bases. L'intérêt d'un tel dictionnaire est de ne pas avoir à être modifié si l'on vient ultérieurement à changer de méthode de découpage.

-V- LE DECOUPAGE

Une base une fois identifiée et enregistrée par l'adresse de sa dernière lettre, il s'agit de savoir s'il faut lire parmi les caractères qui suivent dans la suite primaire une nouvelle base ou un affixe.

Le problème serait simple si l'on avait affaire à l'une ou l'autre de ces deux éventualités, mais en réalité une base donnée peut être à la fois celle d'un ou plusieurs mots variables et celles d'un ou plusieurs mots invariables.

Exemple : TA (en hiragana)

Base du substantif invariable	Rizière
Base du verbe TASU	Additionner
	s'élever
TATU	couper
	se construire

Si les hiraganas suivent cette base, il est donc indispensable de rechercher après la solution TA (rizière) une nouvelle base et de continuer la chaine de découpage jusqu'au point final ; puis de revenir en arrière et d'identifier l'affixe SU, ou l'affixe TU pour former une deuxième ou troisième chaine.

Classe de découpage

Après une base on donne la liste des classes de découpages des affixes qui peuvent la suivre.

Il y a la classe de découpage des bases invariables et environ 25 classes pour le verbe et l'adjectif.

code prédiction

A chaque classe φ est associé un code prédiction ρ qui prédit le système d'affixes qui peut suivre la base.

La prédiction sera dite satisfaite si un affixe $A \in \varphi_a$ bien été identifié.

Plusieurs prédictions peuvent être satisfaites

En effet on peut trouver

$A_1 \in \varphi_1$

$A_2 \in \varphi_2$

tel que A_1 constitue les premières lettres de A_2 on écrira $A_1 \subset A_2$

Par classe on n'identifie qu'un seul affixe, c'est toujours le plus long.

Nous supposerons que quatre prédictions au maximum sont associées à une base.

Pour une bsse donnée, chaque code prédiction sera repéré par son numéro d'ordre dans la mémoire d'adresse A_D associée à la base.

A_D	p_1	p_2	p_3	p_4

Le code prediction comprend un "code f" code fréquence. S'il vaut 0, cela signifie que si la prédiction p_j a été réalisée il n'y a pas lieu d'examiner la prédiction p_{j+1} soit qu'elle n'existe pas soit que la probabilité pour qu'elle soit satisfaite est négligeable.

Ceci afin d'accélérer le découpage et d'éviter de rechercher des affixes improbables.

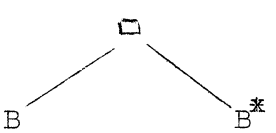
L'ensemble de toutes les chaines de découpage qu'elles conduisent à des solutions correctes ou incohérentes peut se présenter sous forme d'arbre Dans cet arbre on appelle :

noeud de terme complémentaire

le noeud qui exige de rechercher les deux solutions B et B^*

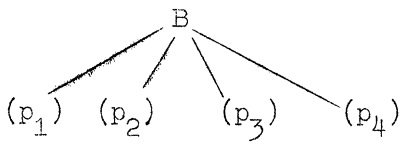
$B \subset B^*$

B^* est la base avec terme complémentaire



noeud de prédiction

celui qui pour une base donnée ouvre la voie à plusieurs embranchements qu'il faudra tous examiner



Dans cet arbre, on s'aperçoit qu'il y a des voies à succès qui donnent les solutions valables sur le plan morphologique des voies qui n'aboutissent pas soit par incohérence entre la base et l'affixe, soit que l'on ne trouve pas de base dans le dictionnaire.

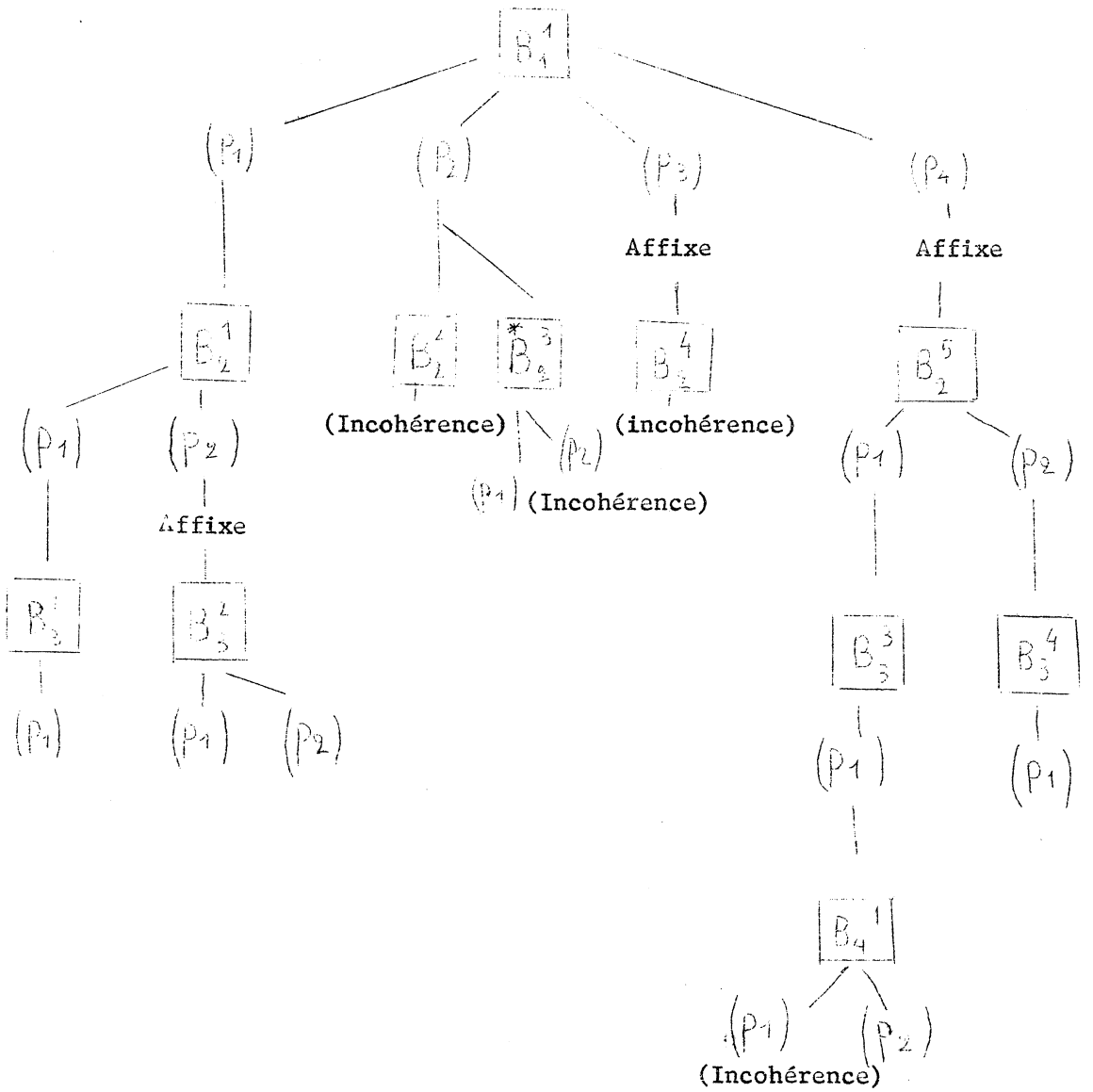
La difficulté du découpage est de construire cet arbre dont on ignore au départ et en cours d'élaboration totalement le nombre de branches Il faut donc en cours de route redonner la possibilité de revenir en arrière pour parcourir toutes les ramifications laissées de côté.

C'est pour cela que pour une base donnée on notera les 5 informations suivantes :

Adresse de la base	2	3	4	5

1- Adresse de la base
2- Admet-elle un terme complémentaire ?
3- Si oui est-ce la base B^* contenant le terme complémentaire ?
4- Numéro d'ordre du code prédiction
5- code f = 0 ? autrement dit : doit-on passer à la prédiction suivante?

EXEMPLE D' ARBRE (1)



L'indice inférieur donne le numéro du niveau de chaque base.

Toutes les chaines de découpage peuvent être identifiées à l'aide d'un tableau matriciel d'éléments

$$\left[B_k^j \ (l) \ \delta \right]$$

ou B_k^j représente la base de niveau k

l le numéro du code prédiction considéré

$\delta=1$ si p_{l+1} doit être examiné

$\delta=0$ si p_{l+1} ne doit pas être examiné.

L'arbre (1) se représente alors sous la matrice suivante :

B_1^1 (1) 1	B_2^1 (1) 1	B_3^1 (1) 0		fin suite primaire
———	B_2^1 (2) 0	B_3^2 (1) 1		fin " "
———	———	B_3^2 (2) 0		fin " "
B_1^1 (2) 1	B_2^2 (1) 0			incohérence
———	B_2^{*3} (1) 1			incohérence
———	B_2^3 (2) 0			fin suite primaire
B_1^1 (3) 1	B_2^4 (1) 0			incohérence
B_1^1 (4) 0	B_2^5 (1) 1	B_3^3 (1) 0	B_4^1 (1) 1	incohérence
———	———	———	B_4^1 (2) 0	fin suite primaire
———	B_2^5 (2) 0	B_3^4 (1) 0		fin suite primaire.

Fin découpage

Une ligne représente une chaine, c'est-à-dire une solution morphologiquement valable lrosqu'il n'y a pas d'incohérence.

Le trait (___) signifie que la chaine est recopiée identiquement à celle du dessus jusqu'à l'élément $\lceil B_i(1)\ \delta \rfloor$ exclus, le plus à droite qui posséde une valeur $\delta = 1$. A partir de là on explore une nouvelle ramification et ainsi de suite jusqu'à ce $\delta = 0$ pour tous les éléments de la chaine.

Le découpage réalisé on ne conserve pour chaque forme que les adresses de sa base, suffixes, désinence, ainsi que le numéro d'ordre de son code prédiction.

Chacune des mémoires précédentes contient respectivement le code morphologique, les codes suffixes et le code désinence de la forme.

L'unité lexicale est donnée par l'adresse de la base et le numéro d'ordre du code prédiction.

Exemple : $Ul_1 = 1000 \quad 1$

$Ul_2 = 1000 \quad 2$

sont deux unités lexicales différentes bien que dans chaque cas l'adresse de la base soit la même.

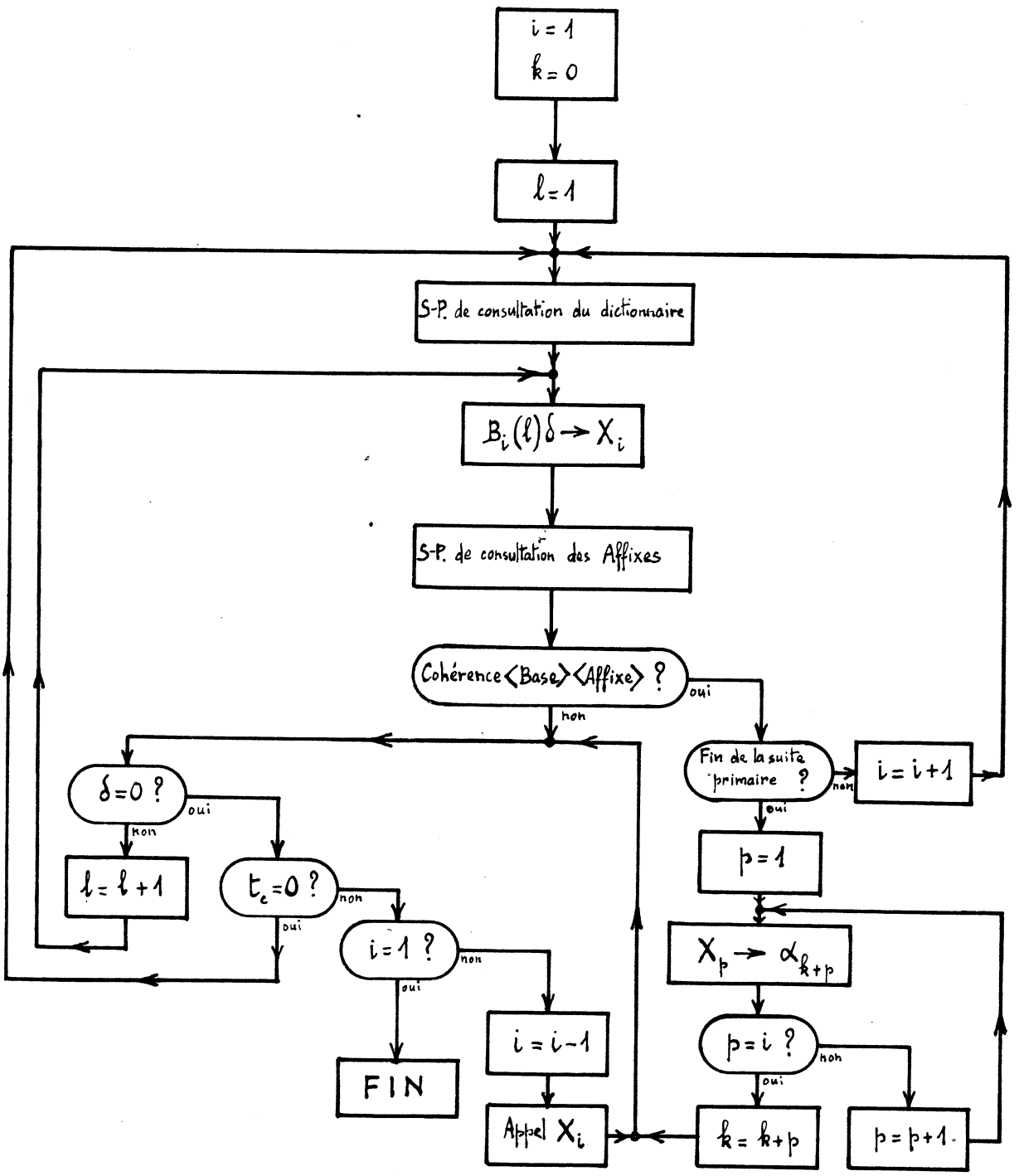
-VI- ORGANIGRAMME DE DECOUPAGE

Cet organigramme permet d'explorer toutes les voies à succès, ou les voies qui n'aboutissent pas, de n'importe quel arbre du type précédent, donc de toute suite primaire.

Les mémoires X_i sont des mémoires de travail tant que l'on n'est pas arrivé à la fin d'une suite primaire.

Les mémoires α_{k+p} sont des mémoires de stokage de toutes les formes valables pour chaque voie à succés.

Sous-programme de recherche d'un arbre de découpage



LES INFORMATIONS MORPHOLOGIQUES DES PRINCIPALES CATEGORIES GRAMMATICALES ET LEUR EXPLOITATION DANS LA RECHERCHE DES AFFIXES

-I- LE VERBE

1-) Préliminaire

Monsieur KUNO propose le découpage suivant du verbe :

$$\langle \text{terme primaire} \rangle ::= \langle \text{base} \rangle \langle \prod_{i=1}^{2} Si \rangle \langle \delta \rangle$$

La base est virtuelle, elle est coupée artificiellement au milieu d'un hiragana. S_1 et S_2 sont des affixes de dérivation et δ la désinence.

La recherche des affixes est relativement simple, mais chaque verbe possède deux ou trois bases, ce qui augmente le volume du dictionnaire nous décomposerons le verbe ainsi :

$$\langle \text{terme primaire} \rangle :: = \langle \text{Base} \rangle \langle \prod_{i=1}^{3} Si \rangle \langle \delta \rangle$$

Si pouvant être vide quel que soit i

Base
••••

La base est réelle et il n'y a qu'une seule base par verbe. Certaines bases admettent obligatoirement un suffixe $S_1 \neq \phi$ les autres n'en tolèrent aucun. Ces dernières s'écrivent entièrement en hiraganas ou en idéogrammes.

La grammaire distingue :

- les verbes réguliers et irréguliers
- les verbes ordinaires, les verbes auxiliaires et semi-auxiliaires
-- les copules (tiennent lieu de verbe être)

Les verbes auxiliaires ajoutent une nuance sémantique (nuance de probabilité, vraisemblance...) aux verbes ordinaires qu'ils accompagnent ; les verbes semi-auxiliaires n'ajoutent cette nuance que s'ils suivent des verbes ordinaires conjugués à des formes bien déterminées, autrement ils conservent leur sens propre.

Suffixe S_1

Il n'est composé que d'un seul Kana

Si quelquefois il a la fonction de potentiel (il joue alors le rôle du verbe auxiliaire pouvoir). Le plus souvent il ne possède aucune fonction

Suffixe S_2

Suffixe composé d'un ou plusieurs kanas
a plusieurs fonctions

soit le causatif (verbe auxiliaire faire)
" " causatif + passif ou politesse
" " potentiel ou passif ou politesse.

Suffixe S_3

Suffixe formé de deux kanas marquant la politesse.

2-) Codes suffixes < Code Si >

< Code Si > ::= < N° du suffixe > < code fonction > < code sδ >

Le code fonction est composé de 4 bits marquant soit :

possibilité de potentiel ; causatif ; causatif + passif ; politesse .

suffixe S_1

Les suffixes S_1 : KE GE SE TE NE ME BE RE E

ont un < code fonction > ::= 1 | 0 | 0 | 0

tous les autres suffixes S_1 n'ont aucune fonction

Suffixe S_2	SO, SA, SI, SE, SU, SASI, SASE, SASU, SE RARE, SASERARE	0 0 1 0
	RE, RARE	0 0 1 1
Suffixe S_3	MASI, MASU, MASE	0 0 0 1

Le < code sδ > est le code système de désinence prédisant le système de désinences qui peut suivre ces suffixes.

Si S_3 et $S_2 = \phi$

A chaque combinaison (B S_1) correspond un ou plusieurs systèmes de désinences.

S'il existe au moins un $S_j \neq \phi$ $2 \leq j \leq 3$ c'est le dernier suffixe le plus à droite qui n'est pas nul qui commande le ou les systèmes de désinences.

Certaines combinaisons (B S_1) peuvent être suivies de suffixes S_2, S_3 ; d'autres n'admettent que S_3, d'autres enfin sont suivies aussitôt de δ.

S_2 S_3 sont cependant toujours facultatifs

Les suffixes S_2 à seule fonction causative sont suivis immédiatement d'une désinence.

3-) Codes verbaux

a) code prédiction du verbe <code p>

<code p> ::= <code Affixe> <code classe> <code f> <code ES_1> <code jS_1>

1 bit 1 bit 1 bit 1 bit 5 bits

Les trois premiers codes se retrouvent pour toutes les catégories grammaticales

code Affixe = 1 il y a lieu de chercher un affixe

= 0 la forme toute entière a été mise dans le dictionnaire

code classe = 1 on fait appel au sous-programme de découpage du verbe

<code ES_1> c'est le code existence d'un suffixe S_1

ES_1= 0 si la base n'est jamais suivie d'un suffixe S_1

ES_1= 1 si la base est toujours suivie d'un suffixe S_1

⟨code sS_1⟩ code système de suffixe S_1 , il prédit le système de suffixes S_1 qui peut s'associer avec la base

b) Code morphologique : ⟨code μ ⟩

⟨Code μ ::= ⟨code classe⟩⟨code régularité⟩⟨code nature⟩⟨code exception⟩

Le code exception = 1 lorsque la forme toute entière a été mise dans le dictionnaire, il est alors suivi immédiatement par les attributs lexicaux qu'il n'y a pas lieu de calculer.

Le code classe est le même que celui du code ρ il est répété ici car le découpage une fois réalisé le code prédiction est perdu.

Le ⟨code régularité⟩ = 0 dans le cas des verbes réguliers, et 1 autrement

Le ⟨code nature⟩ donne la nature du verbe:

verbes ordinaires, auxiliaires , semi-auxiliaires.

Table T_1 = (sS_1, s_1)

Une table $T_1 = (sS_1, S_1)$ donne la cohérence de la décomposition (BS_1). Elle est construite de la façon suivante :

A chaque décomposition (BS_1) permise on affecte un nombre formé par la juxtaposition des nombres (sS_1, S_1)

sS_1 : système de suffixe S_1 donné par le code prédiction

S_1 : numéro de suffixe S_1

lorsqu'il y a incohérence, on ne trouve pas la combinaison (sS_1, S_1) dans la table.

En face de cette table on dispose les codes suivants :

< Table | sS_1, S_1 | > : <p ES_2> <p ES_3> <Adresse dans T_2>

<p ES_2> indique la possibilité d'un suffixe S_2

<p ES_3> indique la possibilité d'un suffixe S_3

Si l'on ne trouve ni suffixe S_2 et S_3 le code <adresse dans T_2> fournit une adresse dans T_2 à laquelle correspondent les systèmes de désinence qui peuvent suivre la combinaison (BS_1).

Si l'on trouve un suffixe S_2 ou S_3 c'est le code du suffixe le plus à droite qui fournit les systèmes de désinences.

4°) Désinences

A chaque désinence est attachée plusieurs renseignements (Mode temps, voix affirmative ou négative) appelés type d'attributs lexicaux

<code désinence> ::= <N° de la désinence> <sδ> <attributs lexicaux>

le <sδ> indique à quel sδ appartient la désinence, puisqu'une désinence peut appartenir à plusieurs systèmes de désinences.

<code attributs lexicaux> : <code affirmation> <code mode> <code temps>
1 bit 4 bits 2 bits

ces trois derniers codes sont les suivants :

Affirmation		Mode		Temps	
Affirmation	1	Indicatif	1	Passé	1
Négation	2	Impératif	2	Présent	2
		Conjonctif	3	futur	3
		Conjonctionnel	6		
		Conditionnel	4		
		Indéfini	0		
		gérondif	5		
		progressif simultané	7		
		fréquentatif			
				indéterminé	0

Le mode est indéfini lorsqu'il n'est pas connu, de même le temps est indéterminé lorsqu'il est inconnu (à déterminer par l'analyse syntaxique) ou tout simplement lorsqu'il n'y en a pas.

Systèmes de désinences sδ

φ1	sδ 1
φ2	sδ 2
φ3	sδ 3
na mai	sδ 4
u'	sδ 5
nu neba zu zuni	sδ 6
nagara tutu	sδ 7
Ta [illegible] tarou tara taraba tari	sδ 8
da darou dara de daraba dari	sδ 9
ru runa rumai reba	sδ 10
ro yo you	sδ 11
qn	sδ 12

Parmi les $(s\delta)$ on relève les systèmes :

$$s\delta\ 1 = \phi\ 1$$
$$s\delta\ 2 = \phi\ 2$$
$$s\delta\ 3 = \phi\ 3$$

Cette distinction entre $\phi 1$ $\phi 2$ $\phi 3$ est nécessaire parce que la desinence nulle ϕ correspond à 3 types d'attributs lexicaux. Les différents $s\delta$ 1, $s\delta$ 2, $s\delta$ 3 permettent de savoir à quel type on a à faire.

Ainsi, le code $\langle s\delta \rangle$ du suffixe $SA \in S_2$
possède un 1 en position 2
1 en position 6

pour permettre de déterminer les deux solutions

$BS_1\ S_2\ \delta_2$ $\qquad\qquad$ $\delta_2 = \phi\ 2$ $\qquad$ (impératif négatif)
$BS_1\ S_2\ \delta^2$ $\qquad\qquad$ si $\delta \in s\delta\ 6$

de cette façon, à ϕ 2 est attaché un seul type d'attributs lexicaux, bien différents de $\phi 3$.

MATRICE DE COHERENCE DE LA DECOMPOSITION

BS_1 DANS LE CAS DES VERBES REGULIERS

A l'intersection de la ligne du suffixe et de la colonne du système de suffixes qu'exige la base verbale, on lit les systèmes de désinences compatibles avec cette décomposition.

s_1 \ 1s1	0	1	2	3	4	5	6	7	8	9	10	11
	6,7, 8,3 10,11											
ku		1,4	1,4									
ka		3,6	3,6									
ke		2,3,6 8,10	2,3,6 8,10									
ka		5	5									
ki		3,7	3,7									
Gu				1,4								
ga				3,-								
ge				2,3,6 8,10								
go				5								
gi				3,7								
su					1,4							
sa					3,6							
se					2,3,6 8,10							
so					5							
si					3,7,8							

S1 \ sδ1	0	1	2	3	4	5	6	7	8	9	10	11
tu			8			8,1,4				8	8	8
ta						3,6						
te						2,3,6 8,10						
to						5						
ti						3,7						
nu							1,4					
na							3,6					
ne							2,3,6 8,10					
no							5					
ni							3,7					
bu								1,4				
ba								6,3				
be								2,3,6 8,10				
bo								5				
bi								3,7				
mu									1,4			
ma									6,3			
me									2,3,6 8,10			
mo									5			
mi									3,7			
ru										1,4		1,4
ra										6,3		6,3
re										2,3,6 8,10		2,3,6 8,10
ro										5		5
ri										3,7		7
u											1,4	
wa											6,3	
e											2,3,6 8,10	
o											5	
l	8	9									3,7	2,3
gn						9	9	9				

SYSTEMES DE DESINENCES DEMANDES PAR LES SUFFIXES S_2 ET S_3

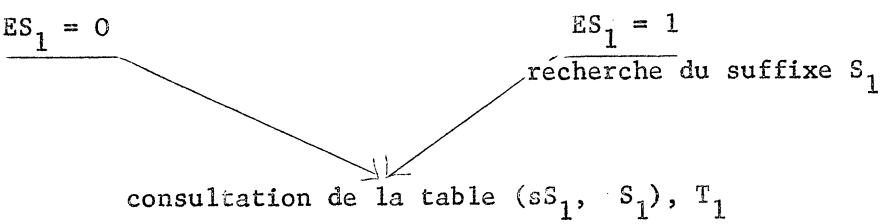
SA	s86
SO	s85
SI, SASI	s83, s87
SU, SASU	s81, s84
SE, SASE, SERARE, SASERARE, RE, RARE	s83, s86, s88, s810, s811
MASI	s88
MASU	s81, s84
MASE	$s\delta_6$, $s\delta_{12}$

5) Attributs lexicaux

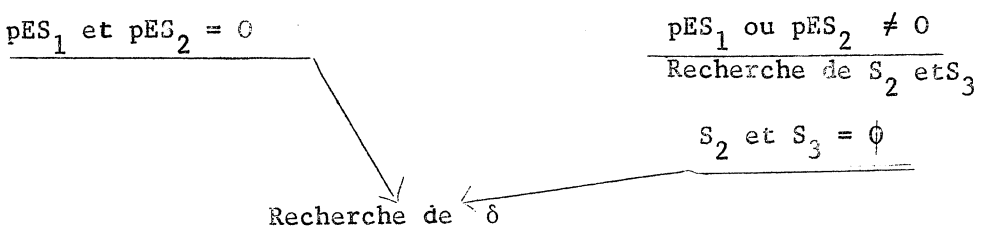
NOM	CODE DESINENCE (Attributs lexicaux)			DESINENCE
Indicatif présent	1	1	1	ϕ_1 , ru
Indicatif passé	1	1	1	ta, da
Indicatif présent néga.	2	1	2	ru, gn
Indicatif futur	1	1	3	u,
Indicatif futur nég.	2	1	3	mai, rumai,
Impératif	1	2	2	ϕ_2 , ro yo
Impératif négatif.	2	2	2	na, runa
conjonctif passé	1	3	1	tarou , darou
Conditionnel présent	1	4	2	ba, reba
conditionnel présent nég.	2	4	2	neba,
conditionnel passé	1	4	1	tara, taraba, da, daraba
Indéfini	1	0	0	ϕ_3
Conjonctionnel	1	6	0	te, de
gérondif	1	5	0	nagara
gérondif, négatif	2	5	0	zu, zuni
progressif simultané	1	7	0	tutu
fréquentatif	1	8	0	tari, dari.

6) Principe du découpage et de la recherche des attributs lexicaux du verbe japonais

Le code classe de la base verbale une fois idientifié, on teste la valeur du code ES_1, si le code affixe = 1 pour savoir si l'on doit se rechercher un suffixe S_1



Cette table établit la cohérence de la décomposition (BS_1) indique si la décomposition (BS_1) peut admettre des suffixes S_2 et S_3, et donne dans le cas ou (BS_1) n'en admet pas l'adresse T_2 des systèmes de désinences qui peuvent suivre (BS_1). C'est cette même adresse que l'on doit consulter si (BS_1) peut admettre des suffixes S_2 ou S_3 et si on n'en trouve pas (puisqu'ils sont toujours facultatifs).



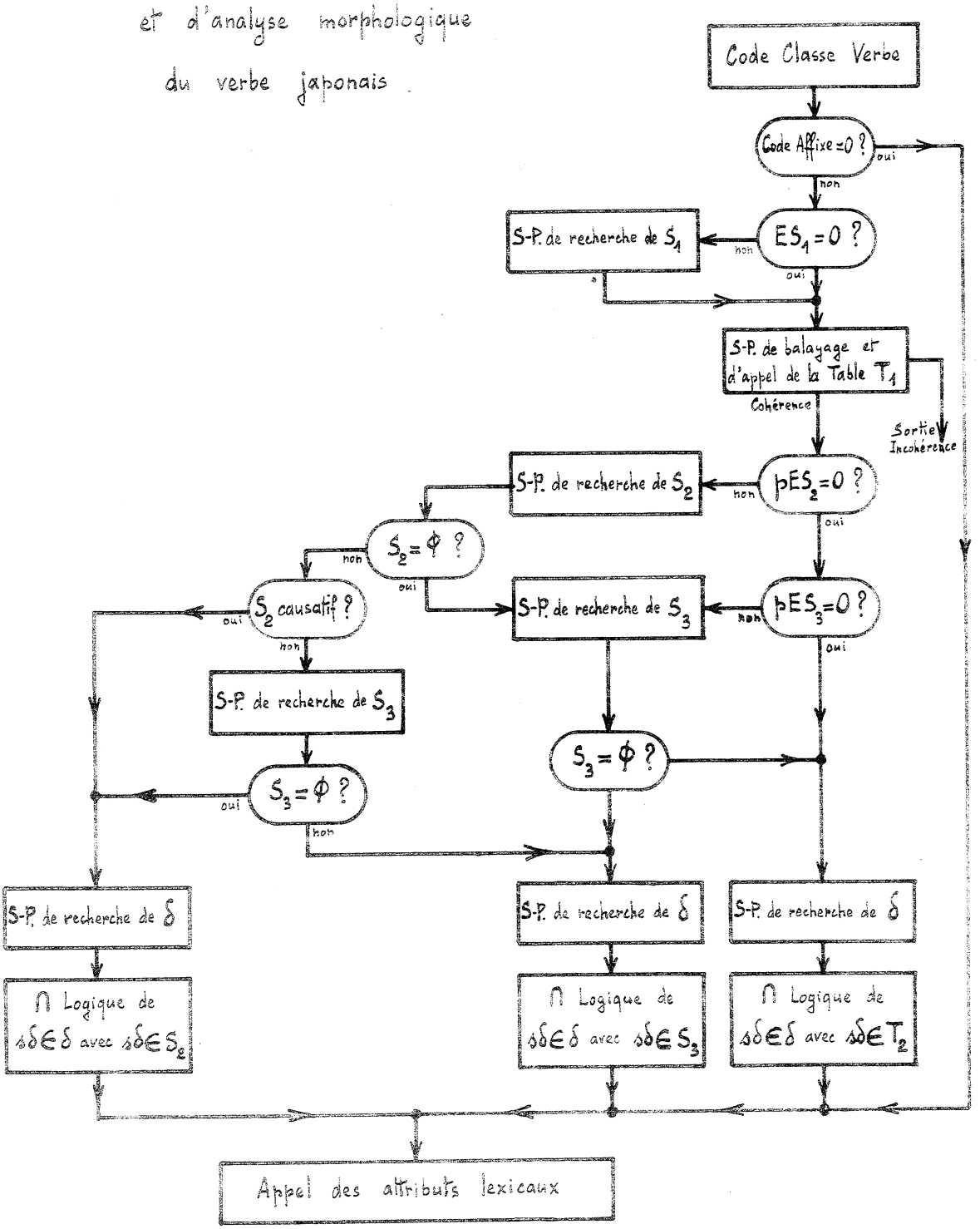
S_2 ou $S_3 \neq \phi$

recherche de δ

intersection logique

de $(s\delta) \in \delta$ et $(s\delta) \in S_2$ ou S_3

Sous-programme de découpage et d'analyse morphologique du verbe japonais



Le but de cette intersection logique est d'établir la cohérence de la décomposition (BS_1) ou $(BS_1\ S_2\ S_3)$ quand les suffixes S_2 ou S_3 existent, avec la désinence trouvée. IL suffit alors en cas de cohérence de lire les attributs lexicaux en face de δ et de recopier ceux correspondants à ϕ_1, ϕ_2 ϕ_3, là où il y a des 1 dans les positions respectives du code δ.

-II- L'ADJECTIF

Le terme primaire est da la forme :

< tp> ::= < base > < S_1 > < δ >

Un même adjectif a quelquefois deux bases
Les suffixes S_1 sont au nombre de deux :

" SOU" suffixe de vraisemblance
"SA" transforme la base de l'adjectif en celle d'un substantif il est alors nécessaire de changer le code classe de la base adjectivale.

code prédiction de l'adjectif

<code ϕ> ::= <Code Affixe> <code classe> <code f> <code sδ>
<le code sδ > fournit le N° du système de désinences qui est cohérent avec l'adjectif. Il y a 6 systèmes de désinences, ce code sera donc un code à 6 bits
code classe = 0 pour l'adjectif.

code morphologique
< code > ::= <code classe> <code nature> <code exception>

code désinence

<code δ> ::= < N° de la désinence> < système de désinences>.

Le code <sδ> comme pour le verbe fournit les systèmes de désinences auxquels appartient la désinence. (Une désinence pouvant appartenir à des systèmes différents).

L'intersection logique entre (sδ)∈<code ρ> et <sδ>∈<code δ> établit la cohérence de la décomposition Bδ.

Désinence

Attributs lexicaux.

L'adjectif attribut tient lieu de verbe être. Une désinence d'un adjectif attribut porte donc comme le verbe, des renseignements de temps de mode et de voix affirmative ou négative.

En plus elle donne une indication de fonction, attribut, épithéte, adverbe. Le code fonction sera

0 : Epithète ou attribut
1 Attribut
2 Epithète
3 attribut ou adverbe.

Le tableau qui suit donne le type d'attribut lexicaux de chaque désinence.

NOM	ATTRIBUTS LEXICAUX				DESINENCES
	Fonction	Affirm.	Mode	Temps	
Epithéte ou attribut présent	0	1	1	2	i
Att.passé	1	1	1	1	katuta,datuta
Att. futur	1	1	1	3	karou, darou
Conjonctif passé	1	1	3	1	katutarou, datutarou
Epit ète	2	1	[illegible]	0	, ki, naru
Epithète négatif	2	2	0	0	karanu, naranu
fréquentatif	1	1	8	0	katutari, datutari
participe adv.	3	1	0	0	ku, ni
participe nég.	1	2	0	0	arazu, narazu
participe	1	1	0	0	u
condit. présent	1	1	4	2	kereba, maraba
condit. passé	1	1	4	1	katutara(ba) datura(ba)
conjonctionnel	1	1	6	0	kute, de
Attribut présent	1	1	1	1	da

SYSTEMES DE DESINENCES

	1	2	3		4
1	i $\emptyset_2$	i	i		$\emptyset$
2			da		da
3	katuta	katuta	katuta	datuta	datuta
4	karou	karou	karou	darou	darou
5	katutarou	katutarou	katutarou	datutarou	datutarou
6		na	na		na
7	ki		ki	naru	naru
8	karanu	karanu	karanu	naranu	naranu
9	katutari	katutari	katutari	datutari	datutari
10	ku	ku	ku	ni	ni
11	u	u	u		
12	karazu	karazu	karazu	narazu	narazu
13	kereba	kereba	kereba	nara(ba)	nara(ba)
14	katutara(ba)	katutara(ba)	katutara(ba)	datutara(ba)	datutara(ba)
15	kute	kute	kute	de	de

SYSTEMES DE DESINENCES

	5	6		7
1	$\emptyset_1$	$\emptyset_1$		
2	da	da		da
3	datuta	datuta	katuta	datuta
4	darou	darou	karou	darou
5	datutarou	datutarou	katutarou	datutarou
6	na no	na	i	$\emptyset$ na
7	naru			
8	naranu	namaru	kanaru	
9	datutari	datutarou	katutarou	datutari
10	ni	ni	ku	ni
11				
12	narazu	narazu	karazu	
13	nara(ba)	nara(ba)	kare(ba)	nara(ba)
14	datutara(ba)	datutara(ba)	katutara(ba)	datutara(ba)
15	de	de	kute	de

-III- LES PARTICULES

La langue japonaise comprend une quarantaine de particules invariables placées à la suite des substantifs, des adjectifs ou des adverbes, et toutes écrites en hiraganas.

Tantôt elles tiennent lieu de préposition, conjonction ou adverbe :nou les appellerons particules à valeur fonctionnelle .

Tantôt elles indiquent un cas (sujet ou objet) nous les appellerons particules à valeur casuelle.

Elles peuvent aussi remplir ces deux rôles à la fois.

Désignons par $\{A\}$ l'ensemble des adverbes
$\{P\}$ " prépositions
$\{C\}$ " conjonctions

$$\{f\} = \{A\} \cup \{P\} \cup \{C\}$$

$\{k\}$ ensemble des particules à valeur casuelle. On peut alors classer les particules sous la forme suivante :

1°) Particules uniquement à valeur fonctionnelle $\in \{f\}$
elles appartiennent à l'un des 4 sous-ensembles suivants :

$\{P\}$
$\{A\}$
$\{C\}$
$\{P\} \cup \{C\}$

2) Particules uniquement à valeur casuelle $\in \{k\}$

cette particule est unique, c'est la particule wo qui indique toujours le cas sujet.

3) Particules $\in \{k\} \cup \{f\}$

4) Particules $\in (\{k\} \cap \{f\}) \cup \{f\}$

Le tableau suivant donne la liste des 40 particules et indique le sous-ensemble auquel elles appartiennent. Au total 8 sous-ensembles.

f			k ∪ f		k	f	(k∩f) ∪f
P	C	A	k∪c	k∪ p	k	P∪C	(k∩A) ∪A
Ni	yori	hodo	ga	no	wo	kara	ha
de	toka	kurai				to	Mo
he	dono.	gurai					sae
	yora						soemo
	,aro						sura
	ka						suramo
	ya						koso
	keredo						demo
	keredomo						bakari
	node						nomi
	noni						Hake
	kose						nodo
	kasenu						mode
	yoinago						siko

Code morphologique de la particule

<code$_{\mu}$> ::= <N° de la particule> <code particule>

<code particule> ::= <code différentiation> <code fonction> <code ensemble>.

Le code ensemble

indique à quel des quatre ensembles suivants appartient la particule

$\{k\} = 0\ 0$
$\{f\} = 0\ 1$
$\{k\} \cup \{f\} = 1\ 0$
$\{k \cap A\} \cup \{f\} = 1\ 1$

Le code fonction

donne la fonction de la particule lorsqu'elle en a une, ou peut en avoir une :

P = 0 0
C = 0 1
A = 1 0
P $\cup$ C = 1 1

Le code différentiation

permet de distinguer certaines particules à l'intérieur des huit sous ensembles précédents, il vaut 1 pour les particules soulignées.

Le code particule aura donc pour les particules de chaque type l'aspect suivant :

		code d	code f	code E
A	Hodo	0	1 0	0 1
P	Ni	0	0 0	0 1
C	yori	0	0 1	0 1
C	ka	1	0 1	0 1
k∪C	ga	0	0 1	1 0
k∪P	no	0	0 0	1 0
k	Wo	0	0 0	0 0
P∪C	kara	0	1 1	1 1
P∪C	Ho	1	1 1	0 1
(k∩A)∪A	Mo	0	1 0	1 1
(k∩A)∪A	Sika	1	1 0	1 1

Remarque

Dans le cas de la particule uniquement casuelle CODE E = 0 0 On n'a pas à tenir compte du code fonction, même s'il vaut 0 0

F I L T R E S

Un filtre rejette une partie des solutions incompatibles entre deux ou plusieurs formes d'une phrase. Il ne laisse passer que les solutions qui répondent à certains critères.

THEORIE GENERALE

Soient $\varphi(\mu 1, \mu 2, \ldots \mu m)$ les codes morphologiques associés à une classe de découpage

Une forme sera caractérisée parlle couple :

$$[\varphi(\mu 1, \mu 2, \ldots \mu m), \delta$$

couple de l'ensemble de ses codes et de son code désinence δ

(F) sera appelé filtre si aux n couples de n formes f_1 , f_2 .... f_n il fait correspondre n autres couples des mêmes formes définis comme suit :

$$[\varphi_1(\mu_1, \mu_2 \cdots \mu_j \cdots \mu_m)\ \delta_1][\varphi_2(\mu_1, \mu_2 \cdots \mu_1)\ \delta_2]$$
$$[\varphi_3(\ldots.)\ \delta_3]$$
$$\xrightarrow{F} [\varphi_1(\mu'_1, \mu'_2 \cdots \mu'_j \cdots \mu_k)\ \delta'_1]\ [\varphi_2(\mu'_1, \mu'_2 \cdots \mu'_n)\ \delta'_2][\varphi(\ )]$$

ou $k \leq m$; $n \leq 1$ .....

et $\mu 1$ et $\mu' 1$ } ont le même code ~~mo~~classe
$\mu 2$ et $\mu' 2$ }

ce qui signifie que (m-k) codes morphologiques ont été filtrés pour la forme f_1 et (n-1) pour la forme f_2 etc.....

Les codes μ_j et μ'_j ont le même code classe, mais leurs autres codes ont pu changer.

Rappelons qu'un code morphologique se compose toujours d'un code classe et de divers autres codes.

En général, $\delta' = \delta$, il est différent lorsque δ était indéterminé et que le filtre a levé l'indétermination.

Remarques

1) Il ne faut pas confondre classe de découpage et code classe

Exemple : La forme HA appartient à la classe de découpage des mots invariables

elle a deux codes classe

HA : $\varphi(\mu_1, \mu_2)$

particule | substantif (feuille)

2) le rôle du filtre est d'éliminer le code morphologique incorrect en tenant compte de la liaison de HA avec la forme qui le précède.

3) les classes de découpage ont été définies de telle sorte que seule la classe de découpage des mots invariables puisse contenir plusieurs code μ , $\delta = 0$ dans ce cas toutes les autres classe ont un seul code μ

$[\varphi(\mu_1, \mu_2 \cdots \mu_m)]$ mots invariables.

$[\varphi(\mu)\delta]$ pour toutes les classes de découpages des mots variables.

II ETUDES DE QUELQUES FILTRES

FILTRE F_1

F_1 agit sur deux formes f_j f_{j+1}

telles que

f_{j+1} forme invariable $[\varphi_{j+1}(\mu_1 \ \mu_2 \cdots \mu_1 \cdots \mu_m)]$

il existe un $1 \leq l \leq m$ tel que $c(\mu_l)$ = code classe particule

Définissons les ensembles suivants dans la liste des particules :

WO DE HE	$\{E_1\}$

NO GA NI KARA TO YORI HA MO SILA SAE(MO) DURA(MC) KOSO DEMO BAKARI NOMI DAKE MADE NADO GURAI DURAI HODO TOKA DANO YARA NARI YA DA	$\{E2\}$

KEREDO (MO) SI NODE NONI KUSENI YAINAYA	$\{E3\}$

A) f_j forme variable $\rightarrow$ $[\varphi_j(\mu), \delta]$

si $\{C(\mu)\}$ = code classe du verbe ou de l'adjectif

et si $f_{j+1} \in E_1$

alors

$[\varphi_j(\mu), \delta][\varphi_{j+1}(\mu_1, \mu_2 \cdots \mu_l, \mu_m] \Rightarrow$

$[\varphi_j(\mu), \delta][\varphi_{j+1}(\mu_1 \cdots \mu_{l-1}, \mu_{l+1} \cdots \mu_m]$

action 1

si $\begin{cases} c(\mu) & = \text{code classe du verbe ou de l'adjectif} \\ \text{et } f_{j+1} \in E_2 \cup E_3 \end{cases}$

et pour des numéros d'attributs lexicaux de δ qui sont fonction de f_{j+1}

Un tableau matriciel donne la liste des attributs lexicaux d

chaque particule.

alors :

$$[\varphi_j(\mu),\delta]\,[\varphi_{j+1}(\mu_1,\mu_2\cdots\mu_l\cdots\mu_m)] \xRightarrow{F_1} [\varphi_j(\mu),\delta]\,[\varphi_{j+1}(\mu_l)]$$

si le numéro d'attributs luxicaux est interdit avec la forme f_{j+1}

le filtre (F1) à l'action 1

B) f_j forme invariable $\longrightarrow [C_j(\mu_1,\mu_2\cdots\mu_K\cdots\mu_l)]$

B_1) Supposons qu'il existe pour la forme f_j un k tel que $C(\mu_k)$ = code classe substantif mais qu'il n'existe pas de i tel que $c(\mu_i)$ = code classe particule

alors si $f_{j+1} \in E_1 \cup E_2$

$$[\varphi_j(\mu_1,\mu_2\cdots\mu_K\cdots\mu_l)]\,[\varphi_{j+1}(\mu_1,\mu_2\cdots\mu_m)] \xRightarrow{F_1} [\varphi_j(\mu_K)]\,[\varphi_{j+1}(\mu_l)]$$

B_2) Pour la forme f_j il existe un i tel que $c(\mu_i)$ = code classe particule

$$\xRightarrow{F_1} [\varphi_j(\mu'_i)\,[\varphi_{j+1}(\mu'l)]$$

tous les codes morphologiques sont éliminés excepté les codes morphologiques particules, qui sont cependant modifiés.

C'est le problème de la combinaison de deux particules juxtaposées, qui modifient tout ou partie de leur code morphologique en s'associant.

	V1	V2	V3	V4	V5	V6	V7	V8	V12	V13	V14	V15	V17	A1	A2	A3	A4	A5	A6	A9	A10	A11	A13	A14	A15	Adv
wo	1																									
De	1																									
he	1																									
No	1	1		1										1		1		1								
Ga	1	1	1	1				1						1	1	1	1	1								
Ni	1	1																								
Kara	1	1								1				1	1	1		1								
To	1	1	1	1	1	1	1	1						1	1	1		1					1	1		1
Yori	1	1												1												
Ha	1	1							1	1	1	1	1	1						1	1	1			1	1
Mo	1	1							1	1	1	1	1							1	1				1	1
Sika	1	1							1		1		1							1	1	1				1
Kiri	1	1							1		1			1												
Sae	1								1	1	1		1							1	1					
Saemo	1								1	1	1		1							1	1					
Sure	1								1	1	1									1	1					
Suramo	1								1	1	1									1	1					
Koso	1								1				1								1					
Demo	1	1							1		1										1					
Bakari	1	1								1				1							1					
Nomi	1	1								1				1							1					
Dake	1	1												1				1	1		1					
Made	1	1											1	1												
Nado	1								1				1							1	1					
Hodo	1	1												1					1							
Kurai	1	1								1				1					1							
Gurai	1	1								1				1												
Toka	1	1	1								1			1	1	1		1		1	1		1	1		
Dano	1	1	1								1			1	1	1		1		1	1		1	1		
Yara	1	1	1								1			1		1		1		1	1		1	1		
Nari	1	1												1	1	1		1								
Ya	1	1																								
Ka	1	1	1	1	1		1	1						1	1	1		1								
Keredomo		1	1	1				1						1	1	1		1								
Si		1		1										1	1	1		1								
Node		1		1										1		1		1	1							
Noni		1		1										1		1		1	1							
Kuseni		1		1										1		1		1	1							
Yainaya		1																								

III COMBINAISON DES PARTICULES

En principe chacune des particules peut se combiner avec les autres, en perdant ou conservant toute ou partie de sa valeur fonctionnelle ou casuelle.

Quelquefois même, on peut rencontrer trois particules côte à côte. Nous désignerons par + la loi de composition associée au couple de deux particules $(P_1, P_2) = P_1 + P_2$ qui aux deux codes morphologiques de P_1 et P_2, fait correspondre un nouveau code morphologique pour le couple $P_1\ P_2$

Elle n'est pas commutative

Elle dépend du sous-ensemble auquel appartient chaque particule et quelquefois même (c'est le cas des particules soulignées) de la particule elle-même.

Il y a une cinquantaine de lois de compositions possibles la liste en est donnée dans les feuilles suivantes.

A chaque combinaison possible des particules, et associée une table T_1, dont chaque nombre et constitué par la juxtaposition des codes particules élémentaires des 2 particules associées.

En face de cette table il y a l'adresse T_2 du nouveau code particule du couple $(P_1\ P_2)$.

Il y a autant d'adresses que de lois de composition entre les particules.

Il est clair que le nouveau code particule est 2 fois plus long que le code particule élémentaire, ceci afin de savoir sur quelle particule porte la ou les fonctions.

$f_n + f_n$

f_n signifie ici : valeur fonctionnelle

$P + C = P + C$

$P + P \cup C = P + C$

$A + P = A + P$

$C + P = C + P$

$P \cup C + C = P + C$

$C + P \cup C = C + P$

$P + A \cup (k \cap A) = P + A$

$P \cup C + A \cup (k \cap A) = P + A$

$A + A \cup (k \cap A) = A + A \cup (A \cap k)$

$A + A \cup (k \cap A) = A + P \cup (A \cap k)$

$P \cup C + A \cup (k \cap A) = P \cup C + A$

$C + A \cup (k \cap A) = C + A$

$C + A \cup (k \cap A) = C + A$

$A + A \cup (k \cap A) = A + A$

$A + A \cup (k \cap A) = A + A$

$f_n + f_n \cup (f_n \cap k)$

$A \cup (A \cap k) + k \cup C = A + k$

$A \cup (A \cap k) + k \cup P = A + (P \cup k)$

$A \cup (A \cap k) + k \cup P = A + 0$

$A \cup (A \cap k) + k \cup C = A + k$

$f_n \cup (f_n \cap k) + f_n \cup k$

Loi de ' composition entre les particules

$f_n \cup (f_n \cap k) + k$

$A \cup (A \cap k) + k = A + k$

$A \cup (A \cap k) + k = A + 0$

$k + f_n \cup (f_n \cap k)$

$k + A \cup (A \cap k) = k + A$

$f_n + k$

$A + k = A + k$

$C + k = C + k$

$C + k = C + k$

$C \cup P + k = C + k$

$f_n + f_n \cup (f \cap k)$

$P + (k \cup P) = P + 0$

$P \cup C + (k \cup C) = C + 0$

$A + (k \cup P) = A + P$

$A + (k \cup C) = A + k$

$C + (k \cup P) = C + P$

$C + (k \cup C) = C + k$

$C + (k \cup C) = C + k$

$P \cup C + k \cup C = C + k$

$P \cup C + (k \cup P) = C + (k \cup P)$

$A + (k \cup P) = A + P$

$$f_n \cup (f_n \cap k) \neq f_n$$

$A \cup (A \cap k) + P = A + P$

$A \cup (A \cap k) + P \cup C = A + P$

$A \cup (A \cap k) + P \cup C = A + P \quad C$

$A \cup (A \cap k) + P = P + 0$

$$f_n \cup (f_n \cap k) + f_n \cup (f_n \cap k)$$

$A \cup (A \cap k) \neq A \cup (A \cap k) = A + k$

$A \cup (A \cap k) + A \cup (A \cap k) = A + A \cup (A \cap k)$

$A \cup (A \cap k) + A \cup (A \cap k) = k \cup P$

FILTRE F_2

F_2 agit sur les formes f_i f_k $k > i$

f_k est un verbe semi-auxiliaire appartenant à l'un des ensembles suivants :

OKU DURU IRU ORU ITADAKU IKU YARU AGERU MIRU KUDASARU NASARU	E_1
SIMAU IRATUSIYARU MAIRU	
URU KANERU TAGARU	E_2
ARU	E_3
SARU DEKIRU	

alors $[\varphi_i(\mu)\delta_1][\varphi_k(\mu)\delta_2] \xrightarrow{F_2} [\varphi_i f(\mu)\delta_1]$

la forme f_k est supprimée, elle perd code morphologique, et valeur des attributs lexicaux

si les conditions ci dessous sont remplies :

$f_k \in E_1$ et δ_1 a des attributs lexicaux conjonctionnels

ou $f_k \in E_3$ et δ_1 " " " ou progressifs

ou $f_k \in E_4$ et δ_1 " " fréquentatifs

ou $f_k \in E_2$ et δ_1 " " indéfinis.

C O N C L U S I O N

Entreprendre l'étude morphologique du Japonais en vue de la traduction automatique, c'est tenir un pari. Parier que le découpage de la phrase apportera suffisamment d'informations morphologiques exploitables dans la suite du programme syntaxique et sémantique.

Il faut bien constater que les informations obtenues sont très faibles :

- le verbe n'a pas de personne
- le pluriel n'existe pas en général, pas plus que le masculin et le féminin
- les adjectifs ont très souvent la fonction des verbes
- la particule du sujet est le plus souvent la même que celle de l'objet.

Bref, à partir de la langue japonaise, on pourrait presque élaborer la philosophie orientale où "tout est dans tout" ce qui apparait même dans l'aspect de la phrase où la notion de mot est floue, la phrase formant bien une entité tout entière.

Modestement on doit reconnaitre que vouloir morceler la phrase japonaise est une gageure difficile à tenir pour un occidental c'est raisonner avec un esprit cartésien sur une langue orientale.

BIBLIOGRAPHIE

C. HAGUENAUER : Morphologie du Japonais moderne

S. KUNO : A preliminary approach to Japenese-English Automatic Translation (First International Congress on Machine Translation of Language- Teddington, Septembre 1961)

S. LAMB and W. JACOBSON : A high-speed-capacity Dictionnary System (Mechanical Translation - Vol. 6 Nov. 1961)

W.P. LEYMANN : A grammar of formal written-Japenese

I. SAKAI : "Syntax in Universal Translation" First International conference on Machine Translationf of languages. Teddington 1962.

TOKEHIKO YOSHIHASHI and EDWINO.REISCHAUER : Elementary Japenese (Harvard-Yenching Institute)

LEHMANN and LLOYD FAUST : A grammar of Formal Written Japenese (Harvard-Yenching Institute)

B. VAUQUOIS : Langages artificiels, systèmes formels et traduction automatique (NATO advanced study Institute, July 1962)

G. VEILLON : Consultation d'un dictionnaire et analyse morphologique en traduction automatique (Thèse de 3ème cycle- Université de GRENOBLE - Juin 1962)

S. YAMADA : Etude Morphologique du Japonais - Document G-1000-1 C.E.T.A.-G Avril 1961

Etude Morphologique du Japonais - Document G-1000-2 C.E.T.A-G Juin 1962

S O M M A I R E