Tentative de formalisation algorithmique de la démarche du phonologue

Michel Jacobson

Tentative de formalisation algorithmique de la démarche du phonologue
Un outil d'aide à la formulation d'hypothèses phonologiques

Michel Jacobson

LACITO/CNRS
8, rue Guy Moquet – 94800 Villejuif
Tél.: ++33 (0)1 49 58 27 21 - Fax: ++33 (0) 1 49 58 37 79
Mél: jacobson@idf.ext.jussieu.fr - http://lacito.vjf.cnrs.fr

ABSTRACT

We present a formal computerized model of a particular linguistic theory, functional phonology -- a theory which is often criticized precisely for its lack of formalization. This theory proposes on the one hand a general framework for the expression of phonological phenomena and on the other a model for a discovery procedure for phonological units. In formalizing this theory explicitly, we have arrived at (1) a formalism for the expression of data and hypotheses and (2) a computer program emulating the functionalist methods of phonological analysis.

In the paper, we present the principal data structures used and the procedures which we have designed to process them. Methodological obstacles which we have faced in implementing the model are discussed.

1. PRESENTATION

Le linguiste phonologue quand il débute l'analyse d'une langue qu'il ne connaît pas, ne dispose que de ses connaissances en linguistique ou en phonétique générales. Il connaît un certain nombre d'indices phoniques qu'il sait pouvoir mesurer ou estimer. Il sait aussi que certain de ces indices ont participé à la définition de traits distinctifs dans des langues déjà décrites. Ce cadre théorique minimal peut se formaliser par une grille d'observation que le phonologue va utiliser pour décrire de nouvelles langues.

Dans le domaine de la typographie, l'alphabet phonétique international (A.P.I) a été une tentative de normalisation d'une telle grille, permettant la transcription des langues orales. Dans le domaine informatique, il a fallu trouver un équivalent formel pour exprimer toutes ces conventions. SAMPA [Gib97] est une de ces tentatives de codification informatique de l'A.P.I., supplantée plus récemment par l'Unicode [Uni00]. Mais ces deux tentatives ne répondent qu'à une partie du problème, celui du codage informatique d'un code typographique. Il manque aux deux la possibilité d'exprimer de manière non ambiguë la structure du système de notation qu'ils implémentent, c'est-à-dire de définir les dimensions phonologiques qui sont représentées typographiquement par l'organisation des unités, en lignes colonnes et tableaux dans l'A.P.I.. Nous verrons plus loin la proposition de formalisation que nous faisons de cette structure.

Une fois la grille d'observation établie, celle-ci constitue un outil de notation, et va permettre au phonologue de transcrire la langue qu'il étudie. Les théories linguistiques sont relativement neutres quant au système de notation qu'elles emploient. Globalement il y a un accord entre elles pour utiliser des systèmes de type alphabétique (A.P.I. ou autres). Les caractères représentent alors des regroupements d'événements et d'états stables caractéristiques. Elles diffèrent plus largement par l'analyse qu'elles font de ces segments alphabétiques. Ainsi les structuralistes, les fonctionnalistes et une partie des générativistes décomposent les segments en matrices de traits binaires ou non, en inventaires figés ou non. L'approche adoptée par la théorie du « charme et du gouvernement » [Kay88] décompose les segments en combinaisons d'éléments (matrices de traits entièrement spécifiées). Enfin les théories auto-segmentales vont, elles, analyser chaque segment sur plusieurs lignes à la fois (tires) représentant des dimensions d'analyses qui ne correspondent pas toujours à des traits, etc. Nous avons donc besoin dans une formalisation informatique d'expliciter cette ré-analyse spécifique des segments notationels en primitives de la théorie. Cette ré-analyse étant propre à chaque théorie, il n'était pas envisageable de trouver un formalisme qui convienne à toutes. Nous avons donc choisi de travailler uniquement dans le cadre d'une théorie particulière.

La formalisation informatique d'une théorie linguistique passe par l'explicitation de toutes les primitives et de tous les concepts qu'elle manipule. Certaines théories proposent un cadre général pour exprimer les phénomènes d'ordre phonologique observés, d'autres modélisent aussi la démarche de découverte des unités linguistiques.

Dans le travail que nous présentons ici, nous allons examiner les concepts et la méthode de la phonologie fonctionnaliste [Mar56]. Celle-ci bien que déjà ancienne n'a donné lieu qu'à de rares tentatives d'automatisation. Nous proposons dans ce qui suit

d'examiner cette théorie par le biais de la création conjointe d'un formalisme d'expression et d'un logiciel qui implémente la démarche des phonologues de cette école sous forme d'algorithmes .

2. MODÉLISATION INFORMATIQUE

La modélisation des procédures de découverte des unités tout comme celle de mise en lumière des phénomènes phonologiques (variations combinatoires, neutralisations,…) est explicitée sous forme d'algorithmes, puis a été implémenté avec un langage de programmation pour former le logiciel décrit ici.

La structure des données est, consignée dans une syntaxe formelle (DTD) qui définit à la fois les données d'entrées (le corpus), toutes les hypothèses (y compris celles définissant le système de notation utilisé pour le corpus) ainsi que les résultats.

2.1. Les données

Nous situons la constitution du corpus dans une étape phonétique ou pré-phonologique. Il s'agit de mesurer au cours du temps, dans un processus d'échantillonnage, les différents événements phonétiques ou indices. Une fois ce travail effectué, les données observées forment un corpus constitué d'une liste d'items lexicaux avec pour chacun d'eux, sa description phonétique, un identifiant de son sens et éventuellement sa position dans un fichier d'enregistrement. Cette description peut s'étendre bien sûr à de nombreux autres champs tels que ceux que l'on pourrait s'attendre à trouver dans un dictionnaire digne de ce nom (partie du discours, étymologie, etc.), mais seuls les premiers cités sont traités effectivement dans le programme.

```
<SIGNE id="w123">
        <SIGNIFIANT>pakt</SIGNIFIANT>
        <SIGNIFIE>pacte</SIGNIFIE>
        <AUDIO start="34.32" end="35.10">
</TRAIT>
```

Figure1 : Exemple d'entrée minimale d'un corpus.

2.2. Les hypothèses

La structure du système de notation qui constitue notre grille d'analyse est bien sûr la première de nos hypothèses. Elle doit être explicitée en indiquant toutes les relations qui existent entre les segments et les objets manipulés par la théorie. Par exemple : Chaque trait est déclaré 1) soit en listant tous les segments qui sont caractérisés par lui

```
<TRAIT name="occlusif">
        <INVENTAIRE>p t k b d g …</INVENTAIRE>
</TRAIT>
```

Figure2 : Exemple de déclaration du trait 'occlusif'.

2) soit en listant les contraintes sous forme de conditions sur les traits déjà déterminés.

```
<TRAIT name="affriqué">
        <CONDITION>(occlusif)^(fricatif)</CONDITION>
</TRAIT>
```

Figure3 : Exemple de déclaration du trait 'affriqué' .

Les relations entre les traits sont elles aussi notées sous forme de dimensions oppositionnelles spécifiant le type de relations qui les lient.

```
<DIMENSION name="degre d'apperture" ordre="true">
        <TRAITS>ouvert semi-ouvert… fermé</TRAITS>
</DIMENSION>
```

Figure4 : Exemple de déclaration d'une dimension ordonnée.

Enfin les éléments notationnels de l'A.P.I. dont la présence indique une valeur et dont l'absence entraîne la valeur négative correspondante sont déclarés comme des éléments diacritiques.

L'approche fonctionnaliste utilise la notion de syllabe, notamment dans la description des contextes de pertinence, mais sans la définir comme un objet de la théorie. Or il est impossible dans le cadre que l'on s'est fixé d'utiliser une notion sans la définir de manière explicite. Comme la notion de syllabe est fréquemment utilisée par les phonologues de cette école nous avons choisi d'emprunter sa définition à d'autres approches théoriques. En pratique, elle est exprimée dans les hypothèses sous forme de contraintes sur l'enchaînement des types de segments. Ces contraintes peuvent être synthétisées par un automate à états finis (ou canon syllabique).

```
<SYLLABE type="canon">T?(OG?|IL?G?)?(R|VF))</SYLLABE>
```

Figure5 : Exemple de canon syllabique utilisé pour le proto-tamang. (Les lettres 'TOGILRVF' représentent respectivement les classes : Ton, Onset, Glyde, Initiale, Rhyme,Voyelle et Finale.)

Une autre solution consiste à définir un ordre des structures préférentielles comme on pourrait le faire dans le cadre de la théorie de l'optimalité [Pri93].

Une dernière hypothèse est la définition des contextes de pertinence. Ceux-ci sont eux exprimés comme précédemment sous forme de contraintes (clauses logiques exprimées sous forme normale conjonctive) qui portent à la fois sur la position des segments entre eux ou au sein de la syllabe et sur la composition en termes de traits de ces derniers.

2.3. Les traitements

Le logiciel décompose la démarche du phonologue fonctionnaliste en un certain nombre d'étapes successives qui sont parfois répétées de manière cyclique.

· Les deux premières étapes sont la segmentation et la permutation. Elles vont opérer des regroupements de segments en fonction respectivement du système de notation et de la combinatoire utilisée dans la langue. L'enchaînement de ces deux étapes sera opéré en boucle jusqu'à ce que toutes les unités minimales obtenues soient permutables ou jugées telles.

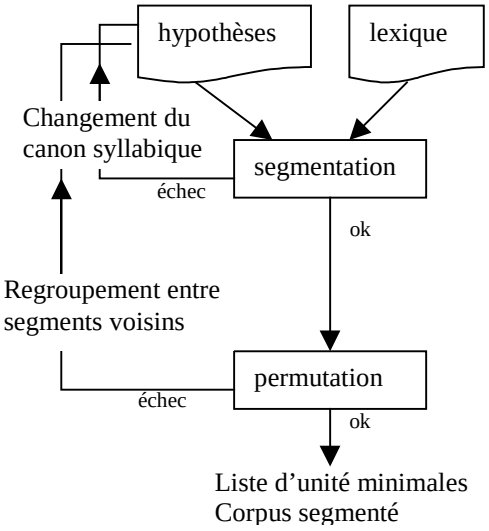

Figure6: Enchaînement des opérations de permutation et de commutation.

Si un segment ne commute pas, c'est-à-dire si l'on peut observer un conditionnement entre ce segment et son environnement proche, il est alors concaténé avec ses plus proches voisins puis le corpus est à nouveau segmenté et les nouveaux regroupements testés par l'opération de permutation. Cet enchaînement se poursuit jusqu'à ce que tous les éléments segmentés soient permutables. Cette optique, adoptée dans une première version du logiciel, a été abandonnée au profit d'une démarche assisté par le linguiste. En effet, en général, ni le corpus, ni même la langue ne sont suffisants pour que tous les éléments permutent.

· L'algorithme de commutation permet d'examiner les influences des unités minimales entre elles en observant leurs comportements à l'intérieur des contextes où elles s'expriment. Dans un premier temps les unités issues de l'étape précédente sont ventilées dans des inventaires contextuels en fonction de leurs conditions d'apparition dans le corpus. Une fois ces inventaires établis on cherche à valider, au sein d'un même inventaire, le plus d'oppositions possibles entre ses unités. C'est la présence ou non de « paires minimales » dans le corpus qui va permettre cette validation. Si la commutation est possible entre deux unités c'est qu'il s'agit de deux phonèmes différents et les traits qui les identifient l'un par rapport à l'autre ont alors une fonction distinctive. Cette étape permet de définir en termes de traits distinctifs la place que les unités occupent au sein des sous-systèmes contextuels.

· Enfin le dernier algorithme est celui du rapprochement des inventaires. Celui-ci opère sur la base de la définition des unités issues de la commutation en termes de traits distinctifs. C'est au cours de cette étape que l'on pourra distinguer les cas de variantes combinatoires, les cas de neutralisation, les assimilations, les distributions lacunaires, etc.

3. IMPLÉMENTATION LOGICIELLE

Le logiciel mis au point pour implémenter l'analyse de la méthode fonctionnaliste est écrit en Java. Il ne sait traiter qu'un type particulier de fichier XML, ceux qui respectent la syntaxe (ou DTD) définie à cet effet. Quelques facilités ont été ajoutées au programme pour consulter directement les enregistrements audio afin d'aider à la confrontation de l'analyse et des données.

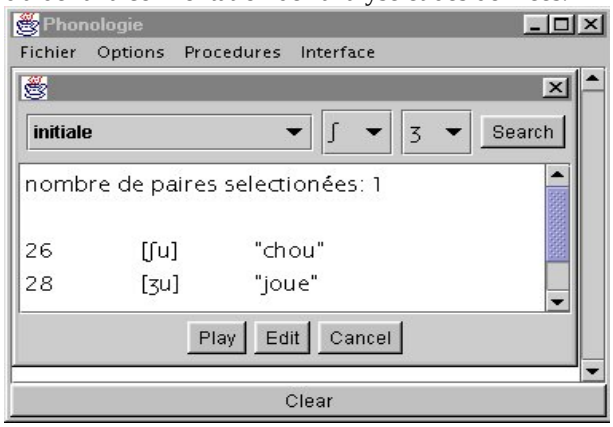

Figure7: Image du logiciel (extraction des paires minimales entre la chuintante sourde et la chuintante sonore du français en contexte initial absolu).

Le formalisme choisi pour coder le corpus, les hypothèses et les résultats nous permet, entre autres avantages, 1) de structurer explicitement ces derniers 2) de noter n'importe quel caractère existant dans Unicode (potentiellement plus de 65000 caractères avec notamment tous ceux de l'API), 3) d'être indépendant de la plate-forme, 4) de bénéficier de tous les développements faits dans ce domaine (parsers, éditeurs, langages de requêtes, etc.).

Le but de cet outil est qu'il reproduise la méthode fonctionnaliste telle que nous avons pu la modéliser. Il est par exemple capable de lister tous les phonèmes d'une langue par simple application des hypothèses sur un corpus et ceci sur la base des principes structuraux (cases vides, corrélations, etc.) et des principes fonctionnels (économie) de la théorie fonctionnaliste. Il a été conçu comme un outil de validation d'hypothèses, qui permet de faire des aller-retours entre des hypothèses et leurs validations dans le but d'ajuster et d'affiner ces hypothèses.

Une autre préoccupation dans la création de cet outil a été l'automatisation d'un certain nombre de tâches très répétitives et fastidieuses pour le linguiste, telles que le comptage des fréquences d'apparition de certains segments dans un contexte particulier ou bien la recherche de paires minimales.

Enfin cet outil se veut ouvert vers d'autres utilisations, telle que la confrontation des hypothèses avec les données brutes d'enregistrement (Il est a tout moment possible d'écouter un mot ou bien d'exporter le signal vers l'outil d'analyse phonétique de son choix). Il est aussi possible à moindre coût (c'est-à-dire au moyen de formatages ou de transformations XSL-T) d'exporter ou d'importer des ressources de type dictionnaires ou lexique qui utilisent d'autres types de codage explicites (SGML, XML, Lexware, Shoebox, etc.)

LIMITES ET PERSPECTIVES

Nous avons rencontré dans notre démarche de formalisation informatique un certain nombre d'obstacles méthodologiques. Nous pouvons citer parmi eux notamment l'absence de définition de la syllabe, une conception unidimensionnelle des traits et de leurs relations (par opposition à la théorie de la géométrie des traits). Enfin la phonologie fonctionnaliste est une phonologie du segment, unité postulée, alors que la véritable unité utilisée est le trait, ce qui rend difficile toute analyse dépassant le cadre du segment comme les assimilations à distance (les harmonies vocaliques).

Un des avantages de notre approche est la rigueur que l'informatique nous à obligé à apporter à notre formalisation. Cette même démarche pourrait être adoptée pour implémenter des modélisations issues d'autres théories phonologiques. Mais celles-ci proposent rarement une description du mécanisme de découverte des unités elles mêmes. Elles proposent uniquement un cadre conceptuel dans lequel on va pouvoir exprimer de différentes manières (sous forme de règles, de contraintes, de hiérarchies de contraintes, etc.) les mécanismes phonologiques mis en œuvre dans les langues. Il serait possible dans de tels cas d'aménager le programme pour que les hypothèses sur les mécanismes phonologiques soient directement renseignées par le linguiste. La fonction principale du logiciel serait alors uniquement d'effectuer le passage d'une écriture de type phonétique à une écriture de type phonologique et vice versa, ce qui est une fonction qu'il remplit déjà mais une fois les unités linguistiques déduites du corpus.

C'est dans cette optique que nous avons commencé a travailler avec la théorie de l'optimalité pour la modélisation de la notion de syllabe. Ce travail est d'autant plus aisé que cette théorie bénéficie entre autre avantage d'une formalisation assez poussée.

BIBLIOGRAPHIE

[Bra98] Bray, T., Paoli, J., Sperberg-McQueen, C.M. (Eds.), 1998. Extensible Markup Language (XML) 1.0. W3C Recommendation, 10 February 1998 (http://www.w3.org/TR/REC-xml).

[Gib97] Gibbon D., Moore R. et Winski R. (Eds). *Handbook of Standards and Resources for Spoken Language Systems*, Mouton de Gruyter, Berlin, 1997.

[Kay88] Kaye J., Lowenstamm J. et Vergnaud J-R. La structure interne des éléments phonologiques : une théorie du charme et du gouvernement. *Recherches Linguistiques*, 17. 1988. p 109-135

[Mar56] Martinet, A., La description phonologique avec application au franco-provencal d'Hauteville (Savoie), 1956, Genève, Droz.

[Pri93] Prince, A., et P. Smolensky. (1993). Optimality Theory: Constraint Interaction in Generative Grammar. RuCCS Technical Report 2. Piscataway, NJ: Rutgers Center for Cognitive Science, Rutgers University, and Boulder, CO: Department of Computer Science, University of Colorado.

[Uni00] Unicode Consortium, 2000. The Unicode Standard, Version 3.0. Addison-Wesley, Reading, MA (http://www.unicode.org).