

difficile. La constitution d'une grille parfaite de six cases de côté exige plusieurs heures de travail et «la consultation forcenée, sinon fiévreuse, d'un dictionnaire spécialisé». Il termine par une grille de sept cases de côté, avec une seule case noire au prix de «contorsions douloureuses» dont un «encrier renversé», c'est-à-dire un mot écrit de droite à gauche. Il constate également qu'il est très difficile de former des grilles de neuf cases de côté avec quatre cases noires et il lui semble impossible de placer moins de cinq cases noires pour une grille de dix cases de côté. Il admire enfin les grilles de 12 par 10 que Robert Scipion publie dans le *Nouvel Observateur* et qui ont régulièrement onze cases noires seulement.

Un autre problème voisin de la construction des mots croisés est la construction des anagrammes. Contrairement aux contrepèteries, qui sont assez faciles à créer ou à décoder pour un esprit entraîné tel celui de Luc Étienne, qui tenait la rubrique intitulée *L'album de la comtesse*, dans le *Canard enchaîné*, ou de Joël Martin, qui lui a succédé, les anagrammes constituent un exercice combinatoire des plus ardu : ce ne sont plus des sons ou phonèmes qu'il faut permuter, mais des lettres considérées comme des objets combinatoires avec lesquels on forme des mots. L'un des premiers auteurs ayant, de façon attestée, fabriqué des anagrammes est François Rabelais, qui signait Alcofribas Nasier. Plus près de nous, Boris Vian se transforma en Bison ravi. La construction d'anagrammes va nous montrer comment on peut automatiser la construction de mots croisés.

DICTIONNAIRES ET AUTOMATISATION

On a systématisé la méthode indiquée par Pérec, en utilisant un dictionnaire qui comporte, outre les féminins et les pluriels des substantifs et des adjectifs, les formes conjuguées des verbes, des noms de lieux, etc. Le dictionnaire est constitué d'une liste de mots triés dans l'ordre alphabétique. Par mot, on entend les mots de base, ou lemmes, et l'ensemble des flexions correspondantes : pluriels, féminins, formes conjuguées des verbes.

Supposons, par exemple, que nous voulions chercher dans un dictionnaire classique la signification du mot *gouleyantes*, dont la flexion est ici un féminin pluriel ; nous chercherions de fait à *gouleyant*, qui est le lemme et, dans ce cas, l'entrée du dictionnaire. La méthode que nous

avons utilisée, en collaboration avec l'ingénieur en informatique Joël Armengaud, pour constituer le dictionnaire-liste est générative : chacun des lemmes possède quatre indicateurs, qui donnent sa nature syntaxique, ainsi que les règles de formation des féminins et des pluriels. Ces indicateurs sont : la nature syntaxique (substantif, adjectif, adjectif numéral ordinal, pronom possessif, verbe, adverbe, locution, etc.) ; le genre (masculin singulier ou pluriel, féminin singulier ou pluriel, sans genre, etc.) ; la formation du féminin s'il en existe un (régulier, comme pour *grand, grande* ; ou générable, comme dans *beau, belle* ; ou encore irrégulier comme *lièvre, hase*) ; la formation du pluriel (régulier, comme *sol, sols* ; générable, comme *gavial, gavials* ; ou irrégulier comme *ail, aulx* ; invariable, etc.).

Dans le cas des lemmes qui ne sont pas des verbes, la génération des flexions se fait par programme, à partir des indicateurs décrits précédemment. Par exemple, à partir du substantif *vendeur*, on engendre *vendeurs, vendeuse, vendeuses, venderesse* (terme notarial), *venderesses*.

En ce qui concerne les verbes, nous avons mis au point un programme spécifique de conjugaison, qui tient compte du fait que, sur 9 000 verbes répertoriés, 500 seulement sont irréguliers ou défectifs. Les verbes irréguliers sont regroupés en sous-ensembles homogènes qui correspondent à une centaine de tables, tels les verbes en *ondre* (*tondre, pondre, fondre*, etc.). Les verbes réguliers du premier et du deuxième groupe ont leurs formes flexionnelles engendrées de façon totalement algorithmique, avec une décomposition en sous-groupes (*payer, je paye et je paie ; appeler, j'appelle* ; *je pèle ; déléguer, je délègue ; semer, je sème ; neiger, il neige ; etc.*).

pelle ; peler, je pèle ; déléguer, je délègue ; semer, je sème ; neiger, il neige ; etc.).

Quelques dictionnaires spéciaux ont été créés. Par exemple, pour les nombres de un à cent, exprimés de manière littéraire, par exemple *trente-et-un*, ou *nonante-neuf* ; pour les noms propres qui ont été arbitrairement entrés et forment une liste à part ; pour les noms de lieux (villes, régions, pays), qui forment également un dictionnaire particulier ; pour les prénoms, usuels ou exotiques.

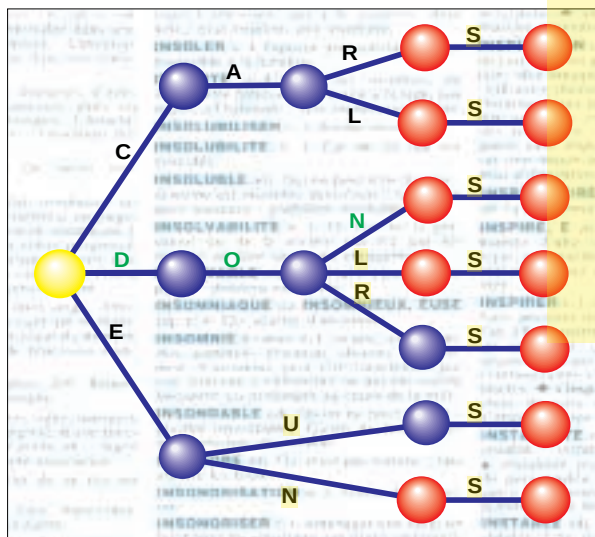
Pour obtenir le dictionnaire-liste complet, il suffit de regrouper les dictionnaires partiels précédents, de transformer les lettres minuscules en lettres majuscules et de trier la liste formée en éliminant les doublons. Naturellement on peut aussi ne retenir que certains des dictionnaires partiels, afin de constituer un dictionnaire sans noms propres ou sans formes conjuguées.

STRUCTURATION DES DONNÉES ET RECHERCHE

Tant pour la composition automatique de grilles que pour la recherche d'anagrammes, il est intéressant de savoir si un mot donné appartient au dictionnaire de référence, s'il existe des mots qui commencent par une séquence de lettres donnée, et s'il existe des mots qui se terminent par une séquence de lettres donnée.

Quand une liste de mots n'est pas triée, la seule méthode de recherche d'un mot particulier consiste à parcourir séquentiellement toute la liste. Cependant l'introduction d'un ordre de tri, alphabétique, par exemple, dans la liste des mots permet de réduire la complexité de recherche par une méthode dichotomique. Si l'on adopte une représentation arborescente de l'ensemble des mots (voir la figure 2), la résolution du problème est considérablement accélérée, parce que l'on diminue considérablement le nombre d'accès au graphe : par exemple, on n'accédera jamais aux mots commençant par un *m* lorsque l'on cherchera un mot commençant par un *e*.

En outre, l'exploration d'un arbre nécessite moins de mémoire, car les débuts de mots ne sont présents qu'une fois. Avec Yves Gattégno, ingénieur à la Faculté d'Évry-Val-d'Essonne, et avec l'ingénieur en informatique Stéphane Moreau, nous nous sommes inspirés des travaux de Andrew Appel, de l'Université Princeton, et de Guy



2. Arbre établi pour un dictionnaire qui serait réduit à : CAR, CARS, CAL, CALS, DOL, DOLS, DON, DONS, DORS, EN, ENS, EU, EUS. Sur la figure, les nœuds terminaux sont en rouge. On a indiqué en lettres vertes le chemin qui correspond au mot DON.

Jacobson, de l'Université de Pittsburgh ; nous avons utilisé des arbres et des structures encore plus condensées, les graphes acycliques orientés, où, dans les deux cas, les chemins représentent les mots ; dans ces structures, les lettres des mots ne sont plus des nœuds, mais des arêtes du graphe, de sorte que les débuts de mots similaires ne sont représentés qu'une fois. On les parcourt à l'aide d'algorithmes de parcours récursif.

LA RECHERCHE D'ANAGRAMMES

Pour représenter le dictionnaire par un arbre, on associe une branche à chaque lettre, de sorte qu'un chemin jusqu'à un nœud terminal représente un mot du dictionnaire. Pour chercher si un mot, par exemple *abacab* est un mot du dictionnaire, on part de la racine de l'arbre et l'on se demande s'il existe des mots commençant par la lettre *a*. Si aucune arête partant de la racine n'est indexée par un *a*, il n'existe pas de mot commençant par un *a* ; si l'on trouve une arête indexée par la lettre *a*, on cherche s'il existe des chemins qui permettent de passer du *a* à un *b*. On répète alors

l'opération pour les lettres suivantes : *a, c, a, b*.

L'ordinateur ainsi muni d'un dictionnaire peut-il dignement succéder à Rabelais ou à Vian ? Examinons la recherche d'anagrammes du mot *esprit*. Pour résoudre ce problème, nous utilisons une méthode de recherche récursive, qui permet, quand on est dans une impasse ou que l'on a terminé d'explorer une série de solutions, de revenir en arrière jusqu'au dernier embranchement de l'arbre pour lequel la clause de vérification n'était pas encore invalidée.

Pour rechercher les anagrammes d'*esprit*, mot que l'on indique à l'ordinateur dans une variable *Mot-initial*, nous procédons de la façon suivante. Nous réservons un espace mémoire pour les résultats, le *Mot-résultat*, vide au début de l'exécution du programme. Puis nous remplissons un tableau d'une ligne et de 26 colonnes (pour les 26 lettres de l'alphabet), le *Panier*, en stockant dans chaque case le nombre d'apparitions des diverses lettres dans le mot dont on cherche les anagrammes. Par exemple, le *Panier* associé à ESPRIT est :

00001000100000010111000000.

On regarde alors quelle est la lettre correspondant au nœud exploré. Puis on

regarde dans le *Panier* la valeur de la case associée à cette lettre. Si ce nombre est nul, on ignore le sous-arbre dont le nœud exploré est la racine et on passe au nœud suivant. Si le nombre examiné dans le *Panier* n'est pas nul, on ajoute la lettre correspondant au nœud exploré à la fin de *Mot-résultat* et on diminue d'une unité, dans le panier, le nombre contenu dans la case examinée ; on applique l'algorithme général à chacun des nœuds dont le nœud exploré est le père. De plus, on vérifie si le nombre de lettres de *Mot-résultat* est égal au nombre de lettres de *Mot-initial*. Si oui, on a dans *Mot-résultat* une anagramme de *Mot-initial*, anagramme que l'on affiche (à ce stade, toutes les cases du panier sont nulles).

Dans l'exemple de la recherche d'anagrammes du mot ESPRIT, le premier sous-arbre exploré sera celui qui correspond aux mots commençant par la lettre E, avec un *Panier* où la case de E est réduite d'une unité. Une fois qu'on aura exploré ce sous-arbre tout entier, on augmentera le compteur des E dans le *Panier* (on aura alors un *Panier* identique à celui du départ) et on tentera ensuite d'appliquer l'algorithme général au sous-arbre des mots commençant par la lettre suivante dont la case du *panier* n'est pas vide (la lettre I), avec *Mot-résultat* vide et le *Panier* égal à celui de départ, à l'exception de la case de la lettre I qui aura été réduite d'une unité. Puis on répètera l'opération avec les lettres P, R, S, T. L'avantage des algorithmes récursifs est qu'ils s'appliquent de la même façon à tout un arbre ou à un quelconque sous-arbre inclus dans l'arbre total. Ils s'appliquent notamment aux feuilles terminales, et la recherche d'anagrammes, par ce biais, se fait en un seul parcours de l'arbre dictionnaire.

Les anagrammes d'ESPRIT sont ainsi obtenues en une fraction seconde sur un ordinateur à base d'un processeur *Pentium* 120 MHz de la Société *Intel* : PETRIS, PISTER, PITRES, PRIEST, PRITES, REPITS, TRIPES.

On peut aussi étendre l'algorithme de sorte que le programme trouve des anagrammes de *Mot-initial* composées de plusieurs mots. À cette fin, on ajoute simplement un test sur le nœud exploré : ce nœud correspond-il à une fin de mot valide ? Si oui, alors on ajoute un espace à la fin de mot-résultat et on applique l'algorithme général avec ce *Mot-résultat* sur l'arbre dictionnaire complet (à partir de sa racine), avec le *panier* réduit de la façon habituelle. Sinon, on ne fait rien de plus.

Notre méthode d'exploration combinatoire est exhaustive et conduit, si l'on n'y prend garde, à une inflation d'anagrammes composées de mots qui existent tous, mais dont l'association manque

CABINET A M A R R E T	CABINET AGITONS MANETTE ACERAIT ROTAMES ENTIERE TSETSES	CABINET ALUNONS MASTITE AMERRIT RATATES ENTIERE TSETSES	CABINET AZEROLE MEDITER ARASAIS ROTASSE ELEISON TESTENT	CABINET AZEROLE MEGITES ARASAIT ROYASSE ELEISON TESTENT	CABINET AZEROLE MOTIVER ANISAIT RASASSE ELEISON TESTENT
CABINET AGAMIDE MARPAIN APERITE REMISAT ETAMAIT TETASSE	CABINET AGITONS MANETTE ACERAIT ROTATES ENTIERE TSETSES	CABINET ANATIDE MOLENES AMURENT RABATTE ELAITEE TASTEET	CABINET AZEROLE MEDITES ARASAIS ROTASSE ELEISON TESSENS	CABINET AZEROLE METIVES ARISAIT ROSASSE ELEISON TESSENS	CABINET AZEROLE MOTIVES ANISAIT RASASSE ELEISON TESSENS
CABINET AGAMIDE MARPAIN APERITE REMISAT ETAMAIT TISASSE	CABINET AGITONS MINETTE ARERAIT ROTAMES ENTIERE TSETSES	CABINET ANATIDE MOLENES AMURENT RABATTE ELAITEE TESTEES	CABINET AZEROLE MEDITES ARASAIT ROTASSE ELEISON TESTENT	CABINET AZEROLE METIVES ARISAIT ROSASSE ELEISON TERSENT	CABINET AZEROLE MOTIVES ANISAIT RASASSE ELEISON TERSENT
CABINET AGERATE MEDISES ARASANT RAMARDE ETENDRE TESTEES	CABINET AGITONS MINETTE ARERAIT ROTATES ENTIERE TSETSES	CABINET ANATIDE MOLENES AMURENT RABATTE ELAITER TASSERA	CABINET AZEROLE MEGITES ARASAIS ROYASSE ELEISON TESSENS	CABINET AZEROLE METIVES ARISAIT ROSASSE ELEISON TESTENT	CABINET AZEROLE MOTIVES ANISAIT RASASSE ELEISON TESTENT
CABINET AGERATE MEDISES ARASANT RAMARDE ETENDUE TESTEES	CABINET ALIBABA MENAGER AVALERA RIRIONS ENTATES TESSERE	CABINET ANATIDE MOLENES AMURENT RABATTE ELAITER TESTERA	CABINET AZEROLE MEGITES ARASAIT ROYASSE ELEISON TERSENT	CABINET AZEROLE MOTIVER ANISAIT RASASSE ELEISON TERSENT	

3. Georges Pérec n'avait pas réussi à construire une grille ayant pour potence les mots CABINET (1 horizontal) et CAMARET (1 vertical). L'ordinateur, lui, en a trouvé 28, qui sont indiquées sur cette figure.