QUAND « Z » VIENT-IL AVANT « A » ?
ALGORITHME DE TRI RESPECTANT LANGUES ET CULTURES

Alain LaBonté
informaticien-conseil
Secrétariat du Conseil du trésor
Gouvernement du Québec

Ce document a été présenté par M. Alain LaBonté le 16 août 1990 à La Nouvelle-Orléans à la 75ème conférence de SHARE. Il a été publié originellement en 1990 par le ministère des Communications du Québec. La version originelle a été revue et corrigée en août 1998 par l'auteur, M. Alain LaBonté, maintenant à l'emploi du Secrétariat du Conseil du trésor, qui a actuellement la responsabilité de ce dossier.

Contacter : Alain LaBonté

Note de navigation :
À chaque titre en gras correspond une figure en format gif, présentée dans l'ordre.

INTRODUCTION

Lorsque le titre de cet exposé m'a été proposé par SHARE, je me suis dit qu'il me faudrait d'abord expliquer de quoi il s'agit et de quoi il ne s'agit pas. Dès les premières années de l'informatique, les machines pouvaient effectuer avec une grande efficacité le tri des nombres et exécuter la multitude d'algorithmes intelligents, voire surnaturels, que nous avons inventés pour accélérer nos savants calculs. Ce ne sera pas le sujet de mon exposé. Je ne vous parlerai pas de la science du tri. De quoi s'agit-il alors? Je vais vous présenter un algorithme qui sert à structurer les données d'une manière qui permet aux ordinateurs d'effectuer le traitement naturel des données alphabétiques. Pour les machines, le traitement naturel signifie la possibilité de comparer, de rechercher, de trier et finalement de traiter des données de manière souvent complexe, sans qu'il soit nécessaire de leur faire subir un prétraitement chaque fois. Depuis longtemps, les ordinateurs peuvent traiter efficacement les nombres, à condition qu'ils soient stockés dans un format convenant au traitement naturel par les machines. Jusqu'à maintenant toutefois, les ordinateurs et leurs programmes se sont révélés peu efficaces dans les opérations de tri de données alphabétiques destinées à l'usage des humains, cela non seulement en chinois, en thaï, en arabe, en allemand, en espagnol et en français, mais également en anglais, comme je vous le montrerai bientôt. Mon exposé portera donc sur ces problèmes et sur les derniers progrès réalisés dans la recherche d'une solution qui, je crois, pourrait être appliquée prochainement aux traitements informatisés.

LE SYNDROME DU EMC

FIGURE 1

J'étais doublement heureux d'assister à la projection du film australien « Young Einstein », cette année, d'abord parce qu'il s'agissait d'un film divertissant, produit et interprété par un curieux jeune homme du nom de Yahoo Serious, qui n'était pas très sérieux en fait, et aussi parce que ce film était l'illustration parfaite d'un principe que j'affirme depuis des années. En effet, lorsqu'il consulte et parcourt une liste triée en ordre supposément : alphabétique, le commun des mortels néglige les particularités telles que les caractères spéciaux, les majuscules et minuscules, les signes diacritiques, etc., comme si ces détails n'existaient pas, mais il tient compte cependant du seul ordre de classement appris à l'école, c'est-à-dire l'alphabet. Lorsque le jeune Einstein, dans le film en question, montre la désormais célèbre formule « $E=mc^2$ » à son père, ce dernier lit simplement « emc ». Même s'il n'y comprend absolument rien, il lit ce qui lui est familier, les lettres, et tente d'en faire un mot, en laissant tout le reste de côté.

La simplicité du tri alphabétique peut sembler évidente, mais on se rend compte que ce n'est pas le cas quand on compare ce qui se passe dans différents pays, nous le verrons un peu plus loin. S'il n'y avait que des différences entre pays, on pourrait croire que le problème est relativement simple, car la plupart du temps, la majorité des gens qui s'occupent de tri alphabétique n'ont pas à travailler hors de leur milieu. Mais le problème reste entier même pour un anglophone qui ne connaît rien aux ordinateurs et qui vit dans le pays où sont apparus les premiers ordinateurs, c'est-à-dire ici même aux États-Unis. Voyons pourquoi.

UNE SYMÉTRIE NUISIBLE

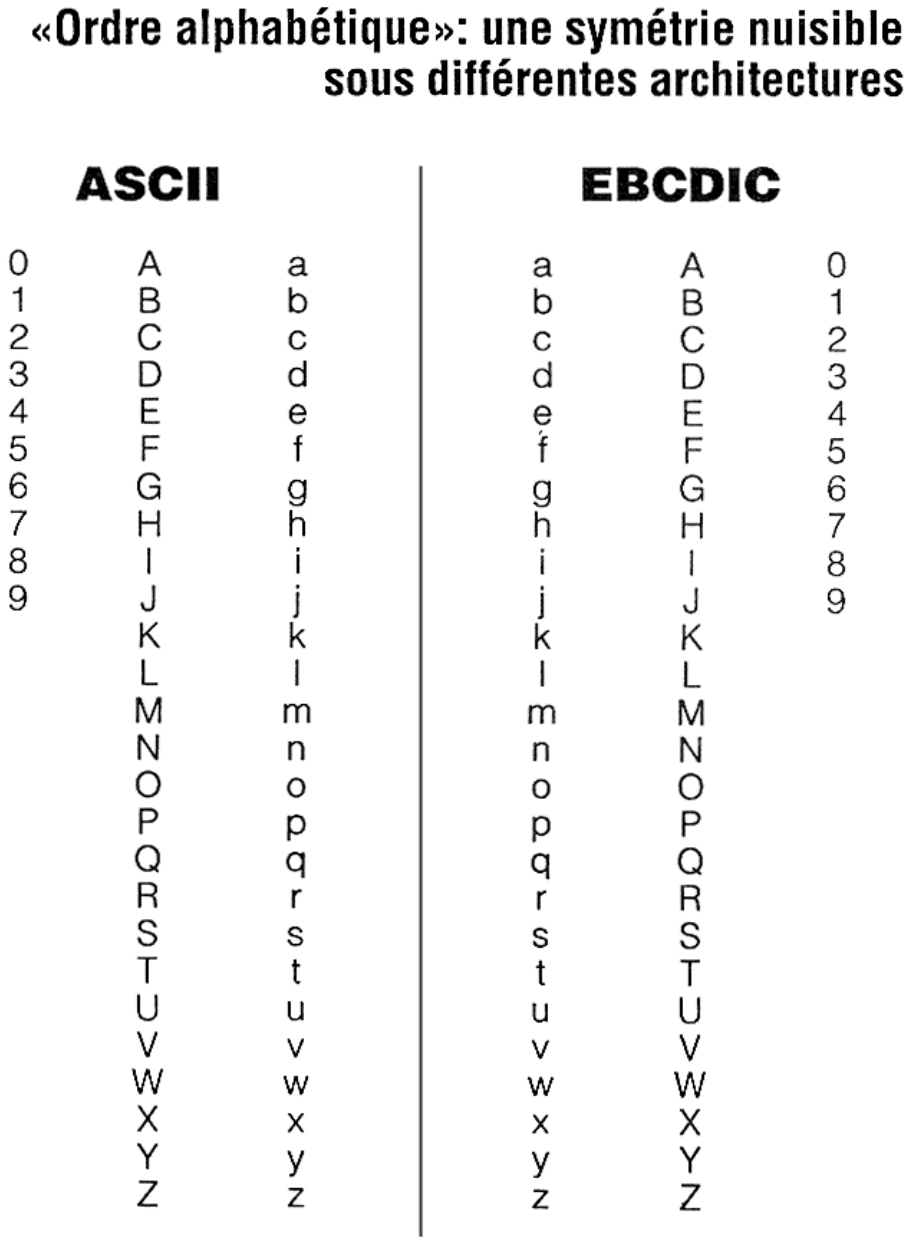


FIGURE 2

Le premier problème est causé par les tris qui se limitent aux valeurs de caractères binaires. Cette méthode produit des résultats qui diffèrent selon les architectures, comme c'est le cas avec les tables symétriques ASCII-EBCDIC, fort peu pratiques puisque ces tables de codes sont une image réfléchie l'une de l'autre.

TRIER AVEC ASCII OU EBCDIC ?

Tri

Code ASCII	Code EBCDIC
August	august
Vice versa	co-op
Vice-president	container
august	coop
co-op	August
container	Vice versa
coop	Vice-president

FIGURE 3

Si vos données contiennent des majuscules et des minuscules, vous aurez l'impression, si vous êtes une personne normale, que le tri est incorrect : il pourra arriver que vous ne trouviez pas ce que vous cherchez, et vous abandonnerez vos recherches. Par contre, si vous êtes un expert de l'informatique doué du flair d'un Hercule Poirot, vous arriverez peut-être à découvrir dans quel type d'environnement le tri a été effectué et vous pourrez ainsi retrouver l'information que vous cherchez.

Pour simplifier notre propos, cependant, posons comme hypothèse que les environnements sont tous identiques.

ORDRE ALPHABÉTIQUE EN ANGLAIS POUR LES LETTRÉS DE L'INFORMATIQUE

FIGURE 4

Les tris effectués à l'aide de codes de comparaison d'éléments hautement efficaces n'en donnent pas moins des résultats très étranges, comme peut le constater immédiatement tout individu normal qui n'est pas spécialiste de l'informatique comme nous. Et bien sûr, parce qu'il n'est pas naturel de chercher les mots en fonction des positions, caractère par caractère, le résultat obtenu est trompeur et incorrect même pour un spécialiste.

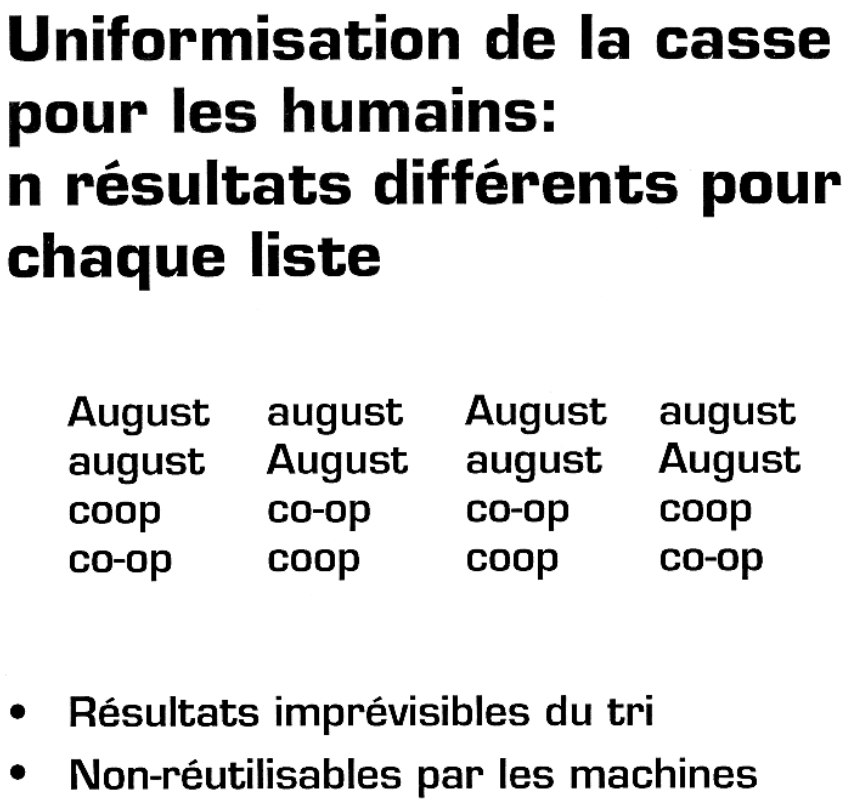


FIGURE 5

L'une des solutions possibles, qui vise à obtenir des résultats acceptables pour les humains, consiste à recourir à une seule table de conversion pour chaque tri. Une telle table attribue à chaque caractère une valeur normalisée pour chaque environnement, convertit toutes les lettres en minuscules et ne tient pas compte des caractères spéciaux. Cependant, les résultats ainsi obtenus ne sont pas prévisibles et ne peuvent être réutilisés, par exemple, dans la fusion de fichiers supposément triés. Avec ce genre de solution, un programme de tri n'effectuera en fait aucun tri dans certains cas extrêmes. Voilà une première constatation d'inefficacité.

L'AJOUT DES SIGNES DIACRITIQUES COMPLIQUE LE PROBLÈME

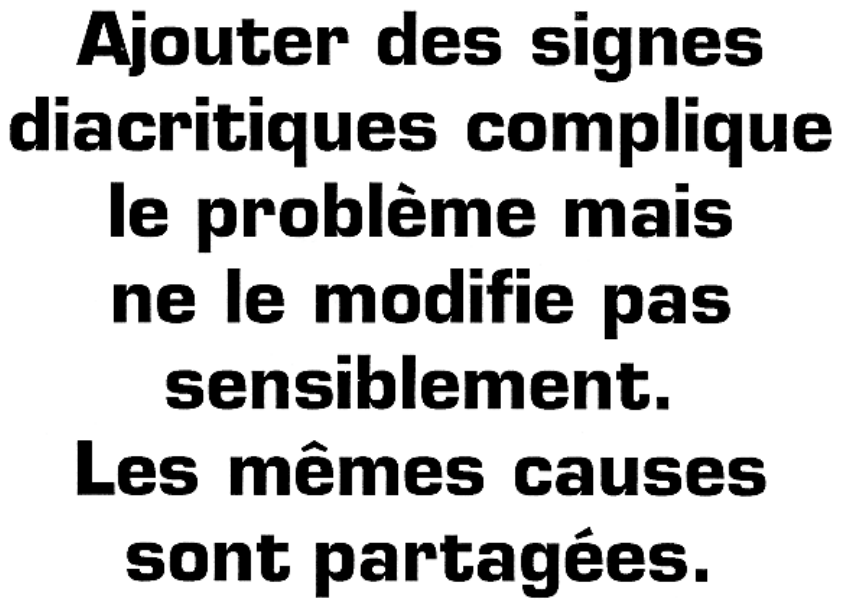


FIGURE 6

Jusqu'à présent, j'ai utilisé des exemples en anglais, et vous commencez peut-être à reconnaître un problème que vous connaissez sans doute tous, mais qui ne vous préoccupe pas vraiment, car nous avons tous pris l'habitude de travailler avec des applications et des environnements spécialisés. L'ajout des signes diacritiques, pour respecter la graphie propre aux autres langues, ne pose pas vraiment de problèmes additionnels. Cela ne fait que compliquer le problème existant, et les mauvais résultats obtenus procèdent des mêmes causes. Et comme si tout cela n'était pas assez, il y a aussi, en français, le problème des quasi-homographes, qui existent également en anglais malgré l'absence des signes diacritiques, dans le cas de doublets comme « co-op » et « coop ». Dans les opérations de tri détaillées, les quasi-homographes constituent en fait le problème le plus important dans toutes les langues, y compris le chinois (langue sans alphabet), comme je l'ai montré il y a quelques années au cours d'un exposé présenté lors d'un congrès de SHARE Europe.

PÊCHE PÉCHÉ PÉCHER PÊCHER...

FIGURE 7

À la suite de consultations avec de nombreux spécialistes, nous avons constaté que le français est la langue occidentale qui possède le plus grand nombre de quasi-homographes. Il existe d'ailleurs des règles sibyllines, dans le cercle des éditeurs de dictionnaires français, qui servent à établir l'ordre exact des quasi-homographes dans une liste triée : chaque accent se voit bien sûr attribuer un ordre de priorité; mais pour des raisons d'ordre linguistique que j'ai découvertes dans mes recherches, et sans aucun doute dans le but de simplifier les règles (même si cela paraît curieux à première vue), comme les accents, en français, possèdent généralement une valeur sémantique plus grande lorsqu'ils se trouvent à la fin des mots, l'accent discriminant est celui qui se trouve le plus près de la fin du mot. C'est ce qui explique pourquoi l'ordre «pêche péché pécher pêcher» paraît bizarre pour un anglophone. Lorsqu'on connaît la règle, cependant, il devient facile de résoudre le problème et ainsi, de nombreuses langues occidentales, y compris l'anglais, l'allemand, le néerlandais, l'italien, le portugais, etc. peuvent s'accommoder instantanément et sans difficulté d'un tri fait selon les règles du français. Si vous voulez bien patienter encore un peu, nous reviendrons sur cette question un peu plus tard.

Solution:

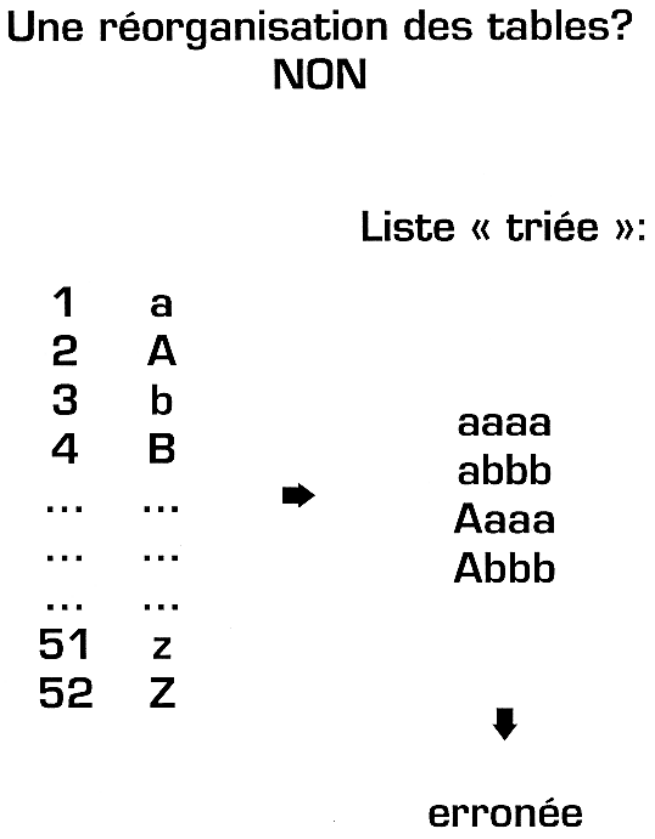


FIGURE 8

Pour résoudre le problème, beaucoup de spécialistes de l'informatique ont toujours cru qu'il suffisait de réorganiser les tables de codes, en groupant tous les «A» ensemble (minuscule et majuscule, avec ou sans signes diacritiques), les «B» ensemble, etc. Mais, contrairement à la croyance générale, cette solution n'en est pas une non plus. La véritable solution n'a rien à voir avec l'organisation des tables de caractères.

TROIS LOIS RÉGISSENT LES PROBLÈMES DE TRI DANS LE CADRE DE LA TECHNOLOGIE ACTUELLE

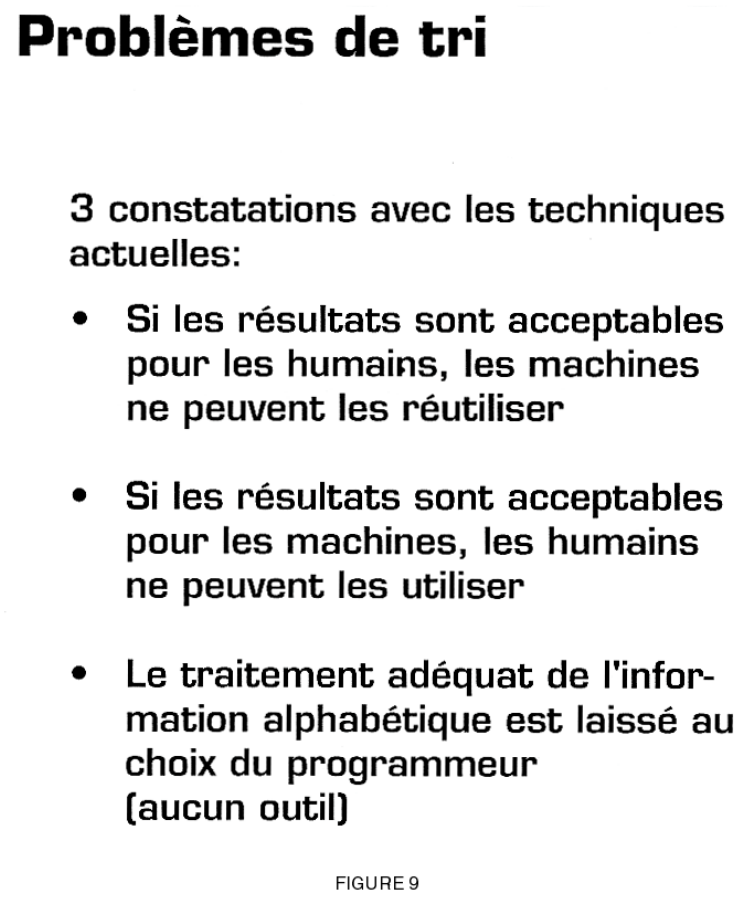


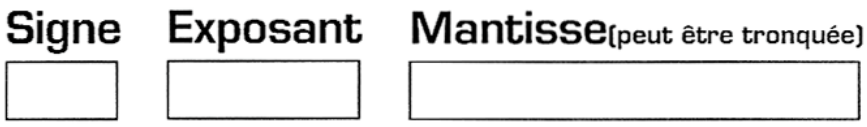
FIGURE 9

Des faits que je viens de présenter, le programmeur peut tirer trois lois régissant les opérations de tri à l'heure actuelle :

1. Si les résultats conviennent aux humains, ils ne peuvent être réutilisés dans les machines comme s'ils étaient triés selon un ordre absolument prévisible.
2. Si les résultats conviennent aux machines, ils ne sont pas utilisables par les humains non spécialistes de l'informatique.
3. Les langages de programmation, les méthodes d'accès et les systèmes de gestion de bases de données n'offrent au programmeur aucun outil qui lui permettrait de résoudre le problème.

Que devrons-nous faire pour trouver une solution au problème? Avant de vous présenter la solution que je préconise, j'aimerais vous rappeler pourquoi nous parvenons par contre à traiter les chiffres si efficacement.

Pour trouver une solution, analysons les propriétés des nombres à virgule flottante, utilisés pour le traitement de l'information



- Si la mantisse est tronquée ou même éliminée, il y a perte de précision, mais le nombre est encore utilisable

- Si l'exposant est éliminé et qu'il ne reste que le signe, on peut encore effectuer des comparaisons

FIGURE 10

Pour effectuer le traitement des nombres (ce pourquoi les ordinateurs ont d'abord été inventés), on a intégré aux ordinateurs une structure établie d'après les propriétés des nombres réels, qui permet d'en effectuer le traitement de façon adéquate. Cette structure, dont les variantes continueront d'exister jusqu'à ce que soit établie une norme vraiment universelle, s'appelle la structure numérique à virgule flottante pour les ordinateurs. Elle comprend trois parties : un élément pour le signe, un nombre déterminé d'éléments représentant l'exposant ou l'ordre de grandeur du nombre et, enfin, un certain nombre d'éléments, selon le degré de précision souhaité, représentant la mantisse, c'est-à-dire les ultimes détails caractérisant le nombre. Ce type de structure est tel que lorsque le nombre est tronqué n'importe où à partir de la droite, il y perd en précision mais il conserve toujours l'essence de sa valeur. Même s'il ne reste qu'un seul élément, les données gardent encore une certaine signification et restent comparables : on sait en effet si le nombre est positif ou négatif.

Pourquoi maintenant ne pas considérer aussi les propriétés des données alphabétiques, d'un point de vue similaire ?

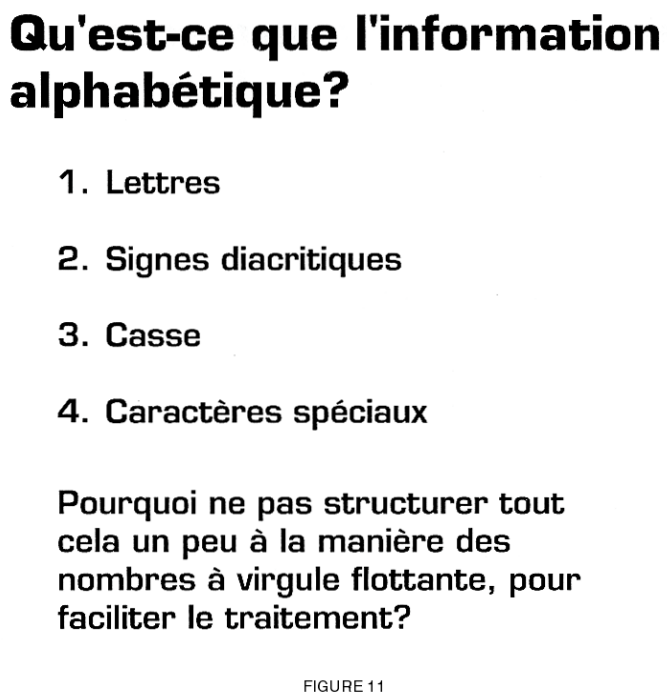


FIGURE 11

Comme nous sommes à la recherche d'une méthode de traitement d'éléments essentiellement culturels que sont les données alphabétiques, nous devons nous demander en quoi celles-ci consistent sur le plan culturel. Dans la mesure où nous appartenons à un groupe linguistique dont la langue est fondée sur un alphabet, les données alphabétiques sont constituées de lettres, dont nous avons tous appris l'ordre à l'école, selon la culture qui nous est propre.

Dans la plupart des langues dont l'écriture repose sur un alphabet, les données alphabétiques sont modifiées par des signes diacritiques, quoique dans une moindre mesure en anglais.

La représentation des lettres en majuscules ou en minuscules a moins d'importance sur le plan de la précision. Dans certaines langues, comme l'arabe, il existe des caractéristiques analogues où une lettre peut avoir plus de deux formes différentes, et leur usage ne correspond pas nécessairement aux majuscules et aux minuscules des langues occidentales. Certains parmi vous ignorent peut-être aussi que dans les langues occidentales utilisant l'alphabet latin, l'usage des majuscules et des minuscules ne joue pas toujours le même rôle : il varie d'une langue à une autre. Dans certaines langues qui utilisent un alphabet, cette distinction n'existe même pas. Ainsi, le latin classique ne faisait pas la distinction majuscules-minuscules et n'utilisait pas de signes diacritiques.

Les caractères spéciaux, pour leur part, constituent une autre caractéristique aux connotations culturelles imprécises, dont l'ordre de classement est ignoré de tous sauf des spécialistes de l'informatique qui travaillent avec les tables de codes.

Qu'arriverait-il si nous donnions à toute cette information une structure un peu similaire à celle des données à virgule flottante ?

NOUVELLE STRUCTURE POUR LES DONNÉES ALPHABÉTIQUES

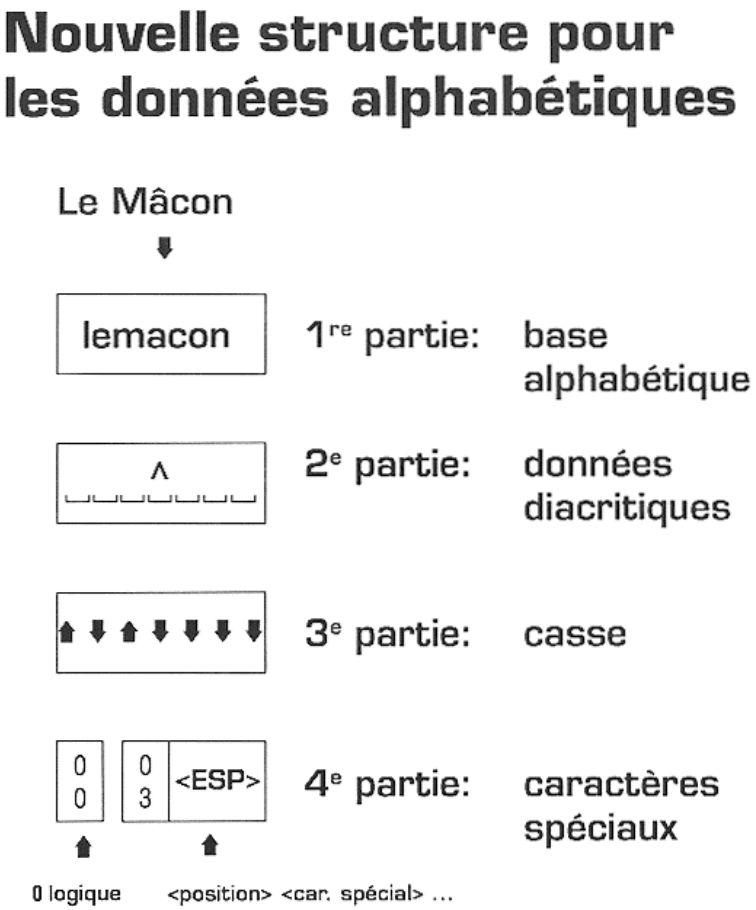


FIGURE 12

La nouvelle structure des données alphabétiques reprend les mêmes caractéristiques de traitement que les nombres à virgule flottante. Ce type de structure comprend, pour la plupart des langues à alphabet, quatre parties qui se suivent dans l'ordre de précision. Comme dans le cas des données à virgule flottante, si on part de la droite et qu'on supprime des éléments en allant vers la gauche, on n'y perd que de la précision, sans altérer l'essence même de la donnée.

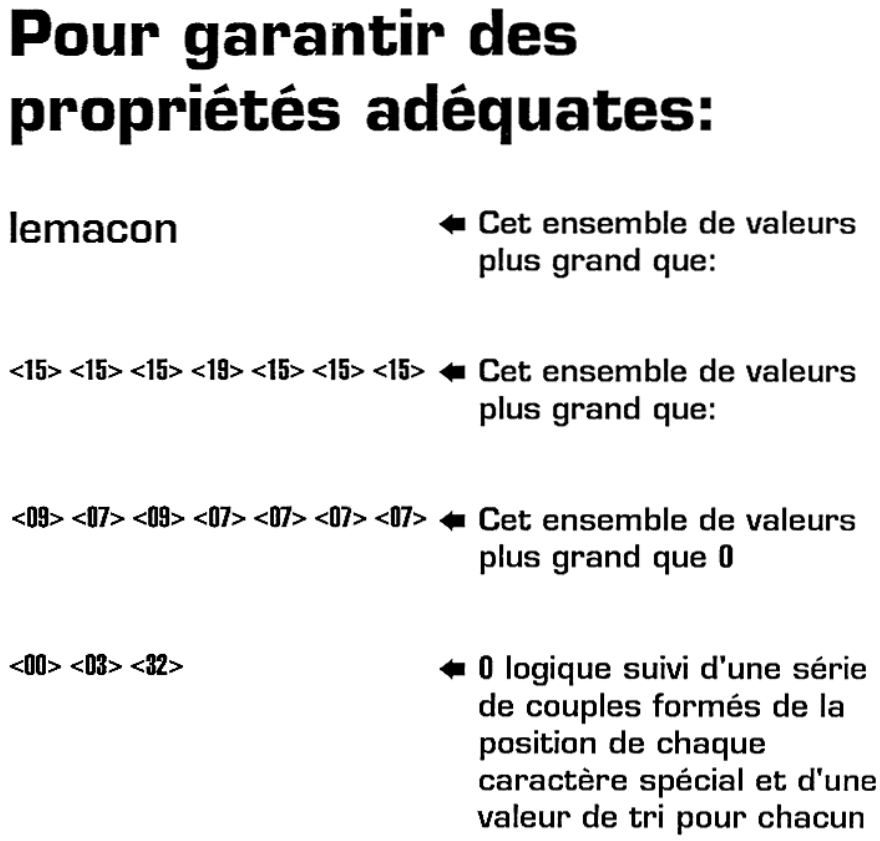


FIGURE 13

Pour que les propriétés soient identifiées adéquatement dans quatre parties de longueur variable, et afin d'assurer le traitement ultérieur des données, il faut prendre bien soin d'attribuer, à chacune des parties, des ensembles de valeurs décroissants. La première partie doit donc contenir un ensemble de valeurs toujours supérieur à l'ensemble de valeurs attribué à la deuxième partie. Le même rapport doit aussi exister entre la deuxième partie et la troisième. La quatrième partie, dont l'écart des valeurs est plus vaste, est délimitée au début par un zéro logique.

CES DONNÉES PEUVENT ÊTRE TRIÉES

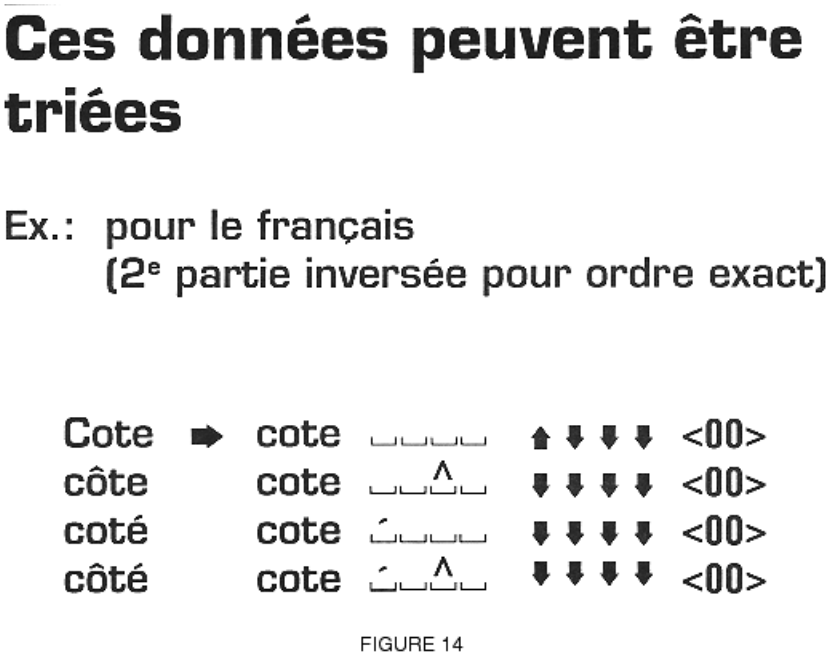


FIGURE 14

Aussi incroyable que cela puisse paraître, quand toutes les parties sont concaténées, les données ainsi traitées peuvent être triées ou comparées directement en fonction des valeurs binaires attribuées, sans autre forme de traitement. Ce qui est aussi extraordinaire, c'est qu'avec cet algorithme et cette nouvelle structure, il est possible de trier correctement les données alphabétiques et de classer un enregistrement alphabétique dans un index VSAM ou une base de données sans qu'il soit nécessaire de modifier les systèmes. Pour continuer d'utiliser les programmes existants, il suffit de réorganiser les données.

LISTE TRIÉE ET REPRÉSENTATION DE SES SOUS-CLÉS INTERNES

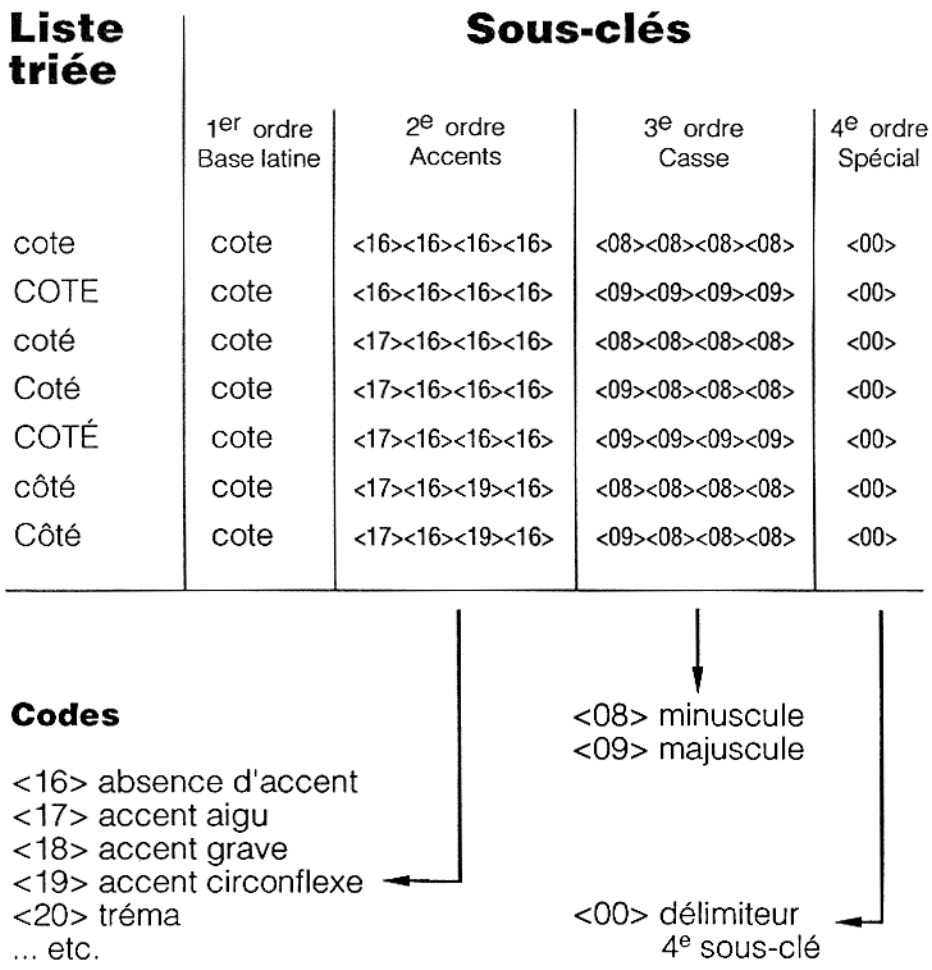


FIGURE 15

Si on prend une liste encore plus complexe de quasi-homographes français (contenant des accents, des majuscules et des minuscules), pour lesquels des ensembles de valeurs adéquats ont été attribués à chaque partie, et qui ont été triés selon les valeurs binaires assignées, le résultat reste totalement prévisible. De plus, il est également conforme au contenu des dictionnaires français, il peut être compris intuitivement par un utilisateur moyen qui ne s'arrête pas à l'analyse des détails, et il peut être réutilisé par une machine qui considérera que le tri est correct.

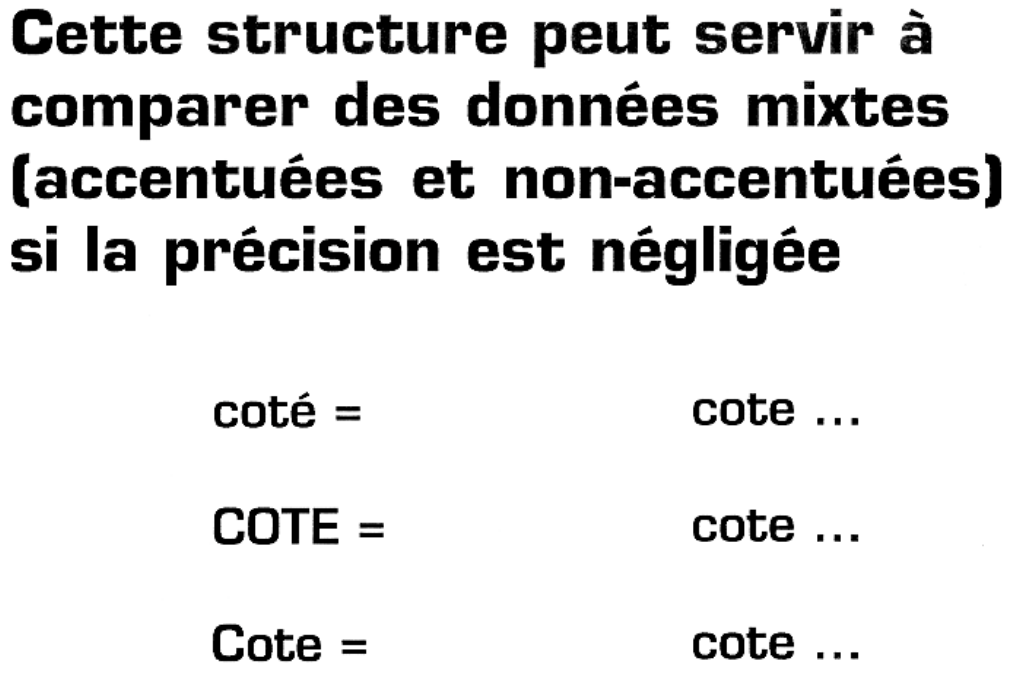


FIGURE 16

La solution proposée peut épargner beaucoup de travail et éviter bien des erreurs si la structure est stockée de manière permanente. Ainsi, quel programmeur n'a pas eu à convertir les majuscules et les minuscules avant de comparer des données alphabétiques ? Cette nouvelle structure des données, stockée en permanence, permettrait de résoudre définitivement ce problème.

Validation alphabétique ?

- Vérifier si la dernière partie contient des caractères « non valides »

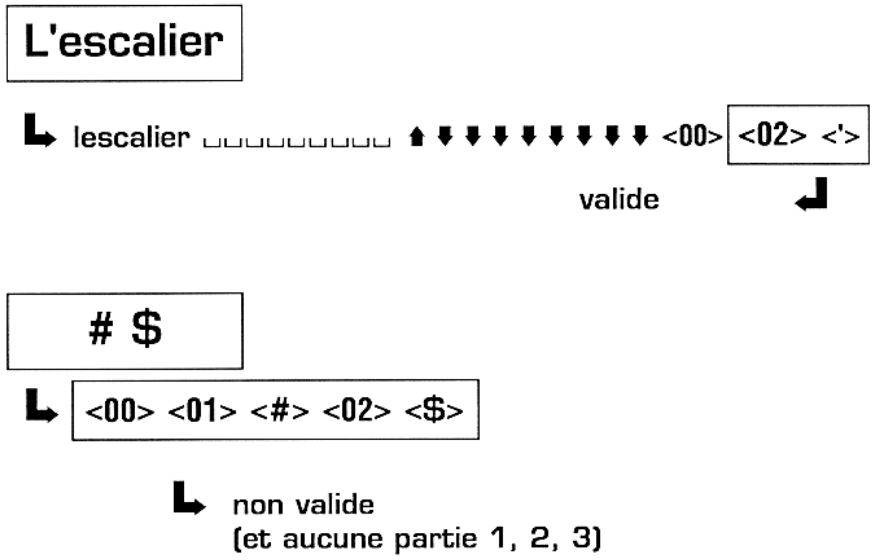


FIGURE 17

Désirez-vous abréger le test de validation alphabétique ? Il suffit de vérifier si la dernière partie contient des caractères non valides ou, ce qui est plus rapide encore, de voir si la structure contient une première partie. Cet effet secondaire n'était pas prévu aux premiers stades de la conception de la structure, mais j'ai la certitude que des algorithmes pourront être perfectionnés afin de permettre des traitements alphabétiques avancés. Parmi les autres effets secondaires, il est possible d'utiliser les données alphabétiques ainsi stockées à la manière d'une lingua fiança qui permet de passer en toute liberté d'une table de codes à une autre.

Si la structure est sauvegardée, l'information d'origine peut être reconstituée

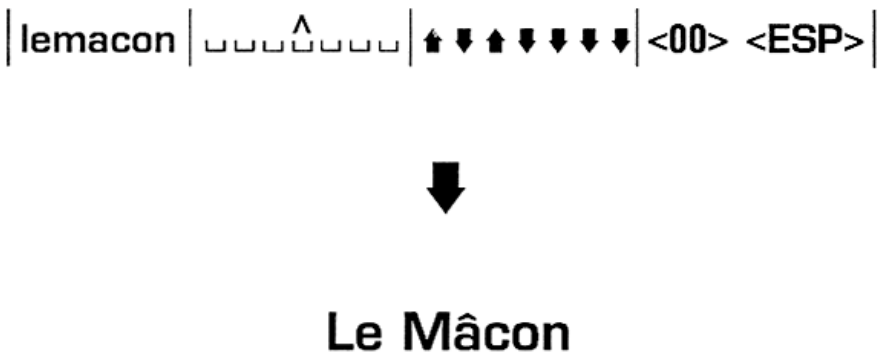


FIGURE 18

Ceux qui craindraient de perdre les données originales, dans le cas où seule cette structure serait conservée, n'ont pas à s'en faire : grâce à cette structure, les données originales peuvent être reconstituées avec exactitude car toute l'information nécessaire s'y trouve contenue.

En effet, s'il s'agissait simplement d'un nouvel algorithme destiné seulement aux programmes de tri-fusion, le problème du processus de comparaison resterait entier, de même que le problème des méthodes d'accès et du classement dans les bases de données. Par contre, la solution proposée est une solution systémique, appliquée à un environnement culturel bien défini.

Certains diront que le stockage de cette nouvelle structure exigera beaucoup de mémoire additionnelle. L'analyse de la quantité de mémoire requise montre que la structure exigerait deux ou trois fois plus de mémoire que les chaînes originales, sauf que l'utilisation de chaînes de longueur variable permettrait de gagner de l'espace, ce qu'il serait d'ailleurs possible de faire avec les structures traditionnelles.

TECHNIQUE DE RÉDUCTION

La structure codée peut être réduite et être encore propice au traitement

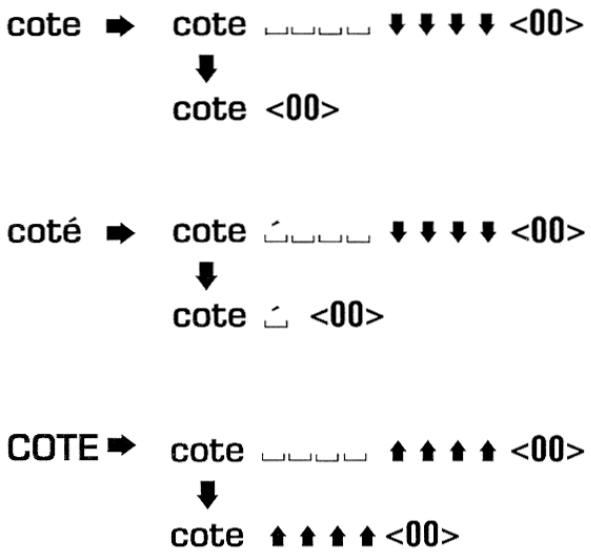


FIGURE 19

Grâce aux propriétés mathématiques des ensembles de valeurs utilisés dans chacune des parties de la structure, il est possible de déduire facilement une technique de réduction permettant d'économiser beaucoup d'espace mémoire. Si, dans la deuxième et la troisième partie de la structure proposée, les dernières valeurs sont égales aux plus petites valeurs que peut contenir chacune de ces parties, elles peuvent être supprimées. Cela signifie que pour une langue comme l'anglais, par exemple, il n'y aura presque jamais de deuxième partie, sauf dans le cas des mots contenant des accents, comme «résumé». Cela signifie aussi que si le mot ne contient que des minuscules, il n'y aura pas de troisième partie non plus. La dernière partie, qui en général ne devrait pas être très longue en ce qui concerne les données alphabétiques, peut aussi être réduite dans certains cas, comme lorsque la chaîne ne contient aucune lettre. Les données identifiant la position ne sont alors plus nécessaires, ce qui réduit de moitié l'espace mémoire exigé. Grâce à cette technique de réduction, il reste possible de traiter les données n'importe où, de les trier, de les comparer, etc. De plus, les données originales peuvent toujours être reconstituées puisque toutes les valeurs supprimées sont implicites.

Serait-il possible de compléter cette structure au moyen d'un ensemble de tables universel couvrant toutes les langues ? J'aimerais bien vous répondre par l'affirmative, mais cela reste absolument impossible. Pour certaines langues occidentales telles que l'anglais, le néerlandais, l'allemand, l'italien, le portugais, par exemple, il est toutefois possible d'utiliser le même modèle et les mêmes tables que pour le français, langue qui présente les difficultés les plus complexes.

CLASSEMENT DANOIS

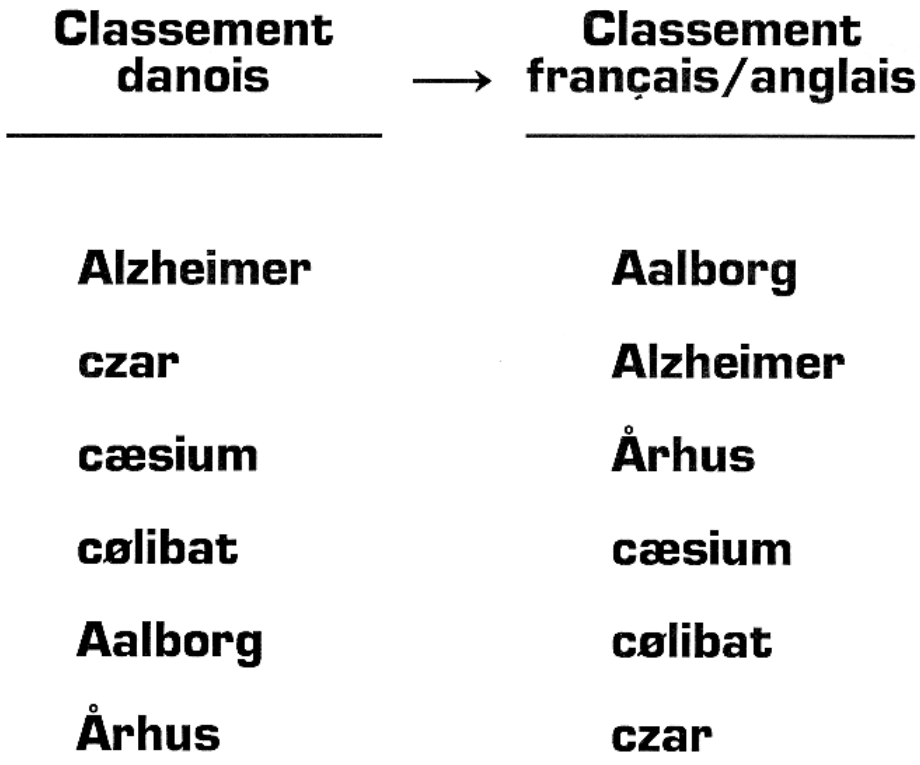


FIGURE 20

Dans les langues scandinaves, en danois dans notre exemple, les règles de tri sont différentes. Tous les caractères scandinaves nationaux sont généralement classés après « Z » : ainsi, pour un Danois, « Å » n'est pas une simple variante de « A », comme l'interprètent la plupart des autres langues occidentales, mais une lettre bien distincte qui est classée à la fin de l'alphabet. Même le double « A » (« AA ») est généralement classé après « Z » au Danemark, sauf dans un certain nombre de cas qui s'expliquent par la prononciation. Cela signifie que pour les langues scandinaves, il faut ajouter à l'algorithme fondé sur les tables de codes un simple sous-programme d'analyse contextuelle dont la création ne devrait pas poser de difficultés majeures. La structure même du modèle reste intacte.

EL ALFABETO ESPAÑOL

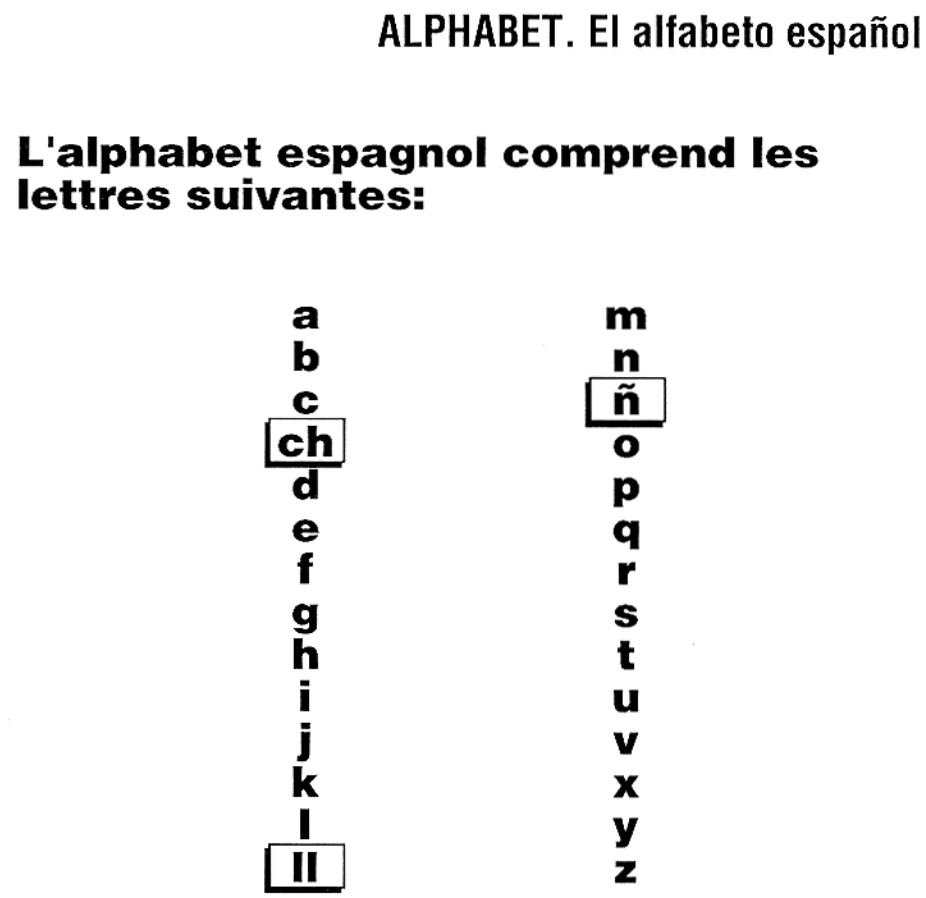


FIGURE 21

L'alphabet espagnol possède les mêmes caractéristiques que les alphabets scandinaves. Il comprend des lettres qui sont considérées comme lettres simples même si elles semblent être des lettres doubles dans les autres langues, comme « CH » et « LL ». En espagnol, il s'agit bien de lettres simples, au même titre que le double « A » en danois, pour des raisons à la fois phonétiques et historiques. Les linguistes ont établi il y a longtemps que la combinaison « CH », en espagnol, est la transposition de la lettre simple « khi » du grec.

CLASSEMENT ESPAGNOL

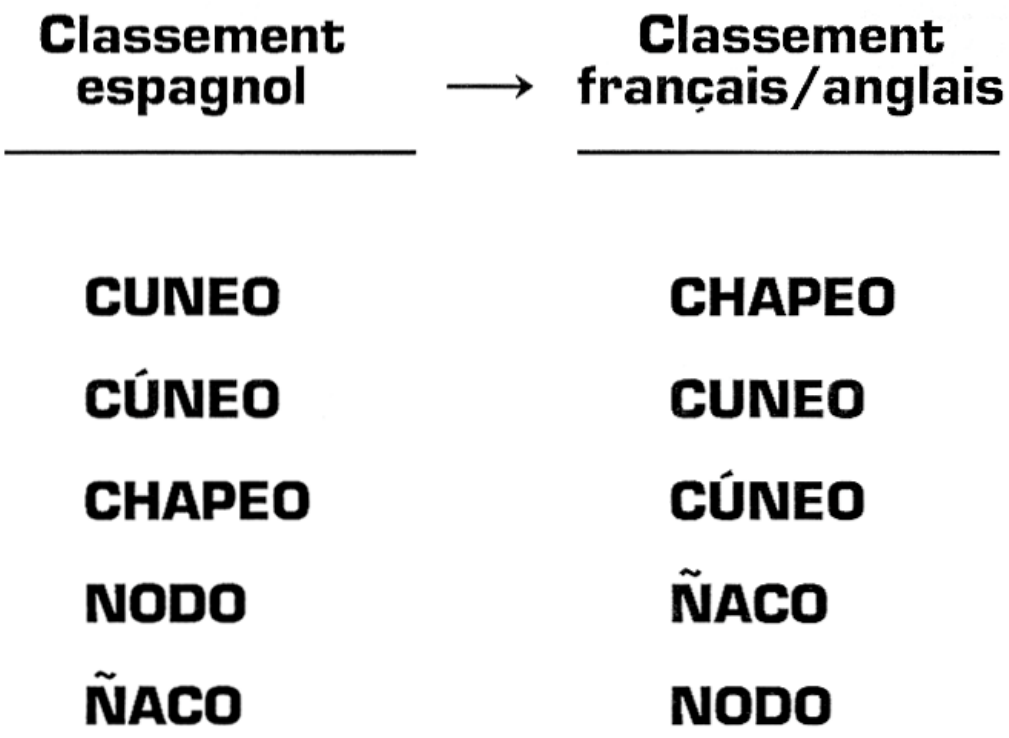


FIGURE 22

La lettre « N tilde (Ñ) » de l'espagnol est également différente de la lettre « N », comme le « Å » danois est différent du « A ». Par conséquent, pour obtenir un tri qui respecte l'esprit de la culture espagnole, les listes de mots espagnols doivent être triées différemment des listes de mots français ou anglais.

INTERPRÉTATION DES MÊMES LETTRES EN DIFFÉRENTES LANGUES

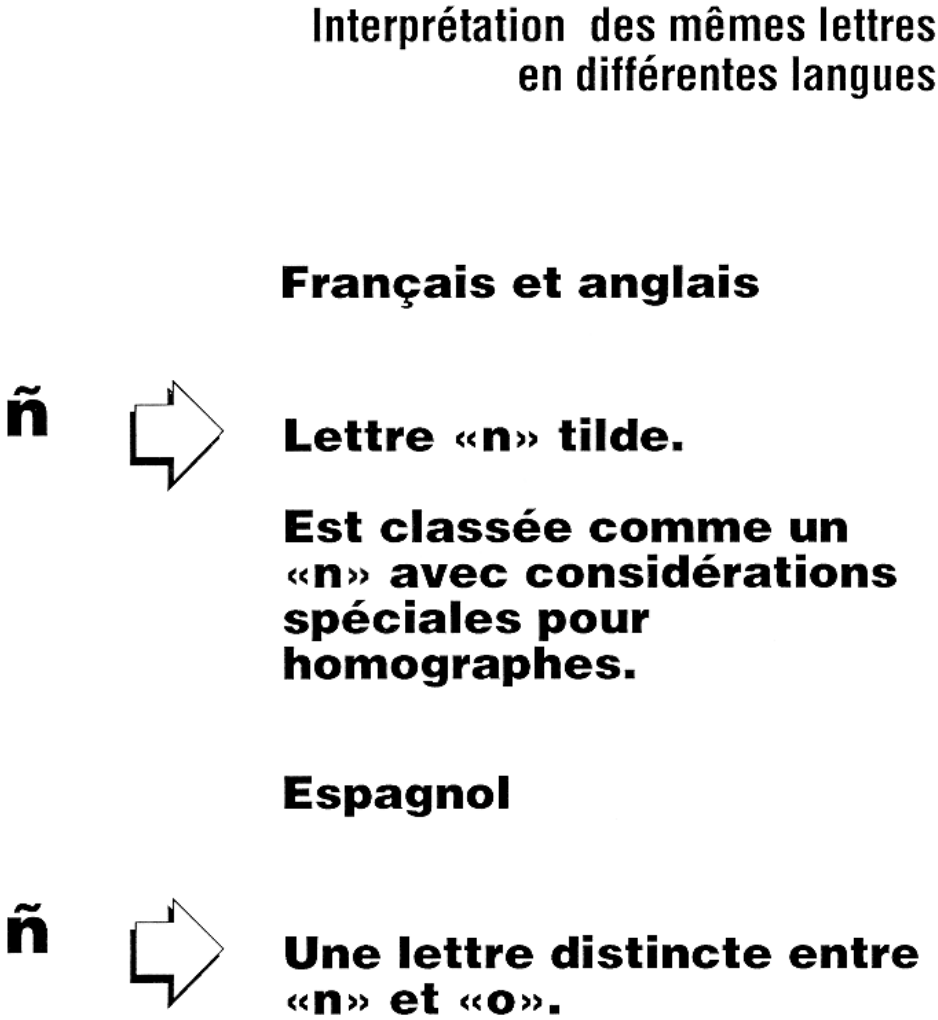


FIGURE 23

Si on considère les lettres simples, le « N tilde », par exemple, sera interprété différemment, comme je viens de l'expliquer, par des locuteurs anglophones ou francophones et par des locuteurs hispanophones ayant reçu leur formation scolaire dans un pays où l'on parle espagnol.

De même, l'allemand présente certaines difficultés dont la solution exige de faire un choix, car il existe deux tendances traditionnelles en ce qui concerne les voyelles infléchies en Allemagne.

PROBLÈMES POTENTIELS DU CLASSEMENT PHONÉTIQUE DANS CERTAINES LANGUES

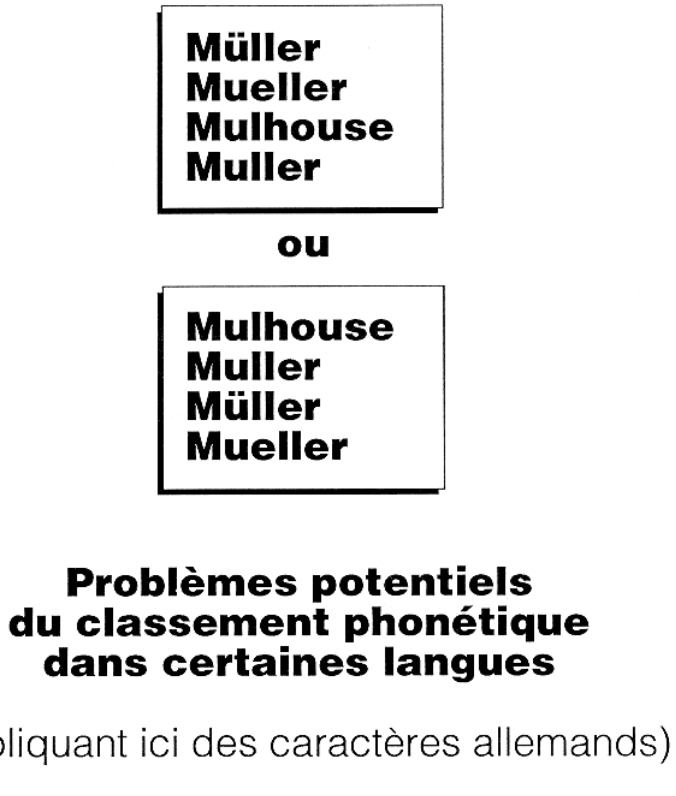


FIGURE 24

À cause de certaines restrictions d'ordre technologique, les Allemands ont pris l'habitude de remplacer les voyelles infléchies par la même voyelle suivie de la lettre « E » lorsqu'il est impossible de représenter le symbole umlaut au-dessus de la voyelle. Cette pratique a des conséquences sur les méthodes de tri, car certains noms de famille ont ainsi été modifiés, et dans les annuaires téléphoniques on a contourné la difficulté en triant les noms d'après des règles particulières qui ont changé au fil des ans. Par exemple, si la règle dicte que « UE » doit être trié comme s'il était écrit « U umlaut », le résultat sera légèrement différent de celui qui sera obtenu si la règle exige de trier « U umlaut » comme s'il était écrit « UE », ce qu'on voit dans la présente illustration. Une autre pratique, qui est suivie dans les dictionnaires mais qui n'est pas montrée ici, consiste à ne pas faire de transformation en vue du tri, ce qui la rend entièrement compatible avec les méthodes de tri utilisées en français et en anglais.

Nous savons maintenant comment effectuer les opérations de tri et le traitement des données alphabétiques en respectant l'esprit de chaque culture, mais il nous reste encore à mettre ces connaissances en pratique. Nous savons aussi que lorsqu'un fichier est dit trié, il est nécessaire de savoir selon quelle méthode le tri a été effectué. Pour assurer l'interchangeabilité des données entre applications, il est donc important de connaître les caractéristiques de la méthode de tri, qui ne peuvent se limiter simplement à une table de caractères, solution couramment utilisée pour le traitement des données mais visiblement insuffisante. Il me semble aussi nécessaire d'instituer un registre des méthodes de tri et d'adopter un mécanisme d'identification.

Cette méthode, fondée essentiellement sur la nouvelle structure de données que je viens de vous présenter, constitue maintenant une exigence architecturale globale préconisée par SHARE Europe, à laquelle SHARE Inc. a donné son accord de principe ces dernières années.

PUBLICATIONS IMPORTANTES

En conclusion, je signale à l'intention des personnes intéressées un certain nombre de publications qui traitent en détail du sujet que je viens d'exposer un peu rapidement. Il s'agit de la Norme canadienne Z243.4.1 sur le tri alphabétique et de deux ouvrages publiés par le **National Language Technical Centre** d'IBM : le premier, qui est le volume 2 du National Language Design Guide d'IBM, présente une généralisation de l'algorithme à toutes les langues alphabétiques prises en charge par IBM, tandis que l'autre, qui s'intitule **Keys to Sort and Search for Culturally Expected Results**, est un livre qui explique en détail la méthode que j'ai exposée. Je passe sous silence divers travaux que j'ai publiés avant ces derniers parce qu'ils sont rédigés en français, mais il me fera plaisir de fournir une bibliographie d'ouvrages en français aux personnes qui m'en feront la Demande[1].

J'espère que vous prendrez le temps de consulter ces ouvrages et je souhaite de plus que nous aurons la chance de voir cette méthode de traitement se généraliser. Au gouvernement du Québec, avant même que soient mis sur le marché des programmes distribués à grande échelle, la méthode que je viens de vous présenter sera mise en application au cours des prochains mois dans un certain nombre de ministères qui agissent la plupart du temps dans un environnement francophone et qui desservent une clientèle francophone. Vous aurez sans aucun doute compris que les méthodes classiques de traitement des données ne respectent pas davantage la logique culturelle de l'anglais que celle du français ou de toute autre langue, mais que nous avons maintenant les moyens de résoudre les problèmes inhérents à ces méthodes. En fait, nous avons même la responsabilité de résoudre ces problèmes. Ce sera notre contribution à l'idéal de qualité de la société de l'avenir.

Dépôt légal - 3e trimestre 1990
Bibliothèque nationale du Québec
ISBN 2-550-21180-4

[1] Bibliographie plus complète dans la version française.