ISO/CEI JTC 1/SC 2

Date: 2007-05-11

ISO/CEI FDIS 14651:2007(F)

ISO/CEI JTC 1/SC 2/AGT-TRI

Secrétariat: JISC

Technologies de l'information — Classement international et comparaison de chaînes de caractères — Méthode de comparaison de chaînes de caractères et description du modèle commun et adaptable d'ordre de classement

Information technology — International string ordering and comparison — Method for comparing character strings and description of the common template tailorable ordering

Type du document: Norme internationale
Sous-type du document:
Stade du document: (50) Approbation
Langue du document: F

D:\Donnees\SC2\NORMICLA\Rev2006\ISO-CEI_14651_(F).doc STD Version 2.1c2

Sommaire

Page

Avant-propos

L'ISO (Organisation internationale de normalisation) et la CEI (Commission électrotechnique internationale) forment le système spécialisé de la normalisation mondiale. Les organismes nationaux membres de l'ISO ou de la CEI participent au développement de Normes internationales par l'intermédiaire des comités techniques créés par l'organisation concernée afin de s'occuper des domaines particuliers de l'activité technique. Les comités techniques de l'ISO et de la CEI collaborent dans des domaines d'intérêt commun. D'autres organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'ISO et la CEI participent également aux travaux. Dans le domaine des technologies de l'information, l'ISO et la CEI ont créé un comité technique mixte, l'ISO/CEI JTC 1.

Les Normes internationales sont rédigées conformément aux règles données dans les Directives ISO/CEI, Partie 2.

La tâche principale du comité technique mixte est d'élaborer les Normes internationales. Les projets de Normes internationales adoptés par le comité technique mixte sont soumis aux organismes nationaux pour vote. Leur publication comme Normes internationales requiert l'approbation de 75 % au moins des organismes nationaux votants.

L'attention est appelée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. L'ISO et la CEI ne sauraient être tenues pour responsables de ne pas avoir identifié de tels droits de propriété et averti de leur existence.

L'ISO/CEI 14651 a été élaborée par le comité technique mixte ISO/CEI JTC 1, *Technologies de l'information*, sous-comité SC 2, *Jeux de caractères codés*.

Cette deuxième édition annule et remplace la première édition (ISO/CEI 14651:2001), qui a fait l'objet d'une révision technique.

Introduction

La présente Norme internationale fournit une méthode universelle de mise en ordre des données textuelles. La norme fournit également une table-modèle commune qui, lorsque adaptée, peut satisfaire aux exigences de tri d'une langue donnée tout triant de manière raisonnable les autres écritures.

La table-modèle commune est conçue de telle sorte qu'une adaptation s'avère nécessaire pour chaque environnement local. C'est pourquoi la conformité à la présente Norme internationale requiert que les modifications à cette table commune, appelées « deltas », soient déclarées de manière à documenter les différences dans les résultats.

La présente Norme décrit une méthode pour classer l'information textuelle de manière indépendante du contexte.

Le rapport technique ISO/CEI TR 14652 contient des dispositions complémentaires pour le tri à celle de la présente Norme internationale ; on y trouvera aussi des renseignements complémentaires sur les mots-clés de tri définis dans la présente Norme internationale.

Technologies de l'information — Classement international et comparaison de chaînes de caractères — Méthode de comparaison de chaînes de caractères et description du modèle commun et adaptable d'ordre de classement

1 Domaine d'application

La présente Norme internationale définit :

- une méthode de référence pour la comparaison de deux chaînes de caractères ayant pour but de déterminer leur ordre de classement dans une liste triée. La méthode s'applique à des chaînes utilisant le répertoire complet de l'ISO/CEI 10646, des sous-répertoires tels que ceux des divers jeux normalisés ISO/CEI à 8 bits ou tout autre jeu de caractères, et permet de produire des résultats de tri valables (après adaptation) pour un ensemble de langues de chaque système d'écriture. Cette méthode de référence utilise des tables de tri dérivées soit de la table-modèle commune de classement définie dans la présente norme internationale, soit d'une de ses adaptations. La méthode procure un format de référence de la table-modèle commune. Ce format est décrit en notation BNF (forme de Backus-Naur). Son emploi est normatif dans la présente Norme internationale.

- une table-modèle commune de classement utilisée par la méthode de référence. Cette table décrit un ordre de base pour tous les caractères de l'ISO/CEI 10646:2003 jusqu'à son amendement 2, plus les caractères LETTRE DÉVANÂGARÎ GGA, LETTRE DÉVANÂGARÎ DJDJA, LETTRE DÉVANÂGARÎ DDDA et LETTRE DÉVANÂGARÎ BBA (respectivement, les caractères U097B, U097C, U097E et U097F).

Tout cela permet de spécifier un ordre complètement déterministe. Cette table constitue le point de départ permettant de préciser un ordre de classement adapté aux règles de classement locales, sans qu'il soit nécessaire de connaître tous les systèmes d'écriture repris dans le JUC.

NOTE 1 Cette table-modèle commune de classement est destinée à être modifiée pour satisfaire les besoins d'environnements locaux. L'avantage principal de cette pratique, sur le plan mondial, réside dans le fait que pour d'autres systèmes d'écriture que celui de l'utilisateur, aucune modification n'est nécessaire et que cet ordre demeurera aussi cohérent que possible et prévisible dans un contexte international.

NOTE 2 Le répertoire de caractères utilisé dans la présente Norme internationale est équivalent à celui du standard Unicode, version 5.0.

- un nom de référence représentant cette version particulière de la table-modèle commune, à utiliser comme point de départ à toute adaptation. Ce nom implique notamment que la table est liée à un stade de développement particulier du jeu universel de caractères codés sur plusieurs octets (ISO/CEI 10646).

- des exigences pour la déclaration de différences (delta) entre une table de tri et la table-modèle commune.

La présente Norme internationale *ne* prescrit *pas* :

- de méthode particulière de comparaison ; toute méthode équivalente conduisant aux mêmes résultats est acceptable ;

- de format précis pour décrire ou pour adapter les tables dans une mise en œuvre donnée ;
- de symboles précis à utiliser par les mise en œuvre, sauf pour ce qui est du nom de la table-modèle commune de classement ;
- d'interface utilisateur particulière destinée à choisir les options ;
- de format interne particulier pour les clés intermédiaires utilisées dans les comparaisons ou pour la table de tri. L'utilisation de clés numériques n'est pas prescrite non plus ;
- d'ordre dépendant du contexte ;
- de prétraitement particulier des chaînes de caractères avant comparaison.

NOTE 1 Bien que ceci ne soit pas prescrit par la présente Norme internationale, il s'avère souvent nécessaire de préparer les chaînes de caractères avant leur comparaison (cf. l'annexe informative C).

NOTE 2 Bien que l'on ne prescrive aucune interface utilisateur destinée à choisir les options ou à adapter la table-modèle commune, la clause de conformité exige de toujours déclarer un delta, c'est à dire l'ensemble des différences par rapport à cette table. Il est fortement recommandé que l'application présente à l'utilisateur les options et adaptations disponibles.

2 Conformité

Un processus est conforme à la présente Norme internationale s'il produit des résultats identiques à ceux qui résultent de l'application des spécifications décrites aux articles 6.2 à 6.5.

Toute déclaration de conformité à la présente Norme internationale doit être accompagnée, directement ou par référence, d'une déclaration de ce qui suit :

- le nombre de niveaux de tri que le processus peut utiliser ; ce nombre doit être égal ou supérieur à trois ;
- si le paramètre de traitement `forward,position` est permis ;
- si le paramètre de traitement `backward` est permis et à quel niveau ;
- le *delta* d'adaptation décrit à l'article 6.4 et le nombre de niveaux définis dans ce delta ;
- si un processus de prétraitement est utilisé, la méthode utilisée doit être déclarée.

Il incombe au producteur de montrer en quoi sa déclaration de delta est reliée à la syntaxe de la table décrite à l'article 6.3 et comment la méthode de comparaison utilisée, si elle est différente de celle mentionnée à l'article 6, peut être considérée comme produisant les mêmes résultats que ceux prescrits par la méthode décrite à l'article 6. L'usage d'un processus de prétraitement est optionnel et ses détails ne sont pas précisés dans la présente norme internationale.

3 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

- ISO/CEI 10646:2003, *Technologies de l'information — Jeu universel de caractères codés sur plusieurs octets (JUC)*
- ISO/CEI 10646:2003/Amd 1:2005, *Technologies de l'information — Jeu universel de caractères codés sur plusieurs octets (JUC) — Amendement 1 – Glagolitic, Copte, Géorgien et autres caractères*

- ISO/CEI 10646:2003/Amd 2:2006, *Technologies de l'information — Jeu universel de caractères codés sur plusieurs octets (JUC) — Amendement 2 – N'Ko, phénicien et autres caractères*

4 Termes et définitions

Dans le cadre de la présente Norme internationale, les définitions suivantes s'appliquent :

4.1
chaîne de caractères
suite de caractères considérée comme un objet simple

4.2
classement
équivalent de « tri »

4.3
clé de tri
série de sous-clés utilisée pour déterminer un ordre

4.4
delta
liste des différences que présente une table de classement donnée par rapport à une autre. Une table de tri donnée associée à un delta donné forme une nouvelle table de tri. Sauf mention contraire, le terme « delta » désignera les différences par rapport à la table-modèle commune définie dans la présente Norme internationale

4.5
élément de poids
liste d'un certain nombre de poids séquentiellement ordonnés par niveau

4.6
élément de tri
suite constituée d'un ou de plusieurs caractères considérés comme une seule entité aux fins du tri

4.7
méthode de comparaison de référence
méthode de détermination de l'ordre relatif de deux clés (cf. l'article 6)

4.8
niveau (de tri)
numéro d'une sous-clé dans la série de sous-clés formant une clé

4.9
poids (de tri)
entier positif, utilisé dans les sous-clés pour indiquer l'ordre relatif des éléments de tri

4.10
prétraitement
procédé par lequel des chaînes de caractères sont transformées en d'autres chaînes avant le calcul de la clé de tri de chaque chaîne

4.11
sous-clé
suite de poids calculée pour une chaîne de caractères.

4.12
symbole
élément de tri

4.13
symbole de tri
symbole utilisé pour préciser les poids attribués à un élément de tri

4.14
table (de poids) de tri
table reliant les éléments de tri aux éléments de poids

4.15
tri
procédé par lequel on détermine si de deux chaînes la première est plus petite, égale ou plus grande que la seconde

5 Symboles et abréviations

Selon l'ISO/CEI 10646, les caractères se représentent à l'aide de U*X*, où *X* correspond à une série d'un à huit chiffres hexadécimaux (où toutes les lettres de la série de chiffres hexadécimaux sont en majuscules) et où *X* est le numéro du caractère dans l'ISO/CEI 10646. Cette convention est reprise dans la présente Norme internationale.

Dans la table-modèle commune, des symboles arbitraires représentent des poids selon la notation BNF décrite à l'article 6.3.1.

6 Comparaison de chaînes

6.1 Prétraitement des chaînes de caractères avant comparaison

Il peut s'avérer nécessaire de transformer les chaînes de caractères avant de leur appliquer la méthode de comparaison de référence (l'annexe C fournit un exemple d'une telle préparation). Bien que n'étant pas l'objet de la présente Norme internationale, le prétraitement peut être une partie importante du processus de tri. On consultera l'annexe C pour des exemples de prétraitement.

S'il y a lieu, une partie importante de la phase préparatoire consiste à transformer les caractères d'un codage non-JUC à des caractères du JUC fournis en entrée à la méthode de comparaison. Cette tâche peut comprendre notamment le traitement correct de séquences d'échappement dans le codage original, la transformation de caractères sans attribution dans le JUC à des positions de code dans la zone privée et la transposition de caractères dans le cas de chaînes qui ne seraient pas stockées en ordre logique. Par exemple, dans le cas de codages arabes en ordre visuel, les caractères doivent être mis en ordre logique ; dans le cas de certains codages à usage bibliographique, les accents combinatoires stockés avant leur caractère de base doivent être déplacés après le caractère de base. La suite résultante peut devoir être re-transformée dans le codage original.

NOTE 1 La table-modèle commune est conçue de telle sorte que les séquences combinatoires et les caractères simples (précomposés) correspondants aient exactement le même ordre. Pour éviter de violer par mégarde cet invariant (et au passage la conformité à Unicode), l'adaptation devrait changer le classement des séquences combinatoires quand le classement des caractères précomposés correspondant est changé. Par exemple, si Ä est déplacé après Z, alors le classement de la séquence combinatoire <A>+<tréma combinatoire> devrait aussi être changé. Pour éviter de révéler des différences de codage invisible à l'utilisateur, on recommande de normaliser les chaînes selon la forme FND de l'algorithme de normalisation Unicode – voir le Unicode Technical Report nº 15 dans la bibliographie.

NOTE 2 Les séquences d'échappement et les caractères de commande sont très délicats à interpréter ; il est fortement recommandé de les filtrer ou de les transformer.

NOTE 3 Puisque la méthode de comparaison de référence est une description logique du procédé de comparaison de chaînes, rien n'empêche une mise en œuvre de cette méthode d'utiliser exclusivement un codage autre qu'un codage du JUC, pour autant que les résultats obtenus soient les mêmes que si la méthode de référence était utilisée.

6.2 Construction des clés et comparaison

6.2.1 Préliminaires

6.2.1.1 Hypothèses

La table de tri est une transformation des éléments de tri en éléments de poids. Pour chaque élément de poids, la table-modèle commune décrit quatre niveaux. L'adaptation peut augmenter ou réduire ce nombre de niveaux, mais pas à moins de trois.

NOTE Dans la table-modèle commune, les niveaux ont généralement les significations suivantes, bien que cet usage ne soit pas absolu :

Niveau 1 : ce niveau correspond généralement au jeu de lettres de base pour une écriture alphabétique, au jeu de caractères courants pour une écriture idéographique ou syllabique.

Niveau 2 : ce niveau correspond généralement aux diacritiques pouvant accompagner les caractères de base de chaque écriture. En certaines langues, les lettres accentuées sont considérées comme des lettres de base de l'alphabet et ne sont pas affectées par ce niveau, mais seulement par le premier niveau. En espagnol par exemple, le N TILDE est considéré comme une lettre de base de l'alphabet latin ; par conséquent, une adaptation pour l'espagnol changera la définition de N TILDE de « le poids d'un N au premier niveau et le poids d'un TILDE au second niveau » à « le poids d'un N TILDE (entre N et O) au premier niveau et une indication de l'absence de diacritique au second niveau ». Pour certains caractères, on prend également en compte des variantes de forme au second niveau, par exemple ß (la LETTRE MINUSCULE LATINE S DUR), qui est traitée comme un équivalent de ss au premier niveau mais s'en distingue traditionnellement au second niveau.

Niveau 3 : ce niveau est généralement associé aux distinctions de casse (majuscules-minuscules) ou aux variantes de formes (comme la distinction entre hiragana et katakana).

Niveau 4 : ce niveau est généralement consacré aux distinctions pondérales plus fines que celles des autres niveaux. Le dernier niveau (le quatrième dans la table-modèle commune) est souvent utilisé pour donner des poids additionnels à des caractères « spéciaux », c'est à dire des caractères qui ne sont pas normalement utilisés dans l'orthographe des mots d'une langue (ponctuation, vignettes, etc.), souvent appelés « ignorables » dans le contexte du tri informatique.

6.2.1.2 Propriétés de traitement

Une table de tri adaptée donnée possède des propriétés spécifiques de balayage et de classement. Ces propriétés peuvent avoir été changées par l'adaptation.

Une direction de balayage (vers l'avant ou vers l'arrière) pour chaque niveau est utilisée pour indiquer comment traiter la chaîne. La direction de balayage est une propriété globale de chaque niveau défini dans la table adaptée.

Si le dernier niveau est supérieur à trois, il existe une propriété optionnelle de ce niveau appelée l'option « position » : lorsque active, une comparaison des positions numériques de chaque caractère « ignorable » dans les deux chaînes est effectuée, avant de comparer leurs poids. En d'autres mots, si deux chaînes sont équivalentes à tous les niveaux sauf le dernier, la chaîne contenant un caractère ignorable en position la plus basse est classée avant l'autre. Si les caractères ignorables ont les mêmes positions, alors leurs poids sont considérés jusqu'à ce qu'une différence soit trouvée. Le traitement correct de cette propriété optionnelle n'est pas nécessaire à la conformité à la présente Norme internationale.

NOTE La direction de balayage (vers l'avant ou vers l'arrière) n'est normalement pas reliée à la direction naturelle d'écriture. La direction de balayage s'applique à la suite logique de la chaîne de caractères codés.

Dans le cas d'écritures de droite à gauche comme l'arabe, l'ISO/CEI 10646 prescrit que les premiers caractères en ordre logique sont ceux apparaissant à droite en ordre de présentation. En écriture latine au contraire, les premiers caractères en ordre logique apparaissent à gauche en ordre de présentation.

Le balayage vers l'avant commence au début de la séquence en ordre logique, alors que le balayage vers l'arrière commence à la fin, sans égard à la direction de présentation. La direction de balayage pour fins de tri est une propriété globale de chaque niveau décrit dans la table.

Dans l'ISO/CEI 10646, l'écriture arabe est artificiellement séparée en deux pseudo-écritures : 1) l'écriture arabe logique, intrinsèque, codée indépendamment des formes contextuelles et 2) les formes de présentations arabes. Les deux permettent le codage complet de l'arabe, mais le codage intrinsèque est normalement privilégié pour sa meilleure capacité de traitement, alors que certaines applications de présentation préfèrent les formes de présentation. L'ISO/CEI 10646 ne prescrit pas l'ordre de stockage des formes de présentation ; dans certaines réalisations, elles sont stockées en ordre inverse de celui utilisé pour le codage intrinsèque. Par conséquent, la phase de préparation devrait assurer que les formes de présentation arabes et les autres caractères arabes soient fournis en ordre logique à la méthode de comparaison.

Une table de tri adaptée peut être séparée en sections pour faciliter l'adaptation. On donne alors à chaque section un nom, conformément aux dispositions de l'article 6.3.1. Une des possibilités d'adaptation est de donner un certain ordre à chaque section et de changer l'ordre relatif d'une section par rapport à d'autres.

6.2.2 Méthode de référence de construction des clés

Lorsque deux chaînes doivent être comparées pour déterminer leur ordre relatif, elles sont d'abord analysées en séquences d'éléments de tri, en tenant compte des déclarations « `collating-element` » à caractères multiples présents dans la table de tri (*si* la syntaxe de l'article 6.3.1 est utilisée). Dans la syntaxe utilisée pour exprimer la table-modèle commune, le nom d'un élément de tri associé à un seul caractère est formé de la lettre « U » suivie du numéro du caractère dans le JUC, en notation hexadécimale. Les noms et caractères associés aux éléments de tri multi-caractères sont définis par les déclarations d'éléments de tri.

Une suite de *m* sous-clés intermédiaires est alors formée de chaque chaîne, *m* étant le nombre de niveaux décrits dans une table de poids de tri adaptée.

Chaque clé de tri est une suite de sous-clés. Chaque sous-clé est une liste de poids numériques. Une sous-clé est construite en ajoutant successivement la liste des poids attribués à chaque élément de tri de la chaîne au niveau de la sous-clé en construction. Dans la table-modèle commune, le mot-clé « `IGNORE` » trouvé en place d'une suite de poids à un niveau indique que la suite de poids à ce niveau pour cet élément de tri est vide.

Il y a trois façons de former des sous-clés : vers l'avant (paramètre de traitement « `forward` »), vers l'arrière (paramètre de traitement « `backward` ») et de façon positionnelle (paramètre de traitement « `forward,position` »). Les sous-clés formées de façon positionnelle ne peuvent apparaître qu'au dernier niveau et seulement si ce niveau est supérieur à trois. La conformité n'exige pas la formation de sous-clés de façon positionnelle ; une réalisation incapable de formation positionnelle interprétera « `forward,position` » comme s'il s'agissait de « `forward` ».

Si la table adaptée ne contient pas d'entrée pour un caractère de la chaîne d'entrée, les poids de ce caractère ne sont pas définis. Les caractères de poids indéfinis devraient être triés, par rapport aux caractères ayant des poids définis, comme s'ils avaient le poids nommé « UNDEFINED » au premier niveau. Si le symbole « UNDEFINED » n'a pas de poids attribué avant l'attribution d'un poids à symbole <SFFFF> dans la table adaptée, on considèrera que le poids de « UNDEFINED » est tout juste inférieur à celui de <SFFFF>. Le classement relatif des caractères de poids indéfinis entre eux n'est pas précisé par la présente Norme internationale.

NOTE 1 Une manière possible de classer les caractères de poids indéfinis entre eux est de supposer l'existence de lignes d'adaptation comme celles qui suivent, rangées en ordre de numéro de caractères dans le JUC (<PLAIN> représente ici le poids maximal de niveau 4) :

<UXXXX> "<UNDEFINED><UXXXX>";<BASE>;<MIN>;<PLAIN>

NOTE 2 <SFFFF> est le plus grand poids de premier niveau dans table-modèle commune.

6.2.2.1 Formation de sous-clé vers l'avant (paramètre de traitement « `forward` »)

En présence du paramètre de traitement « `forward` » à un niveau donné, on construit la sous-clé de la façon suivante :

On balaie vers l'avant un à un les éléments de tri de la chaîne de caractères d'entrée pour leur attribuer un poids. On obtient les poids en recherchant les éléments de tri dans la table de poids de tri adaptée donnée et en extrayant la liste de poids pour le niveau considéré. Cette liste de poids s'ajoute à la fin de la sous-clé.

6.2.2.2 Formation de sous-clé vers l'arrière (paramètre de traitement « `backward` »)

En présence du paramètre de traitement « `backward` » à un niveau donné, on construit la sous-clé vers l'avant et on la renverse, poids par poids.

6.2.2.3 Formation de sous-clé positionnelle (paramètre de traitement « `forward,position` »)

En présence du paramètre de traitement « `forward,position` » au dernier niveau, on construit la sous-clé de la même façon que vers l'avant, si ce n'est que les poids des éléments de tri qui sont pris en compte à tous les niveaux sauf le dernier sont remplacés par un seul poids (appelé <PLAIN> ici) supérieur à tous les poids du dernier niveau dans la table adaptée donnée. Les éléments de tri qui sont ignorés à tous les niveaux sauf le dernier conservent leurs poids tels que donnés dans la table adaptée donnée. Enfin, toute séquence de queue de la valeur maximale (<PLAIN>) est retirée de la sous-clé (ce qui en pratique remplace chaque <PLAIN> par un poids nul).

NOTE Il est permis, à chaque niveau, d'appliquer une réduction de toutes les sous-clés de ce niveau, en autant que cette réduction préserve l'ordre. Une telle réduction est utile pour les niveaux 2, 3 et 4. Les sous-clés de niveau 2 contiennent souvent de longues suites du poids appelé <BASE> dans la table donnée à l'annexe A. Les sous-clés de niveau 3 contiennent souvent de longues suites du poids appelé <MIN> à l'annexe A. Les sous-clés de niveau 4 contiennent souvent de longues suites du poids appelé <PLAIN> ici. Une telle technique de réduction préservant l'ordre consiste à coder, dans la sous-clé de dernier niveau, la position relative de chaque caractère autrement ignoré ; c'est là l'origine du nom de l'option « *`position`* ».

6.2.3 Méthode de comparaison de référence pour le tri des chaînes de caractères

La méthode de comparaison de référence pour le classement de deux chaînes de caractères (*après* le prétraitement, qui ne fait pas partie de cette méthode de comparaison) consiste à comparer les clés de tri construites selon la méthode de référence décrite à l'article 6.2.2 de la présente Norme internationale :

- En utilisant une table de poids de tri adaptée donnée, construire une clé de tri pour chacune des chaînes à comparer.
- Comparer ensuite les clés selon la définition de l'ordre des clés donnée ci-dessous dans cet article. Les clés peuvent être comparées jusqu'à un niveau donné ou jusqu'au dernier niveau de la table de poids de tri adaptée donnée.

NOTE 1 La comparaison peut être effectuée *pendant* la construction des clés, en arrêtant cette construction dès que l'ordre des chaînes peut être déterminé. Cette technique est parfois appelée *évaluation paresseuse* et certains systèmes l'utilisent implicitement. Elle permet d'éviter la construction complète des clés quand une différence peut être trouvée tôt pendant la construction. Quand un ensemble important de chaînes doit être trié, il est recommandé de construire et de stocker les clés – ou tout au moins un segment initial – avant de les comparer.

Les poids associés à des niveaux différents ne doivent pas être comparés, on ne doit donc pas comparer les sous-clés de différents niveaux. Les clés construites à partir de tables adaptées différentes ne doivent pas être comparées.

NOTE 2 Ceci permet aux mises en œuvre d'attribuer les poids à chaque niveau indépendamment des autres niveaux et sans égard à d'autres tables adaptées.

m est le plus grand niveau d'une table adaptée donnée. Rappelons qu'une clé est une liste, de longueur m, de sous-clés ; une sous-clé est une liste de poids ; un poids est un entier positif. D'autres notations utilisées ci-dessous sont :

- L_z est la longueur de la sous-clé z, c'est-à-dire le nombre de poids dans cette sous-clé.
- $z_{\mathbf{pd}(a)}$, où $1 \le a \le L_z$, est le poids à la position a (un entier > 0) de la sous-clé z.
- $u_{\mathbf{sc}(b)}$, où $1 \le b \le m$, est la sous-clé de niveau b (un entier > 0) de la clé u.

Les ordres des poids, des sous-clés et des clés de tri (jusqu'à un certain niveau ou jusqu'au dernier niveau) sont des relations d'ordre total, définies pour une table de tri adaptée donnée comme suit :

1. Les poids sont des valeurs entières positives (dans la méthode de référence) et sont comparés comme tels aux fins du classement.

2. Une sous-clé v est *plus petite* qu'une sous-clé w (on notera $v < w$) **si et seulement s'il** existe un entier i, où $1 \le i \le L_v+1$ et $i \le L_w$, tel que

   - $i = 1$ et $v_{\mathbf{pd}(i)} < w_{\mathbf{pd}(i)}$, ou
   - pour tous les entiers j, $1 \le j < i$, l'égalité $v_{\mathbf{pd}(j)} = w_{\mathbf{pd}(j)}$ est maintenue, et soit
   - $i \le L_v$ et $v_{\mathbf{pd}(i)} < w_{\mathbf{pd}(i)}$, soit
   - $i = L_v+1$ et $0 < w_{\mathbf{pd}(i)}$.

   Une sous-clé v est *plus grande* qu'une sous-clé w (on notera $v > w$) **si et seulement si** w est plus petite que v. Une sous-clé v est *égale* à une sous-clé w (on notera $v = w$) **si et seulement si** v n'est pas plus petite que w et w n'est pas plus petite que v.

3. Une clé de tri x est *plus petite* qu'une clé de tri y au niveau s (on notera $x <_s y$) **si et seulement s'il** existe un entier i, où $1 \le i \le s$ et $i \le m$, tel que

   - $i = 1$ et $x_{\mathbf{sc}(i)} < y_{\mathbf{sc}(i)}$, ou
   - pour tous les entiers j, $1 \le j < i$, l'égalité $x_{\mathbf{sc}(j)} = y_{\mathbf{sc}(j)}$ est maintenue, et $x_{\mathbf{sc}(i)} < y_{\mathbf{sc}(i)}$.

   Une clé de tri x est *plus grande* qu'une clé de tri y au niveau s (on notera $x >_s y$) **si et seulement si** y est plus petite que x au niveau s. Une clé de tri x est *égale* à une clé de tri y au niveau s (on notera $x =_s y$) **si et seulement si** x n'est pas plus petite que y au niveau s et y n'est pas plus petite que x au niveau s.

4. Pour les clés de tri, <, > et = sont définis comme $<_m$, $>_m$ et $=_m$ respectivement.

NOTE 3 Pour les clés de tri, $x <_t y$ implique que $x <_{t+1} y$, $x >_t y$ implique que $x >_{t+1} y$, $x =_t y$ implique que $x =_{t-1} y$, $x <_0 y$ est faux, $x >_0 y$ est faux et $x =_0 y$ est vrai. Au-delà du niveau m, pour une table adaptée donnée, il n'y a plus de distinctions d'ordre. On notera que cette définition implique que si une clé est « plus petite » qu'une autre au niveau 1, elle est aussi « plus petite » aux niveaux 2, 3, 4, etc. En général, lorsqu'une clé est plus petite qu'une autre à un certain niveau, elle l'est aussi à tous les niveaux subséquents. *A contrario*, lorsque deux clés sont égales à un certain niveau, elles sont aussi égales à tous les niveaux inférieurs.

6.3 Table-modèle commune : composition et interprétation

Cet article précise :

- La syntaxe utilisée par la table-modèle commune donnée à l'annexe A ou par une table adaptée basée sur la table-modèle commune telle que donnée à l'annexe A.
- Les contraintes de forme d'une table utilisant cette syntaxe.

- L'interprétation à donner aux énoncés d'adaptation dans les deltas pour des tables utilisant cette syntaxe.
- L'évaluation des symboles en poids dans les tables adaptées utilisant cette syntaxe.
- Les conditions d'équivalence de deux tables.
- Les conditions d'équivalence des résultats de comparaison.

6.3.1 Règles de syntaxe BNF pour la table-modèle commune de l'annexe A

Les définitions <entre crochets> utilisent des termes qui ne sont pas définis dans cette syntaxe BNF et suivent l'usage général en français.

Autres conventions :
* indique une répétition (0 fois ou plus) d'un atome ou d'un groupe d'atomes ;
+ indique une répétition (1 fois ou plus) d'un atome ou d'un groupe d'atomes ;
? indique l'apparition optionnelle (0 ou 1 fois) d'un atome ou d'un groupe d'atomes ;
des parenthèses servent à grouper des atomes ;
les productions se terminent par un point-virgule.

Définition des tables de tri comme des suites de lignes :

```
table_de_poids = table-modèle_commune | table_adaptée ;

table-modèle_commune =
                 ligne_simple+ ;

table_adaptée =  ligne_de_table+ ;
```

Définition des types de ligne :

```
ligne_simple =   (définition_de_symbole | élément_de_tri |
                  attribution_de_poids | fin_ordre)?   achèvement_de_ligne
                 ;

ligne_de_table = ligne_simple | ligne_adaptation ;

ligne_adaptation =
                 (reclasser_après | début_ordre | fin_de_reclassement |
                  définition_de_section | reclasser_section_après)
                 achèvement_de_ligne ;
```

Définition de la syntaxe de base pour les poids de tri :

```
définition_de_symbole =
                 'collating-symbol' espace+ élément_symbole ;

élément_symbole = symbole | symbole_intervalle ;

symbole_intervalle = symbole '..' symbole ;

symbole =        symbole_simple | symbole_juc;

symbole_juc =    ('<U'  chaîne_de_un_à_huit_hexa '>') |
                 ('<U-' chaîne_de_un_à_huit_hexa '>') ;

symbole_simple = '<' identifiant '>' ;

élément_de_tri =
                 'collating-element' espace+ symbole espace+
                 'from' espace+ suite_de_symboles_cités ;
```

```
suite_de_symboles_cités =
                ’"’ poids_simple+ ’"’ ;

attribution_de_poids =
                poids_simple | poids_symbolique ;

poids_simple =  élément_symbole | ’UNDEFINED’ ;

poids_symbolique =
                élément_symbole espace+ liste_de_poids ;

liste_de_poids = atome_de_niveau (point-virgule atome_de_niveau)* ;

atome_de_niveau =
                groupe_de_symboles | ’IGNORE’ ;

groupe_de_symboles =
                élément_symbole | suite_de_symboles_citée ;

fin_ordre =     ’order_end’ ;
```

Définition de la syntaxe d’adaptation :

```
reclasser_après =
                ’reorder-after’ espace+ symbole_cible ;

symbole_cible =  symbole ;

début_ordre =   ’order_start’ espace+ direction_multi_niveaux ;

direction_multi_niveaux =
                (direction point-virgule)* direction (’,position’)? ;

direction =     ’forward’ | ’backward’ ;

fin_de_reclassement =
                ’reorder-end’ ;

définition_de_section =
                définition_de_section_simple |
                définition_de_section_liste ;

définition_de_section_simple =
                ’section’ espace+ identifiant_de_section ;

identifiant_de_section =
                identifiant ;

définition_de_section_liste =
                ’section’ espace+ identifiant_de_section espace+
                liste_de_symboles ;

liste_de_symboles =
                élément_symbole (point-virgule élément_symbole)* ;

reclasser_section_après =
                ’reorder-section-after’ espace+ identifiant_de_section
                espace+ symbole_cible ;
```

Définition des atomes de base utilisés par le reste de la syntaxe :

```
identifiant =   (lettre | chiffre) partie_id* ;

partie_id =     lettre | chiffre | ’-’ | ’_’ ;
```

```
achèvement_de_ligne =
                espace* commentaire? FDL ;

commentaire =   début_commentaire caractère* ;

chaîne_de_un_à_huit_hexa =
                hex_maj | hex_maj hex_maj |
                hex_maj hex_maj hex_maj |
                hex_maj hex_maj hex_maj hex_maj |
                hex_maj hex_maj hex_maj hex_maj hex_maj |
                hex_maj hex_maj hex_maj
                hex_maj hex_maj hex_maj |
                hex_maj hex_maj hex_maj hex_maj
                hex_maj hex_maj hex_maj |
                hex_maj hex_maj hex_maj hex_maj
                hex_maj hex_maj hex_maj hex_maj ;

chaîne_numérique_hexa =
                hex_maj+ ;

espace =        ' ' | <TAB> ;

point-virgule = ';' ;

début_commentaire =
                '%' ;

chiffre =       '0' | '1' | '2' | '3' | '4' |
                '5' | '6' | '7' | '8' | '9' ;

hexa_maj =      chiffre | 'A' | 'B' | 'C' | 'D' | 'E' | 'F' ;

lettre =        'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' |
                'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' |
                'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' |
                'v' | 'w' | 'x' | 'y' | 'z' |
                'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' |
                'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N' |
                'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' |
                'V' | 'W' | 'X' | 'Y' | 'Z' ;

FDL =           <fin de ligne selon la convention en cours> ;

caractère =     <tout caractère membre du répertoire du jeu de caractères
                codés utilisé, sauf les caractères indiquant les fins de
                ligne> ;
```

6.3.1.1 Mots-clés

Les mots-clés de la syntaxe sont utilisés comme suit :

collating-symbol définit un symbole de tri représentant un poids.

collating-element définit un symbole d'élément de tri représentant un élément de tri à plusieurs caractères.

order_start définit des règles de tri. Après reclassement, cet énoncé est suivi d'un ou plusieurs énoncés d'ordre de tri, qui attribuent des poids de tri multi-niveaux à des éléments de tri. Un élément de tri est soit un caractère, soit une sous-chaîne définie.

order_end indique la fin des énoncés d'ordre de tri.

reorder-after redéfinit des règles de tri, en précisant après quel poids de symbole de tri les lignes se trouvant entre ce « reorder-after » et le suivant (ou un « reorder-end ») doivent être déplacées. Cet énoncé est suivi de lignes de table. Le résultat est de réattribuer des valeurs de symboles de tri ou des poids d'éléments de tri.

reorder-end indique la fin des énoncés d'ordre de tri après un « reorder-after ».

section définit une section de la table. Une section peut être déplacée en entier par « reorder-section-after ».

reorder-section-after redéfinit l'ordre des sections. Ce mot-clé est suivi d'un identifiant de section et d'une section cible. Le résultat est de réattribuer des valeurs de symboles de tri ou des poids d'éléments de tri.

6.3.2 Contraintes de forme

CF 1. Tout *symbole_simple* apparaissant dans une *liste_de_poids* doit aussi paraître dans l'*élément_symbole* initial d'un *poids_symbolique*, ou d'une *définition_de_symbole* n'apparaissant pas après dans la suite de lignes formant une *table_de_poids*.

NOTE Tous les *symbole_simple* doivent être définis avant d'être utilisés.

CF 2. Aucun *symbole* apparaissant dans une *définition_de_symbole* dans une *table_de_poids* dépourvu de *ligne_adaptation* ne peut apparaître dans une autre *définition_de_symbole* dans la même *table_de_poids*.

NOTE Les symboles de tri en double sont interdits. Cette contrainte est respectée dans la table-modèle commune. Elle doit être respectée dans une *table_adaptée* après que tous les reclassements de lignes ont été appliqués.

CF 3. Toutes les *liste_de_poids* dans une *table_adaptée* doivent contenir le même nombre d'*atome_de_niveau*. Un *atome_de_niveau* vide doit être interprété comme l'*élément_de_tri* lui-même.

NOTE Une table adaptée doit posséder un nombre de niveaux cohérent.

CF 4. Une *table_adaptée* doit contenir un énoncé *début_ordre*. Après que tous les reclassements de lignes ont été effectués, cet énoncé doit suivre toutes les *définition_de_symbole* et précéder tous les *poids_symbolique*.

CF 5. Une *direction_multi_niveaux* dans une *table_adaptée* doit contenir le même nombre de *direction* que le nombre d'*atome_de_niveau* dans les *liste_de_poids* de cette *table_adaptée*.

NOTE Tout énoncé *début_ordre* doit avoir le nombre de niveaux généralement utilisé dans la table.

CF 6. Si un *atome_de_niveau* dans une *liste_de_poids* est formé d'un *groupe_de_symboles*, tous les *atome_de_niveau* de cette *liste_de_poids* doivent aussi être formés de *groupe_de_symboles*.

NOTE *IGNORE* ne doit pas être utilisé à un niveau après un symbole explicite pour un poids.

CF 7. Tout *identifiant_de_section* paraissant dans un *reclasser_section_après* doit paraître dans une *définition_de_section* qui se présente auparavant dans la suite de *ligne_de_table* qui forment une *table_adaptée*.

NOTE Les *identifiant_de_section* doivent être définis avant d'être utilisés.

CF 8. Deux *définition_de_section* dans une *table_adaptée* ne doivent pas contenir les mêmes valeurs d'*identifiant_de_section*.

NOTE Les définitions de section multiples sont interdites ; les *identifiant_de_section* doivent être uniques.

CF 9. Chaque *reclasser_après* dans une *table_adaptée* doit éventuellement être suivi soit d'un *fin_de_reclassement*, soit d'un autre *reclasser_après*.

CF 10. Une *table_adaptée* doit contenir un *début_ordre* et un *fin_ordre*.

CF 11. Si l'*identifiant_de_section* d'un énoncé *reclasser_section_après* correspond à l'*identifiant_de_section* d'une *définition_de_section_liste*, alors le *symbole_cible* de cet énoncé *reclasser_section_après* ne doit avoir la valeur d'aucun des *symbole* de la *définition_de_section_liste*.

NOTE Une *section* ne peut être reclassée après une ligne contenue dans cette même *section* ; le reclassement récursif de lignes est interdit.

CF 12. Tout *symbole_intervalle* doit contenir deux *symbole* remplissant les conditions suivantes : 1) Les deux *symbole* doivent avoir un préfixe commun. 2) La partie de l'*identifiant* de chaque *symbole* suivant le préfixe commun doit être une *chaîne_numérique_hexa* contenant le même nombre de chiffres hexadécimaux. 3) Les *chaîne_numérique_hexa* étant interprétées comme des valeurs numériques, celle du premier *symbole* doit être inférieure à celle du second *symbole*. Le nombre de valeurs dans le *symbole_intervalle* est égal à un plus la différence entre les valeurs numériques des *chaîne_numérique_hexa* du second *symbole* et du premier *symbole*.

NOTE Un *symbole_intervalle* bien formé est de la forme <S4E00>..<S9FA5>, où le préfixe commun est « S » et le reste de la partie *identifiant* de chaque *symbole* est une *chaîne_numérique_hexa*.

CF 13. Tout *poids_symbolique* contenant plus d'un *symbole_intervalle* doit contenir seulement des *symbole_intervalle* respectant la condition suivante : chaque *symbole_intervalle* après le premier doit avoir le même nombre de valeurs que le premier *symbole_intervalle*.

NOTE Cette condition garantit que tous les intervalles seront bien formés, puisque pour tout *poids_symbolique*, tous les intervalles auront le même nombre de valeurs.

6.3.3 Interprétation des tables adaptées

I 1. Une *section* est soit 1) la liste des *ligne_simple* contenant une *définition_de_symbole* dont la valeur est égale à tout *symbole* contenu dans la *liste_de_symboles* d'une *définition_de_section_liste*, soit 2) la liste de *ligne_simple* suivant une *définition_de_section_simple* dans une *table_adaptée*.

NOTE Une *section* est définie 1) par une *liste_de_symboles* spécifique ou 2) par l'ensemble des lignes suivant une *définition_de_section* jusqu'à une ligne d'adaptation telle que *début_ordre* ou *reclasser_section_après*, une autre *définition_de_section* ou la fin de la table.

I 2. Une *ligne_simple* constituée d'une *définition_de_symbole* contenant un *symbole_intervalle* est équivalente à une suite de *ligne_simple*, chacune contenant un *symbole* à la place du *symbole_intervalle*. On crée un *symbole* pour chaque *ligne_simple* en concaténant une *chaîne_numérique_hexa* au préfixe commun du *symbole_intervalle,* en ordre numérique, en commençant avec la valeur associée à la *chaîne_numérique_hexa* du premier *symbole* de l'intervalle et en terminant avec la valeur associée à la *chaîne_numérique_hexa* du second *symbole*. La *chaîne_numérique_hexa* adjointe au préfixe commun doit contenir le même nombre de chiffres que la *chaîne_numérique_hexa* du premier *symbole*. Le nombre de *ligne_simple*s ainsi créé est égal au nombre de symboles dans le *symbole_intervalle*.

NOTE Une définition_de_symbole de la forme « collating-symbol <S0301>..<S0303> » est équivalente aux trois lignes :
collating-symbol <S0301>
collating-symbol <S0302>
collating-symbol <S0303>

I 3. Une ligne_simple constituée d'un poids_symbolique contenant un ou plusieurs symbole_intervalles est équivalente à une suite de ligne_simples, où chaque symbole_intervalle a été transformé en une suite de symboles de la manière décrite en I 2 pour une définition_de_symbole.

NOTE Un *poids_symbolique* de la forme «<U2000>..<U2002> <S0301>..<S0303>;<BASE>;<MIN>; <U2000>..<U2002>» est équivalent aux trois lignes:
<U2000> <S0301>;<BASE>;<MIN>;<U2000>
<U2001> <S0302>;<BASE >;<MIN>;<U2001>
<U2002> <S0303>;<BASE >;<MIN>;<U2002>

I 4a. Une *table_adaptée* contenant un *reclasser_après* est équivalente à une *table_adaptée* où :

1. Toute *ligne_de_table* située avant ce *reclasser_après* et contenant des *attribution_de_poids* dont le *symbole* initial correspond au *symbole* initial de toute *attribution_de_poids* dans les *ligne_de_table* se trouvant entre le *reclasser_après* et la *fin_de_reclassement* est enlevée ;

2. Les *ligne_de_table*s entre ce *reclasser_après* et la première *fin_de_reclassement* qui suit sont déplacées immédiatement après la première *ligne_de_table* dans la *table_adaptée* contenant une *attribution_de_poids* dont le *symbole* initial correspond au *symbole_cible* du *reclasser_après* ;

3. Ce *reclasser_après* et la *fin_de_reclassement* sont enlevés.

NOTE Déplacer le bloc de lignes entre le *reclasser_après* et la *fin_de_reclassement* à la suite du *symbole_cible*, enlever toute *définition_de_symbole* antérieure qui dédouble une *définition_de_symbole* dans les lignes reclassées et enlever les *reclasser_après* et *fin_de_reclassement* eux-mêmes.

I 4b. Quand une *table_adaptée* contient de multiples groupes de lignes à reclasser, la table est interprétée en traitant chaque *reclasser_après* en séquence, en partant de la première ligne de la table.

NOTE Un reclassement peut affecter des lignes déjà déplacées par un reclassement antérieur.

I 5. Une *table_adaptée* contenant un *reclasser_section_après* est équivalente à une *table_adaptée* obtenue en déplaçant la *section* associée à ce *reclasser_section_après* (sans changer l'ordre des *ligne_de_table* de la *section*) immédiatement après la dernière *ligne_de_table* de la *table_adaptée* contenant une *définition_de_symbole* dont le *symbole* correspond au *symbole_cible* du *reclasser_section_après* et en enlevant ce *reclasser_section_après*.

I 6. Une *table_de_poids* est dite en forme normale si elle ne contient ni *reclasser_après* ni *reclasser_section_après*.

NOTE Une *table_adaptée* peut être mise en forme normale au moyen des opérations décrites en I 4 et I 5.

6.3.4 Évaluation des tables de poids

E 1. Une *table_de_poids* en forme normale est dite évaluée quand toutes les *attribution_de_poids* dans la *table_de_poids* sont associées à des entiers positifs (des poids) tels que ces poids augmentent de façon monotone avec l'ordre d'apparition des *attribution_de_poids* dans la *table_de_poids*.

NOTE 1 Les *ligne_de_table* de la *table_de_poids* peuvent d'abord être associées à l'ensemble des entiers positifs en ordre séquentiel. Cette relation définit un ensemble ordonné de numéros de ligne. Les *attribution_de_poids* sont alors associés à un ensemble d'entiers positifs (poids) variant de façon monotone avec les numéros de ligne.

NOTE 2 Cette règle n'impose pas de nombre particulier pour le poids de la première *attribution_de_poids* (qui doit toutefois être positif) ni n'impose que les poids soient des nombres consécutifs.

E 2. Une *table_de_poids* évaluée est dite complètement évaluée quand chaque *symbole_simple* apparaissant dans une *liste_de_poids* dans la *table_de_poids* a été associé au poids correspondant à l'*attribution_de_poids* contenant ce *symbole_simple*.

NOTE 3 Chaque *liste_de_poids* peut être interprétée comme contenant soit des *symbole* associé à des poids entiers, soit la chaîne « IGNORE », laquelle indique une suite vide de poids. À ce point, l'injection mathématique des chaînes peut être définie en utilisant cette *table_de_poids*.

NOTE 4 Dans une table adaptée, la valeur de toute *chaîne_numérique_hexa* associée à un *symbole* ne reflète généralement pas le poids numérique de ce *symbole*.

6.3.5 Conditions d'équivalence de tables spécifiques

Deux *table_de_poids* TBL1 et TBL2 sont dites équivalentes pour un niveau particulier si toute comparaison de chaînes utilisant ces tables jusqu'à ce niveau donne les mêmes résultats.

NOTE Si deux tables étaient présumées équivalentes, on devrait obtenir les mêmes résultats en formant des clés pour deux chaînes avec la table TBL1 et en les comparant qu'en formant des clés avec TBL2 et en les comparant.

6.3.6 Conditions d'équivalence des résultats

Une mise en œuvre du tri international de chaînes est conforme à la présente Norme internationale si, pour tout ensemble de chaînes *S* défini sur un répertoire *R*, la mise en œuvre peut reproduire les mêmes comparaisons que celles résultant de la comparaison des nombres d'une injection construite selon les règles de l'article 6.2.3 de la présente Norme internationale.

6.4 Déclaration d'un delta

L'adaptation doit être faite à partir de la table-modèle commune de l'annexe A. L'adaptation peut être précisée en utilisant toute syntaxe équivalente à celle de la présente Norme internationale.

NOTE 1 Ainsi, l'ISO/CEI TR 14652 utilise pour l'adaptation une extension compatible de la syntaxe de la présente Norme internationale. Un delta d'adaptation peut également être exprimé à l'aide de la syntaxe de l'algorithme de tri Unicode (voir Unicode Technical Report nº 10 dans la bibliographie). On a aussi démontré la possibilité d'exprimer un delta d'adaptation par du balisage conforme à XML.

Toute déclaration de conformité à la présente Norme internationale doit être accompagnée d'une déclaration des différences entre la table de poids de tri et la table-modèle commune. Un delta doit comprendre l'équivalent de :

1. Au moins une ligne *début_ordre* valide tel que décrit en 6.3.1 ; on peut déclarer un nombre illimité de sections contenant une ligne *début_ordre* et une ligne *fin_ordre*.

2. Le nombre de niveaux de comparaison utilisés.

3. La liste des poids des *définition_de_symbole* (tel que défini en 6.3.1) ajoutés, ainsi que la position de la *définition_de_symbole* après laquelle est effectuée chaque insertion.

4. La liste des *ligne_simple* (tel que défini en 6.3.1) enlevées ou ajoutées, ainsi que la position de la *ligne_simple* de la table-modèle commune après laquelle est effectuée chaque insertion.

NOTE 2 Il est recommandé de limiter la taille d'un delta au minimum nécessaire.

Dans les cas où un processus permet à l'utilisateur d'adapter la table lui-même, une déclaration de conformité doit préciser lesquels des 4 éléments de la liste précédente peuvent être adaptés et lesquels ne peuvent l'être. Pour ceux qui ne peuvent l'être, le delta entre les éléments fixes et la table-modèle commune doit être déclaré.

NOTE 3 La déclaration peut utiliser une syntaxe différente de celle de l'article 6.3, pourvu que la relation avec cette syntaxe puisse être établie raisonnablement. Les déclarations suivantes sont ainsi correctes :

« Trier U00E5 après U00FE au premier niveau.
Trier U00E4 après U00E5 au premier niveau. »

ou

« L'ordre alphabétique primaire est changé de façon à ce que z < þ < å < ä. »

Les lettres å et ä sont maintenant triées après la lettre islandaise thorn (þ), laquelle suit toutes les variantes de la lettre z. Autrement dit, å et ä ont des poids de premier niveau supérieur à celui de þ, qui est lui-même supérieur à ceux de toutes les variantes de z.

Les deux expressions informelles précédentes peuvent raisonnablement être considérées équivalentes à l'expression suivante, plus précise (donnant des poids de niveaux 2 et 3 et tenant compte explicitement des å et ä accentués et du symbole Ångström) :

reorder-after <S00FE> *% Poids de þ (après z ; contrairement à z, þ n'a pas de variantes).*

% Déclaration de nouveaux symboles de tri (noms de poids)
collating-symbol <S00E5> *% pour å*
collating-symbol <S00E4> *% pour ä*

% Déclaration de nouveaux éléments de tri pour les décompositions (noms de sous-chaînes)
collating-element <U0061_030A> **from** "<U0061><U030A>" *% décomposition de å*
collating-element <U0041_030A> **from** "<U0041><U030A>" *% décomposition de Å*

collating-element <U0061_0308> **from** "<U0061><U0308>" *% décomposition de ä*
collating-element <U0041_0308> **from** "<U0041><U0308>" *% décomposition de Ä*

% Attribution de poids aux nouveaux symboles de tri (après þ*)*
<S00E5> *% pour å*
<S00E4> *% pour ä*
reorder-end

reorder-after <SFFFF> *% Le seul endroit où mettre la ligne début_ordre.*
order_start forward;forward;forward;forward

% Utilisation des nouveaux symboles et éléments de tri pour adapter les règles de tri

% La lettre **Å**

<U00E5> <S00E5>;<BASE>;<MIN>;<U00E5> *% LETTRE MINUSCULE LATINE A ROND EN CHEF*
<U0061_030A> <S00E5>;<BASE>;<MIN>;"<U0061_030A>" *% décomposition de å*
<U00C5> <S00E5>;<BASE>;<CAP>;<U00C5> *% LETTRE MAJUSCULE LATINE A ROND EN CHEF*
<U0041_030A> <S00E5>;<BASE>;<CAP>;"<U0041_030A>" *% décomposition de Å*
<U212B> <S00E5>;<BASE>;<CAP>;<U212B> *% SYMBOLE ANGSTRÖM* ***(la lettre Å en fait)***

<U01FB> <S00E5>;"<BASE><AIGUT>";"<MIN><MIN>";<U01FB> *% LETTRE MINUSCULE LATINE A ROND EN CHEF ET ACCENT AIGU*
<U01FA> <S00E5>;"<BASE><AIGUT>";"<CAP><MIN>";<U01FA> *% LETTRE MAJUSCULE LATINE A ROND EN CHEF ET ACCENT AIGU*

% La lettre **Ä***:*

<U00E4> <S00E4>;<BASE>;<MIN>;<U00E5> *% LETTRE MINUSCULE LATINE A TRÉMA*
<U0061_0308> <S00E4>;<BASE>;<MIN>;"<U0061_0308>" *% décomposition de ä*
<U00C4> <S00E4>;<BASE>;<CAP>;<U00C4> *% LETTRE MAJUSCULE LATINE A TRÉMA*
<U0041_0308> <S00E4>;<BASE>;<CAP>;"<U0041_0308>" *% décomposition de Ä*

<U01DF> <S00E4>;"<BASE><MACRO>";"<MIN><MIN>";<U01DF> *% LETTRE MINUSCULE LATINE A TRÉMA ET MACRON*
<U01DE> <S00E4>;"<BASE><MACRO>";"<CAP><MIN>";<U01DE> *% LETTRE MAJUSCULE LATINE A TRÉMA ET MACRON*
reorder-end

6.5 Nom de la table-modèle commune et déclaration de nom

Le nom ISO 14651_2006_TABLE1 doit être utilisé pour désigner la table-modèle commune dans toute référence externe à cette table comme base dans un contexte donné, que ce soit un processus, un contrat ou des exigences d'approvisionnement. Si un autre nom doit être utilisé pour des raisons pratiques, une déclaration de conformité doit indiquer la correspondance entre cet autre nom et le nom ISO 14651_2006_TABLE1.

L'utilisation d'un nom précis est nécessaire pour gérer les différents états de développement de cette table. Ceci découle de la nature du répertoire de caractères de référence, amenée à s'étendre pendant des années, voire des décennies.

Annexe A
(normative)

Table-modèle commune

Dans le but de minimiser les problèmes de formatage et les risques d'erreurs de reproduction, la table-modèle commune est fournie séparément, comme composante normative de la présente Norme internationale. Le nom de fichier dans cette version linguistique de la norme diffère du nom de référence normative précisé par l'article 6.5 de la présente Norme internationale à cause de l'existence de versions du fichier avec des commentaires en d'autres langues. Le fichier pour cette version linguistique peut être récupéré du site web de l'ITTF à l'URL suivant:

ISO14651_2006_TABLE1_fr.txt [URL final à établir par l'ITTF au moment de la publication]

Il existe une version anglaise officielle du fichier qui ne diffère que par les commentaires (le contenu technique est identique) et dont le nom est ISO 14651_2006_TABLE 1_en.txt

NOTE 1 La présente norme internationale déconseille mais n'empêche pas la référence aux anciennes tables ISO 14651_2000_TABLE 1 et ISO 14651_2002_TABLE 1, qui contiennent les informations de tri des répertoires des versions précédentes de l'ISO/CEI 10646 et de leurs amendements. Les anciennes tables peuvent être récupérées des URL suivants:

- [informations de tri du répertoire de l'ISO/CEI 10646-1:1993 avec ses amendements 1 à 9]
  http://www.iso.org/ittf/ISO14651_2000_TABLE1.htm.
- Informations de tri du répertoire combiné de l'ISO/CEI 10646-1:2000 et de l'ISO/CEI 10646-2:2001
  http://www.iso.org/ittf/ISO14651_2002_TABLE1_fr.txt
- [informations de tri du répertoire de l'ISO/CEI 10646:2003]
  http://www.iso.org/ittf/ISO14651_2003_TABLE1_fr.txt

La table-modèle commune actuelle prend en compte les répertoires de caractères de l'ISO/CEI 10646:2003, y compris ses deux premiers amendements, plus quatre caractères dévanâgarî, tel qu'indiqué dans le domaine d'application.

NOTE 2 Le répertoire visé par la présente norme internationale est équivalent au répertoire du standard Unicode, version 5.0 (voir la publication intitulée *The Unicode Standard Version 5.0*, publiée par le consortium Unicode).

Quand des données de tri applicables à d'autres amendements à l'ISO/CEI 10646:2003 seront disponibles, la présente Norme internationale, avec notamment sa table-modèle commune, sera amendée pour tenir compte du tri des caractères et écritures ajoutés. Pour satisfaire les exigences culturelles de communautés spécifiques, des déclarations de delta devront être appliquées à la table amendée telle que définie dans la présente Norme internationale.

La version courante de la présente Norme internationale désigne cette table sous le nom de **ISO 14651_2006_TABLE1.**

Annexe B
(informative)

Exemples de deltas d'adaptation

B.1 Exemple 1 – Adaptation minimale

L'exemple suivant est une adaptation minimale de la table-modèle commune :

```
reorder_after <SFFFF>
order_start forward;forward;forward;forward
reorder-end
```

B.2 Exemple 2 – Renversement de l'ordre des minuscules et majuscules

L'exemple suivant est une adaptation simple montrant comment inverser l'ordre des majuscules et des minuscules par rapport à celui prévu dans la table-modèle commune :

```
% Tri des majuscules avant les minuscules et
% balayage des accents vers l'avant au niveau 2.

% L'intervalle complet des symboles de poids tertiaires <MIN>..<CIRCLE>
% est déplacé après <CIRCLECAP>, de façon à ce qu'ils se trient après
% <CAP>, <WIDECAP>, <COMPATCAP>, <FONTCAP> et <CIRCLECAP> dans le même
% ordre relatif par rapport à eux-mêmes. Ceci a aussi pour effet de
% trier toutes les lettres majuscules de compatibilité avant
% leurs minuscules de compatibilité correspondantes. (Par exemple,
% U24B6 LETTRE MAJUSCULE LATINE A CERCLÉE sera triée avant
% U24D0 LETTRE MINUSCULE LATINE A CERCLÉE.

% Pour être correct, un début_ordre est
% inséré pour rendre le delta conforme.

reorder-after <CIRCLECAP>
<MIN>
<WIDE>
<COMPAT>
<FONT>
<CIRCLE>

reorder_after <SFFFF>
order_start forward;forward;forward;forward,position
reorder-end

% Fin de l'adaptation majuscules/minuscules
```

B.3 Exemple 3 – Delta et banc d'essai canadiens

Cette annexe décrit le banc d'essai 1, basé sur la norme canadienne CAN/CSA Z243.4.1-1998 (et 1992). Le delta précédant le banc d'essai *a été simplifié* pour fins d'illustration et est limité au nécessaire pour le banc d'essai. Un delta plus important, notamment pour les caractères spéciaux, est requis pour la pleine conformité avec la norme canadienne. On consultera la norme CAN/CSA Z243.4.1 pour des informations complètes. L'adaptation est appliquée à la table-modèle commune de l'annexe A, avec les différences suivantes :

1. Propriétés de traitement de niveaux :

```
forward; backward; forward; forward,position
```

2. Nombre de niveaux : 4 (inchangé).

3. Aucun changement de symbole.

4. Les changements d'ordre suivants :

- « æ » trié comme s'il s'agissait des lettres séparées « ae » au niveau 1. Les lettres « ae » se distinguent au niveau 2 du caractère « æ » et sont triées avant.
- « ð » trié comme s'il s'agissait de la lettre « d » au niveau 1. La lettre « ð » est distinguée au niveau 2 de la lettre « d » et triée après.
- « þ » trié comme s'il s'agissait des lettres séparées « th » au niveau 1. Les lettres « th » sont distinguées au niveau 2 de la lettre « þ » et triées avant.

Une adaptation canadienne exprimée dans la syntaxe d'adaptation de la présente Norme internationale (normative seulement pour l'annexe A) est :

```
% copy ISO14651_2002_TABLE1

reorder-after <SFFFF>
order_start forward;backward;forward;forward,position

<U00E6> "<S0061><S0065>";"<BASE><VRNT1><BASE>";"<MIN><COMPAT><MIN>";<U00E6>    % æ
<U00C6> "<S0061><S0065>";"<BASE><VRNT1><BASE>";"<CAP><COMPAT><CAP>";<U00C6>    % Æ

<U01E3> "<S0061><S0065>";"<BASE><VRNT1><BASE><MACRO>";"<MIN><COMPAT><MIN><MIN>";<U01E3>
         % æ AVEC MACRON
<U01E2> "<S0061><S0065>";"<BASE><VRNT1><BASE><MACRO>";"<CAP><COMPAT><CAP><MIN>";<U01E2>
         % Æ AVEC MACRON

<U01FD> "<S0061><S0065>";"<BASE><VRNT1><BASE><AIGUT>";"<MIN><COMPAT><MIN><MIN>";<U01FD> % ǽ
<U01FC> "<S0061><S0065>";"<BASE><VRNT1><BASE><AIGUT>";"<CAP><COMPAT><CAP><MIN>";<U01FC> % Ǽ


<U00F0> <S0064>;<VRNT1>;<MIN>;<U00F0>    % ð
<U00D0> <S0064>;<VRNT1>;<CAP>;<U00D0>    % Ð

<U00FE> "<S0074><S0068>";"<BASE><VRNT1><BASE>";"<MIN><COMPAT><MIN>";<U00FE>    % þ
<U00DE> "<S0074><S0068>";"<BASE><VRNT1><BASE>";"<CAP><COMPAT><CAP>";<U00DE>    % Þ

reorder-end
```

Liste en désordre (essai requis par la norme canadienne CAN/CSA Z243.4.1-1998, avec ajouts)			
ou	pêcher	Grossist	août
lésé	les	vice-presidents' offices	NOËL
péché	CÔTÉ	Copenhagen	@@@@@
vice-président	résumé	côte	L'Haÿ-les-Roses
9999	Ålborg	McArthur	CÔTE
OÙ	cañon	Mc Mahon	COTE
haïe	du	Aalborg	côté
coop	haie	Größe	coté
caennais	pécher	vice-president's offices	aide
lèse	Mc Arthur	cølibat	air
dû	cote	PÉCHÉ	vice-president
air@@@	colon	COOP	modelé
côlon	l'âme	@@@air	Thorvardur
bohème	resume	VICE-VERSA	MODÈLE
gêné	élève	gêne	maçon
meðal	Þorvarður	CO-OP	MÂCON
lamé	Canon	révélé	pèche
pêche	lame	révèle	pêché
LÈS	Bohême	çà et là	medal
vice versa	0000	MacArthur	ovoïde
C.A.F.	relève	Noël	pechère
Þorsmörk	gène	île	ode
cæsium	casanier	aïeul	péchère
resumé	élevé	Île d'Orléans	œil
Bohémien	COTÉ	nôtre	
co-op	relevé	notres	

Liste en ordre selon la norme canadienne CAN/CSA Z243.4.1-1998			
@@@@@	COOP	lamé	pêche
0000	CO-OP	les	péché
9999	Copenhagen	LÈS	PÉCHÉ
Aalborg	cote	lèse	pêché
aide	COTE	lésé	pécher
aïeul	côte	L'Haÿ-les-Roses	pêcher
air	CÔTE	MacArthur	pechère
@@@air	coté	MÂCON	péchère
air@@@	COTÉ	maçon	relève
Ålborg	côté	medal	relevé
août	CÔTÉ	meðal	resume
bohème	du	McArthur	resumé
Bohême	dû	Mc Arthur	résumé
Bohémien	élève	Mc Mahon	révèle
caennais	élevé	MODÈLE	révélé
cæsium	gène	modelé	Þorsmörk
çà et là	gêne	Noël	Thorvardur
C.A.F.	gêné	NOËL	Þorvarður
Canon	Größe	notre	vice-president
cañon	Grossist	nôtre	vice-président
casanier	haie	ode	vice-president's offices
cølibat	haïe	œil	vice-presidents' offices
colon	île	ou	vice versa
côlon	Île d'Orléans	OÙ	VICE-VERSA
coop	lame	ovoïde	
co-op	l'âme	pèche	

B.4 Exemple 4 – Delta et banc d'essai danois

L'exemple suivant est un delta d'adaptation danois. Il correspond à la norme danoise DS 377 et au « Retskrivningsordbogen », le standard orthographique danois.

```
% Cette adaptation est conforme à la norme danoise DS 377 (1980) et au
% dictionnaire orthographique danois (Retskrivningsordbogen  par 4, 1986).
% Elle est aussi en accord avec l'orthographe groenlandaise.

collating-symbol <LIGHT> % Poids symbolique plus léger que <BASE>

% Définition d'éléments de tri pour <AA> - LETTRE A ROND EN CHEF
% et combinaisons avec accents
collating-symbol <A-A> % poids symbolique pour <AA>
collating-element <A-rond-en-chef> from "<U0041><U030A>"
collating-element <a-rond-en-chef> from "<U0061><U030A>"

% Définition d'éléments de tri pour les suites a-plus-a
collating-element <A-plus-A> from "<U0041><U0041>"
collating-element <A-plus-a> from "<U0041><U0061>"
collating-element <a-plus-A> from "<U0061><U0041>"
collating-element <a-plus-a> from "<U0061><U0061>"

% Définition d'éléments de tri pour des suites combinatoires
collating-element <U-tréma> from "<U0055><U0308>"
collating-element <u-tréma> from "<U0075><U0308>"
collating-element <U-double-accent-aigu> from "<U0055><U030B>"
collating-element <u-double-accent-aigu> from "<U0075><U030B>"
collating-element <O-tréma> from "<U004F><U0308>"
collating-element <o-tréma> from "<U006F><U0308>"
collating-element <O-double-accent-aigu> from "<U004F><U030B>"
collating-element <o-double-accent-aigu> from "<U006F><U030B>"

% La ligne début_ordre obligatoire.
reorder-after <SFFFF>
order_start forward;backward;forward;forward,position

% copy ISO14651_2002_TABLE1

% Tri des majuscules avant les minuscules
% Cf. exemple 2 pour explication.
reorder-after <CIRCLECAP>
<CAP>
<WIDECAP>
<COMPATCAP>
<FONTCAP>
<CIRCLECAP>
<MIN>
<WIDE>
<COMPAT>
<FONT>
<CIRCLE>

% Introduction d'un poids plus léger que <BASE>
reorder-after <BASE>
<LIGHT>
<BASE>

% Une liste de changements de poids pour traiter des comportements spéciaux
% du danois. Ces changements définissent ou redéfinissent des liste_de_poids ;
% le bloc entier pourrait simplement être reclassé après la ligne
% début_ordre de la table. Toutefois, pour une meilleure clarté et stabilité,
% chaque jeu séparé de poids est reclassé localement dans la table autour
% de la première ligne de ce jeu de poids.

% En fait, un certain nombre d'autres changements de poids devraient être
% ajoutés par rapport à la table de l'ISO/CEI 14651 pour que tous les accents
% soient ignorés au premier niveau, alors que la table 14651 distingue diverses
% versions accentuées de <l> et d'autres lettres latines. On considère ici ce
% problème comme de moindre importance et trop complexe pour cet exemple.

% Cet exemple ne comprend pas non plus de reclassements pour des
% caractères spéciaux comme les symboles de devises SYMBOLE DOLLAR,
% SYMBOLE CENTIME et SYMBOLE LIVRE et des unités comme le SYMBOLE
```

```
% ANGSTRÖM, qui ont des poids de niveau 1 dans la TMC alors que les
% règles de la DS 377 précisent que les caractères spéciaux sont ignorés.

% Reclassement des lettres danoises après z
reorder-after <S007A> % z
<S00E6> % <AE> - LETTRE AE
<S00F8> % <O/> - LETTRE O BARRÉ OBLIQUEMENT
<A-A>   % <AA> - LETTRE A ROND EN CHEF

reorder-after <U007A> %  z – cette ligne seulement pour stabilité
% La lettre ae est une lettre distincte en danois
<U00C6> <S00E6>;<BASE>;<CAP>;<U00C6> % AE
<U00E6> <S00E6>;<BASE>;<MIN>;<U00E6> % ae
<U01FC> <S00E6>;"<BASE><AIGUT>";"<CAP><MIN>";<U01FC> % AE ACCENT AIGU
<U01FD> <S00E6>;"<BASE><AIGUT>";"<MIN><MIN>";<U01FD> % ae ACCENT AIGU

% On donne à la lettre ä le même poids de premier niveau
% qu'à æ, avec distinction au second niveau.
<U00D6> <S00E6>;"<BASE><VRNT1>";"<CAP><MIN>";<U00D6> % A TRÉMA
<U00F6> <S00E6>;"<BASE><VRNT1>";"<MIN><MIN>";<U00F6> % a TRÉMA

% Même classement pour les éléments avec accents combinatoires
<A-tréma> <S00E6>;"<BASE><VRNT1>";"<CAP><MIN>";"<U0041><U0308>"
<a-tréma> <S00E6>;"<BASE><VRNT1>";"<MIN><MIN>";"<U0061><U0308>"

% La lettre ø - O BARRÉ OBLIQUEMENT - est une lettre distincte en danois
<U00D8> <S00F8>;<BASE>;<CAP>;<U00D8> % Ø
<U00F8> <S00F8>;<BASE>;<MIN>;<U00F8> % ø
<U01FE> <S00F8>;"<BASE><AIGUT>";"<CAP><MIN>";<U01FE> % Ø ACCENT AIGU
<U01FF> <S00F8>;"<BASE><AIGUT>";"<MIN><MIN>";<U01FF> % ø ACCENT AIGU

% On donne aux lettres ö et ő% le même poids qu'à ø, avec distinction au niveau 2.
<U00D6> <S00F8>;"<BASE><VRNT1>";"<CAP><MIN>";<U00D6> % O TRÉMA
<U00F6> <S00F8>;"<BASE><VRNT1>";"<MIN><MIN>";<U00F6> % o TRÉMA
<U0150> <S00F8>;"<BASE><VRNT2>";"<CAP><MIN>";<U0150> % O DOUBLE ACCENT AIGU
<U0151> <S00F8>;"<BASE><VRNT2>";"<MIN><MIN>";<U0151> % o DOUBLE ACCENT AIGU

% Même classement pour les éléments avec accents combinatoires
<O-tréma> <S00F8>;"<BASE><VRNT1>";"<CAP><MIN>";"<U004F><U0308>"
<o-tréma> <S00F8>;"<BASE><VRNT1>";"<MIN><MIN>";"<U006F><U0308>"
<O-double-accent-aigu> <S00F8>;"<BASE><VRNT2>";"<CAP><MIN>";"<U004F><U030B>"
<o-double-accent-aigu> <S00F8>;"<BASE><VRNT2>";"<MIN><MIN>";"<U006F><U030B>"

% La lettre å - A ROND EN CHEF – est triée après la lettre ø (cf. ci-dessus)
<U00C5> <A-A>;<BASE>;<CAP>;<U00C5> % Å
<U00E5> <A-A>;<BASE>;<MIN>;<U00E5> %  å
<U01FA> <A-A>;"<BASE><AIGUT>";"<CAP><MIN>";<U01FA> % Å ACCENT AIGU
<U01FB> <A-A>;"<BASE><AIGUT>";"<MIN><MIN>";<U01FB> % å ACCENT AIGU

% Même classement pour les éléments avec accents combinatoires
<A-rond-en-chef> <A-A>;<BASE>;<CAP>;<U00C5>
<a-rond-en-chef> <A-A>;<BASE>;<MIN>;<U00E5>

% Les suites de lettres a-plus-a sont triées comme des variantes secondaires de å
<A-plus-A> <A-A>;"<BASE><VRNT1>";"<CAP><CAP>";"<U0041><U0041>" % AA
<A-plus-a> <A-A>;"<BASE><VRNT1>";"<CAP><MIN>";"<U0041><U0061>" % Aa
<a-plus-A> <A-A>;"<BASE><VRNT1>";"<MIN><CAP>";"<U0061><U0041>" % aA
<a-plus-a> <A-A>;"<BASE><VRNT1>";"<MIN><MIN>";"<U0061><U0061>" % aa

% On donne aux lettres ü et ű le même poids primaire que y, avec distinction au niveau 2
reorder-after <U00DC> % cette ligne seulement pour stabilité
<U00DC> <S0079>;"<BASE><VRNT1>";"<CAP><MIN>";<U00DC> % U TRÉMA
<U00FC> <S0079>;"<BASE><VRNT1>";"<MIN><MIN>";<U00FC> % u TRÉMA
<U0170> <S0079>;"<BASE><VRNT2>";"<CAP><MIN>";<U0170> % U DOUBLE ACCENT AIGU
<U0171> <S0079>;"<BASE><VRNT2>";"<MIN><MIN>";<U0171> % u DOUBLE ACCENT AIGU

% Même classement pour les éléments avec accents combinatoires
<U-tréma> <S0079>;"<BASE><VRNT1>";"<CAP><MIN>";"<U0055><U0308>"
<u-tréma> <S0079>;"<BASE><VRNT1>";"<MIN><MIN>";"<U0075><U0308>"
<U-double-accent-aigu> <S0079>;"<BASE><VRNT2>";"<CAP><MIN>";"<U0055><U030B>"
<u-double-accent-aigu> <S0079>;"<BASE><VRNT2>";"<MIN><MIN>";"<U0075><U030B>"

% La lettre ð a le même poids primaire que d, avec distinction au niveau 2
reorder-after <U0064> % cette ligne seulement pour stabilité
<U00D0> <S0064>;"<BASE><VRNT1>";"<CAP><MIN>";<U00D0> % ETH
<U00F0> <S0064>;"<BASE><VRNT1>";"<MIN><MIN>";<U00F0> % eth
```

```
% La lettre þ est traitée comme t + h, avec distinction au niveau 2
reorder-after <U00DE> % cette ligne seulement pour stabilité
<U00DE> "<S0074><S0068>";"<BASE><VRNT1><BASE>";"<CAP><MIN><MIN>";<U00DE> % Þ
<U00FE> "<S0074><S0068>";"<BASE><VRNT1><BASE>";"<MIN><MIN><MIN>";<U00FE> % þ

% La lettre œ est traitée comme o + e, avec distinction au niveau 2
reorder-after <U006F> % cette ligne seulement pour stabilité
<U01D2> "<S006F><S0065>";"<BASE><LIGHT>";"<CAP><MIN>";<U01D2> % Œ
<U01D3> "<S006F><S0065>";"<BASE><LIGHT>";"<MIN><MIN>";<U01D3> % œ

% On donne à l'espace, au trait d'union-signe moins, au trait d'union
% et à la barre oblique des poids de niveau 1 avant toute lettre ou chiffre,
% le trait d'union-signe moins et la barre oblique ayant des distinctions
% de niveau 2 avec l'espace

reorder-after <U0020> % cette ligne seulement pour stabilité
<U0020> <S0020>;<BASE>;<MIN>;<U0020> % ESPACE
<U002D> <S0020>;"<BASE><VRNT1>";"<U002D><MIN>";<U002D> % TRAIT D'UNION-SIGNE MOINS
<U2010> <S0020>;"<BASE><VRNT1>";"<U0010><MIN>";<U2010> % TRAIT D'UNION
<U002F> <S0020>;"<BASE><VRNT2>";"<U002F><MIN>";<U002F> % BARRE OBLIQUE

% La lettre ĸ (kra groenlandais) est traitée comme un q minuscule, avec une
% distinction de niveau 2.
reorder-after <U0071> % q - cette ligne seulement pour stabilité
<U0138> <S0071>;"<BASE><VRNT1>";"<MIN><MIN>";<U0138> % ĸ

% La lettre ß est traitée comme s + s, avec une distinction de niveau 2
% la plaçant avant – le court précède le long.
reorder-after <U0073> % s - cette ligne seulement pour stabilité
<U00DF> "<S0073><S0073>;"<BASE><LIGHT>";"<MIN><MIN>";<U00DF> % ß

% On donne des poids de niveau 4 à tous les caractères de commande pour
% assurer un tri déterministe.
reorder-after <U0000>
<U0000>..<U001F> IGNORE;IGNORE;IGNORE;<U0000>..<U001F>
<U007F>..<U009F> IGNORE;IGNORE;IGNORE;<U007F>..<U009F>

reorder-end

% Fin de l'exemple d'adaptation pour le danois
```

Banc d'essai pour le danois (en ordre)			
A/S	D.S.B.	RÉE, A	STORM PETERSEN
ANDRE	DSC	REE, B	STORMLY
ANDRÉ	EKSTRA-ARBEJDE	RÉE, L	THORVALD
ANDREAS	EKSTRABUD	REE, V	THORVARDUR
AS	EKSTRAARBEJDE	SCHYTT, B	ÞORVARÐUR
CA	HØST	SCHYTT, H	THYGESEN
ÇA	HAAG	SCHÜTT, H	VESTERGÅRD, A
CB	HÅNDBOG	SCHYTT, L	VESTERGAARD, A
ÇC	HAANDVÆRKSBANKEN	SCHÜTT, M	VESTERGÅRD, B
DA	Karl	ß	ÆBLE
ÐA	Karl	SS	ÄBLE
DB	NIELS JØRGEN	SSA	ØBERG
ÐC	NIELS-JØRGEN	STORE VILDMOSE	ÖBERG
DSB	NIELSEN	STOREKÆR	Århus

B.5 Exemple 5 – Adaptation pour le khmer

L'écriture khmère est utilisée principalement au Cambodge. L'adaptation qui suit n'est pas incluse dans la TMC (voir annexe A) elle-même pour conserver la simplicité de la TMC, spécialement pour les formes de lettres rares (par exemple le ROBAT khmer pour lequel l'adaptation suivante peut ne pas être désirable dans certains cas, puisque ce caractère n'est que très rarement utilisé, mais que l'adaptation permettant de le traiter correctement peut affecter l'efficacité du tri, même pour des textes ne contenant aucun ROBAT).

```
reorder-after <MAX>
% Khmer:

collating-symbol <S1794_S17C9> % LETTRE KHMÈRE BA, SIGNE KHMER MOÛSÉKETOENMOÛSÉKETOEN
collating-symbol <S1794_S17CA> % LETTRE KHMÈRE BA, SIGNE KHMER TREISAP

collating-symbol <S17BB_S17C6> % VOYELLE DIACRITIQUE KHMÈRE OU, SIGNE KHMER NIKAHIT
collating-symbol <S17B6_S17C6> % VOYELLE DIACRITIQUE KHMÈRE AA, SIGNE KHMER NIKAHIT

collating-symbol <C1780>..<C179C>

        % Déclaration des contractions khmères

collating-element <U1794_17C9> from "<U1794><U17C9>" % LETTRE KHMÈRE BA, SIGNE KHMER MOÛSÉKETOEN
collating-element <U1794_17CA> from "<U1794><U17CA>" % LETTRE KHMÈRE BA, SIGNE KHMER TREISAP

collating-element <SW_17CC_1780>..<SW_17CC_17A2> from "<U1780>..<U17A2><U17CC>"
        % LETTRE KHMÈRE KA, SIGNE KHMER ROBAT.. LETTRE KHMÈRE ‘A, SIGNE KHMER ROBAT
collating-element <SW_17CC_17A5>..<SW_17CC_17B3> from "<U17A5>..<U17B3><U17CC>"
        % VOYELLE PLEINE KHMÈRE ‘I, SIGNE KHMER ROBAT..VOYELLE PLEINE KHMÈRE ‘AOU, SIGNE KHMER
        ROBAT

collating-element <U17C6_17BB> from "<U17BB><U17C6>" % VOYELLE DIACRITIQUE KHMÈRE OU, SIGNE KHMER
        NIKAHIT (OM correctement orthographié)
collating-element <U17BB_17C6> from "<U17C6><U17BB>" % SIGNE KHMER NIKAHIT, VOYELLE DIACRITIQUE
        KHMÈRE OU (OM dans la mauvaise séquence de caractères)
collating-element <U17C6_17B6> from "<U17B6><U17C6>" % VOYELLE DIACRITIQUE KHMÈRE AA, SIGNE KHMER
        NIKAHIT (AM correctement orthographié)
collating-element <U17B6_17C6> from "<U17C6><U17B6>" % SIGNE KHMER NIKAHIT, VOYELLE DIACRITIQUE
        KHMÈRE AA (AM dans la mauvaise séquence de caractères)

collating-element <U17D2_1780>..<U17D2_179C> from "<U17D2><U1780>..<U179C>"
        % SIGNE KHMER TCHOENG, LETTRE KHMÈRE KA..SIGNE KHMER TCHOENG, LETTRE KHMÈRE ‘A
collating-element <U17D2_17A5>..<U17D2_17B3> from "<U17D2><U17A5>..<U17B3>"
        % SIGNE KHMER TCHOENG, VOYELLE PLEINE KHMÈRE I..SIGNE KHMER TCHOENG, VOYELLE PLEINE KHMÈRE
        ‘AOU



reorder-after <S1794> % LETTRE KHMÈRE BA
<S1794_17C9> % LETTRE KHMÈRE BA, SIGNE KHMER MOÛSÉKETOEN
<S1794_17CA> % LETTRE KHMÈRE BA, SIGNE KHMER TREISAP


reorder-after <S17C5> VOYELLE DIACRITIQUE KHMÈRE AOU
<S17BB_17C6> % VOYELLE DIACRITIQUE KHMÈRE OU, SIGNE KHMER NIKAHIT
reorder-after <S17C6> SIGNE KHMER NIKAHIT
<S17B6_17C6> % VOYELLE DIACRITIQUE KHMÈRE AA, SIGNE KHMER NIKAHIT



reorder-after <S17D2>
<C1780>..<C1794> % SIGNE KHMER TCHOENG, LETTRE KHMÈRE KA..SIGNE KHMER TCHOENG, LETTRE KHMÈRE BA
<C1795>..<C179A> % SIGNE KHMER TCHOENG, LETTRE KHMÈRE PHA..SIGNE KHMER TCHOENG, LETTRE KHMÈRE RO
<C17AB> % SIGNE KHMER TCHOENG, VOYELLE PLEINE KHMÈRE RY
<C17AC> % SIGNE KHMER TCHOENG, VOYELLE PLEINE KHMÈRE RYY
<C179B> % SIGNE KHMER TCHOENG, LETTRE KHMÈRE LO
<C17AD> % SIGNE KHMER TCHOENG, VOYELLE PLEINE KHMÈRE LY
<C17AE> % SIGNE KHMER TCHOENG, VOYELLE PLEINE KHMÈRE LYY
<C179C>..<C17A2> % SIGNE KHMER TCHOENG, LETTRE KHMÈRE VO..SIGNE KHMER TCHOENG, LETTRE KHMÈRE ‘A



reorder-after <SFFFF>

order_start forward;forward;forward;forward
<U1794_17C9> <S1794_17C9>;<BASE>;<MIN>;<U1794_17C9> % LETTRE KHMÈRE BA, SIGNE KHMER MOÛSÉKETOEN
<U1794_17CA> <S1794_17CA>;<BASE>;<MIN>;<U1794_17CA> % LETTRE KHMÈRE BA, SIGNE KHMER TREISAP

%% Les contractions du ROBAT doivent être utilisées seulement en mode « avancé » d'adaptation
%% pour le khmer, parce que le ROBAT est rarement utilisé et que ces contractions
%% peuvent affecter l'efficacité de calcul de la clé même dans les cas où le ROBAT
%% ne survient pas, ces contractions commençant par des lettres très utilisées
```

```
<SW_17CC_1780>..<SW_17CC_17A2> "<S179A><S17D2><S1780>..<S17A2>";
        "<BASE><VRNT1><BASE><BASE>";"<MIN><MIN><MIN><MIN>"; <SW_17CC_1780>..<SW_17CC_17A2>
        % LETTRE KHMÈRE KA, SIGNE KHMER ROBAT..LETTRE KHMÈRE 'A, SIGNE KHMER ROBAT

<SW_17CC_17A5>..<SW_17CC_17A6> "<S179A><S17D2><S17A2><S17B7>..<S17B8>";
        "<BASE><VRNT1><BASE><BASE><VRNT1><BASE>";"<MIN><MIN><MIN><MIN><MIN><MIN>";
        <SW_17CC_17A5>..<SW_17CC_17A6> % VOYELLE PLEINE KHMÈRE 'I, SIGNE KHMER ROBAT..VOYELLE
        PLEINE KHMÈRE 'II, SIGNE KHMER ROBAT

<SW_17CC_17A7> "<S179A><S17D2><S17A2><S17BB>";"<BASE><VRNT1><BASE><BASE><VRNT1><BASE>";
        "<MIN><MIN><MIN><MIN><MIN><MIN>";<SW_17CC_17A7>
        % VOYELLE PLEINE KHMÈRE 'OU, SIGNE KHMER ROBAT

<SW_17CC_17A8> "<S179A><S17D2><S17A2><S17BB>";"<BASE><VRNT1><BASE><BASE><VRNT2><BASE>";
        "<MIN><MIN><MIN><MIN><MIN><MIN>";<SW_17CC_17A8>
        % VOYELLE PLEINE KHMÈRE 'OUK; SIGNE KHMER ROBAT

<SW_17CC_17A9> "<S179A><S17D2><S17A2><S17BC>";"<BASE><VRNT1><BASE><BASE><VRNT1><BASE>";
        "<MIN><MIN><MIN><MIN><MIN><MIN>";<SW_17CC_17A9>
        % VOYELLE PLEINE KHMÈRE 'OÛ; SIGNE KHMER ROBAT

<SW_17CC_17AA> "<S179A><S17D2><S17A2><S17BC>";"<BASE><VRNT1><BASE><BASE><VRNT2><BASE>";
        "<MIN><MIN><MIN><MIN><MIN><MIN>";<SW_17CC_17AA>
        % VOYELLE PLEINE KHMÈRE 'EOU; SIGNE KHMER ROBAT

<SW_17CC_17AF>..<SW_17CC_17B1> "<S179A><S17D2><S17A2><S17C2>..<S17C4>";
        "<BASE><VRNT1><BASE><BASE><VRNT1><BASE>";"<MIN><MIN><MIN><MIN><MIN><MIN>";
        <SW_17CC_17AF>..<SW_17CC_17B1> % VOYELLE PLEINE KHMÈRE 'ÈÈ, SIGNE KHMER ROBAT..VOYELLE
        PLEINE KHMÈRE 'OO TYPE UN, SIGNE KHMER ROBAT

<SW_17CC_17B2> "<S179A><S17D2><S17A2><S17C4>";"<BASE><VRNT1><BASE><BASE><VRNT2><BASE>";
        "<MIN><MIN><MIN><MIN><MIN><MIN>";<SW_17CC_17B2>
        % VOYELLE PLEINE KHMÈRE 'OO TYPE DEUX; SIGNE KHMER ROBAT

<SW_17CC_17B3> "<S179A><S17D2><S17A2><S17C5>";"<BASE><VRNT1><BASE><BASE><VRNT1><BASE>";
        "<MIN><MIN><MIN><MIN><MIN><MIN>";<SW_17CC_17B3>
        % VOYELLE PLEINE KHMÈRE 'AOU; SIGNE KHMER ROBAT


%%% OM et ÂM khmers (le NIKAHIT devrait être écrit après la voyelle):

<U17BB_17C6> <S17BB_17C6>;<BASE>;<MIN>;<U17BB_17C6> % VOYELLE DIACRITIQUE KHMÈRE OU, SIGNE KHMER
        NIKAHIT
<U17C6_17BB> <S17BB_17C6>;<BASE>;<MIN>;<U17C6_17BB> % SIGNE KHMER NIKAHIT, VOYELLE DIACRITIQUE
        KHMÈRE OU

<U17B6_17C6> <S17B6_17C6>;<BASE>;<MIN>;<U17B6_17C6> % VOYELLE DIACRITIQUE KHMÈRE AA, SIGNE KHMER
        NIKAHIT
<U17C6_17B6> <S17B6_17C6>;<BASE>;<MIN>;<U17C6_17B6> % SIGNE KHMER NIKAHIT, VOYELLE DIACRITIQUE
        KHMÈRE AA


reorder-end
```

Annexe C
(informative)
Prétraitement

C.1 Généralités

Le prétraitement ne s'avère nécessaire que pour la modification ou la reproduction de chaînes originales, dans le but de les rendre indépendantes du contexte avant la comparaison. Un prétraitement sans reproduction transforme une chaîne donnée en une autre chaîne ; ce prétraitement peut se conjuguer avec la création de clé et la comparaison, opérations requises par exemple pour le thaï ou le lao (cf. l'annexe C.2) ou pour le tri correct de nombres (cf. l'annexe C.3). Un prétraitement comportant une reproduction peut transformer une chaîne donnée en plusieurs chaînes (à trier).

Quelques exemples de prétraitement sans reproduction :

- Réarrangement des voyelles et des consonnes, nécessaire pour le thaï (cf. C.2) et le lao.
- Transformation de nombres afin qu'ils soient triés en ordre numérique plutôt que positionnel (cf. C.3). Le tri numérique est une opération délicate et exige des soins particuliers dans bien des cas.
- Suppression ou rotation de caractères nuisibles à des tris spéciaux, comme le traitement des articles (en fonction de la langue) dans les titres de livres :

  ```
  Marquis de Saint-Évremond, Le
  ```

- Transformation de données abrégées en des formes complètes. Exemple : transformation de « McArthur » en « MacArthur ».

Exemples de prétraitement avec reproduction :

- Reproduction d'une chaîne en plusieurs « rotations », comme dans la production de mots-clés en contexte :

  ```
                                      Classement international de chaînes
  Classement international de         chaînes
  Classement                          international de chaînes
  ```

- Reproduction d'une chaîne telle que « 41 » en l'épelant en plusieurs langues (ici gaélique irlandais, allemand, anglais et français) :

  ```
  daichead a haon
  einundvierzig
  forty-one
  quarante et un
  ```

C.2 Tri de chaînes de caractères thaïs

Cette annexe explique quelques principes sous-jacents à l'adaptation de la TMC donnée à l'annexe B.5 ci-avant, de même que le classement de la table TMC pour le thaï (et jusqu'à un certain point pour le lao).

C.2.1 Principes de classement du thaï

Le standard largement accepté pour le tri lexicographique thaï est décrit dans le dictionnaire de l'Institut royal, édition de l'an 2542 de l'ère bouddhiste (1999 de l'ère chrétienne), qui est le dictionnaire officiel thaï. Les principes de classement, étendus pour couvrir entièrement l'écriture thaïe et son informatisation, en sont :

- Les mots sont triés en ordre alphabétique et non phonétique. L'ordre des consonnes est le suivant :

ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท
ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ฤ ล ฦ ว ศ ษ ส ห ฬ อ ฮ

- Les voyelles et le nikhahit, sont aussi triés selon leur forme écrite et non en fonction de leur prononciation. Leur ordre de classement est le suivant :
  - ◌ำ ◌ะ ◌ั ◌า ◌ๅ ◌ำ ◌ิ ◌ี ◌ึ ◌ื ◌ุ ◌ู เ- แ- โ- ใ- ไ- ◌ฺ
  - อ ว ย sont toujours triés comme consonnes, même s'ils agissent parfois comme voyelles.
  - ◌ๅ est une variante hampée de ◌า, utilisée avec les consonnes hampées ฤ et ฦ: ฤๅ et ฦๅ.
  - ◌ำ est un ◌า suivi « logiquement » d'un ◌ั. Toutefois la décomposition de compatibilité Unicode du caractère précomposé se fait vers un ◌ั suivi d'un ◌า, pour faire en sorte que la faute orthographique puisse également être traitée.
- Aucune analyse de la structure syllabique ou de la frontière des mots n'est nécessaire, puisque les lexiques thaïs sont triés en ordre alphabétique et non phonétique.. Soulignons que le thaï n'utilise normalement aucun séparateur de mot, sauf, et très exceptionnellement, l'espace sans chasse.
- La comparaison de chaînes se fait de gauche à droite, mais en traitant les voyelles initiales (เ- แ- โ- ใ- ไ-, qui correspondent aux caractères U0E40-U0E44), comme si elles suivaient la consonne qui les suit immédiatement. En conséquence, on doit effectuer un réarrangement (d'une manière ou d'une autre) avant la comparaison.
- les tons et les diacritiques sont ignorés au niveau 1. Au niveau 2 leur ordre de tri est le suivant:
  ◌็ ◌่ ◌้ ◌๊ ◌๋ ◌์ ◌ฺ
- Puisque le thaï, le lao et le khmer ignorent le concept de casse, il peut a priori sembler que le troisième niveau est inutile pour le tri de ces écritures. Toutefois, on l'utilise pour distinguer le LAKKHANGYAO (◌ๅ) comme variante du SARA Â (◌า), puisque des variantes similaires d'autres systèmes d'écriture sont distinguées de la même manière au niveau 3, tout comme d'autres cas de variantes en thaï, en lao et en khmer.
- Quand les signes de ponctuation thaïs (◌๎ ฯ ๏ ๚ ๛) entrent en jeu, un autre niveau de classement est nécessaire pour les traiter. Il correspond au quatrième niveau de la table-modèle commune. Dans le classement des chaînes, les signes de ponctuation sont moins significatifs que tout signe de ton ou tout diacritique, et ils doivent être ignorés aux trois premiers niveaux. Soulignons que le PAÏYANNOÏ (ฯ) et la LETTRE THAÏE MAÏYAMOK (marque de répétition ๆ) sont considérés et triés comme des signes de ponctuation, et non comme lettres, malgré leur catégorie générale Unicode, respectivement comme « Lo » et « Lm ». Par exemple, “ข้างๆ, ข้างกบ, ข้างๆ คูๆ, ข้างจัน” constitue un ordre correct dans le dictionnaire de l'Institut royal. Au premier niveau, les poids considérés correspondent respectivement à ceux de ขาง, ขางกบ, ขางคู, et ขางจัน.
- Les dix chiffres décimaux thaïs (๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙), sont sémantiquement équivalents aux dix chiffres arabes 0-9, respectivement. Leur poids est donc équivalent à leur équivalent en chiffre arabe au premier niveau, et différent au deuxième niveau, pour distinguer le système d'écriture.

C.2.2 Réarrangement voyelle-consonne

En ce qui concerne le traitement des voyelles antéposées, il est nécessaire d'avoir recours soit à une préparation du tri, soit à un groupage des éléments de tri (comme dans l'adaptation de l'annexe B.5 ci-avant). La préparation du tri balaie la chaîne une fois et déplace chaque voyelle thaïe initiale après le caractère qui la suit (idéalement seulement si ce dernier est une consonne thaïe). La chaîne ainsi préparée est alors soumise au processus normal d'attribution des poids de comparaison. Une autre façon de gérer cette situation consiste à

former des éléments de tri appropriés – ce qui constitue la pratique adoptée autant par la TMC de la présente norme et par la table de tri de l'algorithme de classement d'Unicode (UTS #10). Toute paire possible de voyelle initiale suivie d'une consonne est définie comme `collating-element`, dont le poids est égal à celui de sous-chaînes réarrangées. En outre, puisque deux เ de suite ont la même apparence que แ, ils doivent être traités de manière similaire. Soulignons que le réarrangement de chaque voyelle initiale ne se fait qu'avec la consonne qui la suit et qui lui correspond. Aucune analyse de groupe de consonnes n'est nécessaire. Au contraire, une telle analyse mènerait à des ambiguïtés ou conduirait à un ordre différent de celui précisé dans le Dictionnaire de l'institut royal. À titre d'exemple:

1. Ambiguïtés – le problème d'ambiguïté est illustré par le mot « เพลา », qui peut se prononcer de deux façons : un mot de deux syllabes, « phé-la » (qui signifie « temps ») ou un mot d'une syllabe, « phlao » (« essieu » ou « faiblir »). Un algorithme de réarrangement qui épouse la prononciation distincte du groupe potentiel « พล » de cette chaîne produirait deux clefs distinctes, « พเลา » et « พลเา », et par conséquent en deux poids distincts, qui sont également valables. Pour que l'on puisse les trier adéquatement, par contre, les deux mots doivent avoir le même poids..

2. Tri non conforme – pour illustrer la différence de tri provoquée par le traitement de groupes de consonnes, considérons les mots « เพล, เพลง, เพศ », triés ici dans un ordre conforme. Le réarrangement correct ignore tout groupe pour produire : « พเล, พเลง, พเศ », dans l'ordre. Toutefois, si les paires de consonnes qui forment des groupes valables étaient rassemblées comme éléments simples de tri (sans égard à la prononciation, qui est potentiellement ambiguë), le résultat du réarrangement donnerait « <พล>เ, <พล>เง, พเศ », ce qui conduirait à l'ordre non conforme « เพศ, เพล, เพลง ». Encore une fois, si les groupes de consonnes étaient rassemblés comme éléments simples de tri (avec un effort quelconque de désambiguïsation), le résultat du réarrangement donnerait « พเล, <พล>เง, พเศ », ce qui conduirait à l'ordre non conforme « เพล, เพศ, เพลง ».

C.2.3 Exemple de chaînes triées

Voici un exemple de tri correct :

Exemple de tri en thaï (ordre correct)			
กก	โกน	แข่งขัน	ผัด
กรรม	โกร๋น	แขน	ฯพณฯ
กรรม์	ใกล้	ครรภ-	พณิชย์
-กระแย่ง	ไก่	ครรภ์	ย่อง
กราบ	ไกล	จุมพล	รอง
กะเกณฑ์	ขัน	จุ๋พล	ฤทธิ์
กัก	ขนาบ	ชาย	ฤษี
ก้าว	ข้าง	เฒ่า	ฤๅษี
กำ	ข้างๆ	เณร	ลลิตา
กิน	ข้างกระดาน	ตลาด	ภาชา
กี่	ข้างขึ้น	ทูลเกล้า	วก
กี๋น	ข้างควาย	ทูลเกล้าฯ	ศาล
กุน	ข้างๆ คูๆ	ทูลเกล้าทูลกระหม่อม	หริภุญชัย
กูด	ข้างเงิน	น้า	หฤทัย
เก้ง	ข้างออก	น้ำ	หลง
เกล้า	เข็ด	นี้	แหง่
เกลียว	เขน	บุญหลง	แห่ง
เก้า	เข็น	บุญ-หลง	แหนม
เกาะ	เข่น	ปา	แหนหวง
เกี่ยว	แข็ง	ป่า	แหบ
เกี๊ยะ	แข่ง	ป้า	แหม
เกือก	แข้ง	ป๊า	อาน
แกง	แข้งขวา	ป๋า	ฮา
แกะ	แข็งขัน	ปาน	

C.3 Traitement de sous-chaînes numériques dans le tri

Un numéral est une chaîne représentant un nombre. Les exemples qui suivent traitent de numéraux représentant des nombres de l'ensemble des réels (en fait de sous-ensemble des réels), qui ont un ordre prédéterminé. Seuls les numéraux décimaux sont considérés ici.

La présentation qui suit traitera d'abord des numéraux décimaux en système positionnel, utilisant les chiffre 0-9. Elle passera ensuite aux numéraux pour les nombres entiers, aux numéraux avec parties fractionnaires puis aux exposants. On y trouve également une brève discussion des numéraux avec des chiffres d'autres écritures, des écritures utilisant parfois une autre syntaxe avec des chiffres pour les numéraux (comme les chiffres Han) et des chiffres romains. En certaines circonstances, comme dans les numéros de pièces et les numéros de téléphone, les chiffres ne représentent pas des nombres. Les prétraitements décrits ci-dessous ont des conséquences indésirables dans ces cas et devraient alors être évités.

C.3.1 Traitement des nombres « ordinaires » pour les entiers naturels

La table-modèle commune ne permet pas de trier des chaînes contenant des numéraux de telle façon que l'ordre reflète la valeur numérique des numéraux. Par exemple, soit la suite aléatoire de chaînes suivante :

Livraison 1
Livraison 20
Livraison 12
Livraison 2
Livraison 9

la méthode de classement de la présente Norme internationale les trie dans l'ordre suivant :

Livraison 1
Livraison 12
Livraison 2
Livraison 20
Livraison 9

(Il suffit d'examiner le premier chiffre de chaque nombre pour comprendre la raison de cet ordre.) Un ordre plus acceptable serait :

Livraison 1
Livraison 2
Livraison 9
Livraison 12
Livraison 20

Il est impossible d'adapter la table-modèle commune de la présente Norme internationale pour obtenir cet ordre. Toutefois, on peut utiliser le prétraitement avant l'étape de tri pour obtenir le résultat voulu : l'ordre numérique. Les chaînes prétraitées ne sont normalement pas montrées à l'utilisateur ; seules les chaînes originales le sont. Les chaînes prétraitées ne s'utilisent normalement qu'afin de produire des clés de tri. Une méthode simple, mais pas très générale, de prétraiter des numéraux pour le tri en ordre numérique est de les bourrer avec des zéros initiaux, jusqu'à un nombre donné de chiffres. En bourrant l'exemple original pour former des nombres à trois chiffres, on obtient :

Livraison 001
Livraison 020
Livraison 012
Livraison 002
Livraison 009

La table-modèle commune de la présente Norme internationale permet alors de trier les chaînes dans un meilleur ordre (on montre ici les chaînes prétraitées, contrairement à l'usage habituel) :

Livraison 001
Livraison 002
Livraison 009
Livraison 012
Livraison 020

Cette approche présente toutefois deux problèmes :

- Il faut déterminer à l'avance le nombre (habituellement petit) de chiffres de bourrage. Si ce nombre est trop grand, les chaînes prétraitées peuvent devenir assez longues, surtout si elles contiennent plusieurs numéraux. Si le nombre de chiffres de bourrage choisi est trop petit, on risque de rencontrer des numéraux avec plus de chiffres, on revient donc partiellement à la situation originale où le tri ne suit pas complètement l'ordre numérique.
- Si certains des numéraux originaux sont déjà précédés de zéros, on observe une perte de déterminisme. Ainsi, avec les chaînes originales :

  Livraison 01
  Livraison 1

  les chaînes prétraitées sont identiques et le résultat final, tel que vu par l'utilisateur, peut être soit

  Livraison 01
  Livraison 1

  soit

  Livraison 1
  Livraison 01

  et l'ordre relatif peut être différent pour différentes apparitions des numéraux ou différentes passes du processus de tri appliquant les mêmes règles. Cette perte de déterminisme est indésirable.

Plusieurs méthodes permettent de régler ces problèmes. En voici une :

À chaque sous-chaîne maximale de chiffres, on adjoint un préfixe formé d'un nombre fixe de chiffres représentant le nombre de chiffres dans la sous-chaîne originale. Dans la plupart des cas un préfixe à deux chiffres suffira, permettant jusqu'à 99 chiffres dans les numéraux originaux. Ainsi, avec ces chaînes originales :

Livraison 1
Livraison 01
Livraison 20
Livraison 12
Livraison 2
Livraison 09
Livraison 9

on obtient après prétraitement les chaînes suivantes :

Livraison 011
Livraison 0201
Livraison 0220
Livraison 0212
Livraison 012
Livraison 0209
Livraison 019

qui seront triées comme suit par le mécanisme de base de la présente Norme internationale :

Livraison 011
Livraison 012
Livraison 019
Livraison 0201

Livraison 0209
Livraison 0212
Livraison 0220

et seront normalement présentées à l'utilisateur comme :

Livraison 1
Livraison 2
Livraison 9
Livraison 01
Livraison 09
Livraison 12
Livraison 20

En raison du bourrage par des zéros initiaux, cette méthode de tri regroupe les numéraux ayant un même nombre de chiffres, même si leurs valeurs sont très différentes. S'il est préférable de trier et de regrouper par *valeur*, il vaut mieux reproduire le numéral : d'abord un compte des chiffres du numéral amputé des zéros initiaux, ensuite le numéral amputé lui-même et enfin le numéral original. Cette reproduction est nécessaire pour conserver le déterminisme du tri par rapport aux chaînes originales. On aura donc en utilisant les mêmes chaînes originales que ci-dessus :

Livraison 011 1
Livraison 011 01
Livraison 0220 20
Livraison 0212 12
Livraison 012 2
Livraison 019 09
Livraison 019 9

qui seront triées comme suit par le mécanisme de base de la présente Norme internationale :

Livraison 011 01
Livraison 011 1
Livraison 012 2
Livraison 019 09
Livraison 019 9
Livraison 0212 12
Livraison 0220 20

et seront normalement présentées à l'utilisateur comme :

Livraison 01
Livraison 1
Livraison 2
Livraison 09
Livraison 9
Livraison 12
Livraison 20

Les numéraux avec des zéros de bourrage se retrouvent toujours devant les numéraux sans (ou avec moins de) bourrage. Le prétraitement pourrait déplacer les numéraux originaux (en ordre d'apparition) en fin de chaîne, si l'on souhaite donner au bourrage un poids inférieur à celui du texte qui suit les numéraux.

La présence de plusieurs numéraux dans chaque chaîne n'ajoute rien au problème.

La prise en charge des numéraux naturels suffit dans la plupart des cas et on recommande de le faire dans le cadre du prétraitement usuel de chaînes à trier. Ce prétraitement n'est toutefois pas requis par la présente Norme internationale.

C.3.2 Traitement des nombres positionnels dans les autres écritures

L'ISO/CEI 10646 code des chiffres décimaux pour plusieurs écritures. Dans la plupart des cas ces chiffres sont utilisés dans le cadre d'un système positionnel, tout comme les chiffres 0-9. Toutefois, il convient de ne pas considérer une suite de numéraux de différentes écritures comme un seul numéral ; on doit plutôt considérer chaque sous-chaîne de chiffres d'une même écriture comme un numéral distinct.

C.3.3 Traitement des systèmes de numération non positionnels (p. ex. chiffres romains)

En chinois et en certaines autres langues, on trouvera des chiffres décimaux (en écriture Han, par exemple) mêlés à des idéogrammes signifiant « mille », « dix » etc. Si de tels numéraux doivent être triés selon leurs valeurs numériques, on pourra procéder comme ci-dessus mais en ajoutant une étape après la reproduction initiale : la conversion de la copie au système positionnel dans la syntaxe utilisée ici pour les numéraux complets.

Les chiffres romains peuvent être traités de façon similaire : reproduction, suivie d'une transformation de la première copie au système décimal positionnel. Par exemple « Louis V », où le V est un chiffre romain, sera prétraité en « Louis 5 V ».

Attention : dans de tels cas une intervention humaine ou d'un système expert peut être nécessaire, en effet CHAPITRE DIX peut signifier CHAPITRE 10 ou CHAPITRE 509.

C.3.4 Traitement des nombres entiers

Lorsque des numéraux représentant des nombres entiers négatifs doivent être triés selon leurs valeurs, on doit d'abord tenir compte de la représentation adoptée pour la négativité. Le plus souvent, les nombres négatifs sont indiqués par un signe de négation en préfixe. Ce signe de négation peut être TRAIT D'UNION-SIGNE MOINS U002D (mais ce caractère représente souvent un trait d'union plutôt qu'une négation) ou SIGNE MOINS U2212. Il existe toutefois d'autres conventions : le signe de négation peut parfois être BARRE OBLIQUE U002F ou SYMBOLE POUR CENT U0025 ; le signe de négation peut être en suffixe plutôt qu'en préfixe ; la négativité peut être indiquée en mettant les chiffres entre parenthèses ou en les imprimant d'une couleur contrastante (souvent le rouge). Les exemples qui suivent ne prennent en compte que le cas du préfixe SIGNE MOINS. La positivité est indiquée par l'absence du SIGNE MOINS ou par la présence d'un préfixe SIGNE PLUS U002B.

Exemples de chaînes :

Température : –9 °C
Température : 0 °C
Température : –14 °C
Température : 05 °C
Température : +5 °C
Température : –0 °C
Température : –09 °C
Température : 105 °C
Température : +05 °C
Température : 5 °C

Il est possible d'obtenir des résultats acceptables et déterministes pour les nombres entiers (en utilisant cette syntaxe) à l'aide du prétraitement suivant (les mises en œuvre peuvent optimiser ce processus mais elle doivent fournir un résultat équivalent) :

1. Reproduire les numéraux dans la chaîne (avec leurs signes), en plaçant une copie de chacun, en ordre d'apparition, à la toute fin de la chaîne ; cette copie demeure intacte dans les étapes suivantes. Cette étape garantit le déterminisme.

2. S'assurer que tous les numéraux ont un signe (« + » ou « - »), l'ajouter au besoin.

3. Enlever tous les zéros initiaux (en choisissant systématiquement de représenter 0 soit par un seul chiffre zéro, soit par une chaîne vide) ; on peut aussi choisir de laisser (ou d'ajouter au besoin) exactement un zéro de bourrage.

4. Entre le signe et les chiffres de chaque numéral, insérer deux chiffres représentant le nombre de chiffres de chaque numéral (après effacement des zéros initiaux).

5. Pour chaque numéral négatif, remplacer chaque chiffre par son complément-9. Le complément-9 d'un chiffre x est 9–*x*. Le complément-9 de 0 est donc 9, de 9 : 0, de 5 : 4, etc.

Le tri doit s'accompagner d'une adaptation de la table donnée dans la présente Norme internationale de sorte que le SIGNE PLUS et le SIGNE MOINS soient significatifs au même niveau que les chiffres et que le SIGNE MOINS ait un poids inférieur au SIGNE PLUS. (Dans l'exemple ci-dessous, le poids de SIGNE PLUS est inférieur au poids de 0 ; un tel choix n'est pas essentiel à l'obtention d'un ordre acceptable.)

Nos chaînes d'exemple après prétraitement :

Température : –980 °C –9
Température : +00 °C 0
Température : –9785 °C –14
Température : +015 °C 05
Température : +015 °C +5
Température : –99 °C –0
Température : –980 °C –09
Température : +03105 °C 105
Température : +015 °C +05
Température : +015 °C 5

Une fois triées par le mécanisme de base de la présente Norme internationale :

Température : –9785 °C –14
Température : –980 °C –09
Température : –980 °C –9
Température : –99 °C –0
Température : +00 °C 0
Température : +015 °C +05
Température : +015 °C +5
Température : +015 °C 05
Température : +015 °C 5
Température : +03105 °C 105

Telle que normalement présentées à l'utilisateur :

Température : –14 °C
Température : –09 °C
Température : –9 °C
Température : –0 °C
Température : 0 °C
Température : +05 °C
Température : +5 °C
Température : 05 °C
Température : 5 °C
Température : 105 °C

Ce prétraitement trie en ordre déterministe les chaînes contenant un ou plusieurs nombres entiers, en respectant l'ordre numérique.

La procédure pour d'autres syntaxes de nombres entiers peut être similaire. Il suffit d'ajouter une étape, après reproduction, pour convertir ces nombres vers la syntaxe utilisée ici pour les nombres entiers.

Cette technique de traitement des nombres négatifs peut aussi être utilisée pour les nombres à partie fractionnaire, etc. (cf. ci-dessous).

C.3.5 Traitement des nombres positionnels positifs à partie fractionnaire

Il est aisé d'adapter la méthode présentée ci-dessus pour prendre en compte les cas où des parties fractionnaires peuvent apparaître. Il faut toutefois surmonter le problème lié au fait que des caractères couramment utilisés pour séparer la partie entière de la partie fractionnaire sont aussi utilisés à d'autres fins. Le caractère séparateur est habituellement le POINT U002E ou la VIRGULE U002C. Ces caractères ont divers usages en contexte numérique.

Dans les exemples qui suivent, on suppose que seule la VIRGULE est utilisée comme séparateur décimal.

On fera comme ci-dessus, mais en comptant seulement les chiffres de la partie entière pour calculer le préfixe à deux chiffres. Le séparateur décimal (ici la VIRGULE) peut être enlevé.

Exemple :

–12,34
12,34
3,1415
3,14

Après prétraitement :

–978765 –12,34
+021234 12,34
+013.1415 3,1415
+01314 3,14

Trié :

–978765 –12,34
+01314 3,14
+0131415 3,1415
+021234 12,34

Tel que présenté à l'utilisateur :

–12,34
3,14
3,1415
12,34

C.3.6 Traitement des nombres positionnels positifs à partie fractionnaire et exposant

En présence de nombres très grands ou très petits, on utilise souvent des formats avec exposant comme $2{,}5*10^{7}$. On a ici un exposant, qui doit être combiné avec le nombre de chiffres de la partie entière (les chiffres avant le séparateur décimal) en les additionnant de façon à obtenir un exposant au nombre de chiffres fixe à être inséré comme préfixe juste avant le premier chiffre. Avec notre exemple et un exposant à trois chiffres, on obtient +00825. Il peut toutefois arriver que l'exposant soit négatif ; ce problème se traite au moyen d'un décalage constant de l'exposant. Dans le cas d'un exposant à trois chiffres un décalage de 500 peut être adéquat, ce qui donne +50825 pour notre exemple ; pour $2{,}5*10^{-7}$ on obtiendra +49425. Les valeurs négatives sont traitées comme auparavant, par complément-9 : $–2{,}5*10^{7}$ donne –49174 et $–2{,}5*10^{-7}$ donne –50574. Cette méthode devrait être familière à quiconque connaît l'arithmétique à point flottant en base 10.

Ainsi :

$2{,}5*10^{-7}$
$–2{,}5*10^{7}$

$2,5*10^{7}$
$-2,5*10^{-7}$

Après prétraitement (y compris reproduction de l'original, pour assurer le déterminisme) :

$+49425\ 2,5*10^{-7}$
$-49174\ -2,5*10^{7}$
$+50825\ 2,5*10^{7}$
$-50574\ -2,5*10^{-7}$

Trié :

$-49174\ -2.5*10^{7}$
$-50574\ -2.5*10^{-7}$
$+49425\ 2.5*10^{-7}$
$+50825\ 2.5*10^{7}$

Tel que présenté à l'utilisateur :

$-2,5*10^{7}$
$-2,5*10^{-7}$
$2,5*10^{-7}$
$2,5*10^{7}$

C.3.7 Traitement des dates et heures

Au-delà des nombres simples, les dates et heures contiennent souvent des numéraux (ainsi que des noms de mois et de jour, etc.). On souhaite souvent trier de telles dates ou heures au sein de chaînes.

Le prétraitement nécessaire pour obtenir un tri de dates ou heures par ordre temporel, en certaines syntaxes prédéterminées, est similaire à celles décrites ci-dessus :

1. Reproduire toutes les dates et heures à la fin de la chaîne, pour assurer le déterminisme dans les cas où les chaînes originales sont différentes mais représentent la même valeur temporelle. Ces copies demeurent intactes au cours des étapes suivantes.

2. Convertir toutes les dates et heures au même calendrier, si plusieurs calendriers sont utilisés et traités. Le calendrier choisi doit permettre d'exprimer l'ordre temporel. Nous utiliserons le calendrier grégorien en précisant l'année, le mois et le quantième dans nos exemples.

3. Ranger les éléments des dates et heures en ordre décroissant d'importance (jusqu'à la précision prise en compte) : année, mois, quantième, heure, minute, seconde, fraction de seconde.

4. Exprimer les heures sur une horloge de 24h, en enlevant les indications A.M. et P.M. et en ajustant l'heure au besoin.

5. Ajuster les dates et heures au fuseau horaire TUC (temps universel coordonné). Enlever les indications de fuseau horaire.

6. Remplacer les noms de mois par des nombres à deux chiffres. Ajuster aussi à deux chiffres (par bourrage avec des zéros initiaux) le quantième, les heures, les minutes et les secondes.

7. Exprimer l'année au complet, avec autant de chiffres que nécessaire. Après prétraitement, il ne doit subsister aucune ambiguïté, à savoir si par exemple « 98 » représente l'année 98 ou l'année 1998.

8. Ajouter un préfixe SIGNE PLUS aux années de l'ère courante, un SIGNE MOINS aux années antérieures. Enlever les indications comme « av. J.-C. » ou « apr. J.-C. ». (Pour être rigoureux, il faudrait aussi ajuster l'année *n* av. J.-C. à (1–*n*), négatif si *n* est positif.)

9. Insérer entre le signe et l'année un chiffre indiquant le nombre de chiffres dans l'année. Un seul chiffre devrait suffire.

10. Pour les années négatives, remplacer chaque chiffre (y compris le chiffre indiquant le nombre de chiffres dans l'année originale) par son complément-9.

11. S'assurer que le format textuel de toutes les dates est le même, jusqu'aux traits d'union, espaces, etc. (Cette uniformité s'obtient facilement en les formatant depuis une représentation interne numérique.)

12. En lieu et place, utiliser un nombre indiquant le temps sur une échelle linéaire (par exemple les heures, millisecondes ou jours depuis un point de départ prédéterminé), à la résolution requise, et traiter ce nombre comme un numéral ordinaire (cf. ci-dessus).

Pour l'étape de tri, utiliser une adaptation de la table donnée dans la présente Norme internationale, telle que le SIGNE PLUS et le SIGNE MOINS sont significatifs au même niveau que les chiffres et telle que le SIGNE MOINS a un poids inférieur au SIGNE PLUS.

Par exemple :

Date : 19 juillet 1955, à 1 p.m. TUC
Date : janvier, 20 av. J.-C.
Date : 20 sept. 1995, à 13h HNP
Date : 11-juin/345 apr. J.-C.

Après prétraitement :

Date : +41955-07-19T13:00Z 19 juillet 1955, à 1 p.m. TUC
Date : –780-01 janvier, 20 av. J.-C.
Date : +41995-09-20T10:00Z 20 sept. 1995, à 13h HNP
Date : +3345-06-11 11-juin/345 apr. J.-C.

Trié :

Date : –780-01 janvier, 20 av. J.-C.
Date : +3345-06-11 11-juin/345 apr. J.-C.
Date : +41955-07-19T13:00Z 19 juillet 1955, à 1 p.m. TUC
Date : +41995-09-20T10:00Z 20 sept. 1995, à 13h HNP

Tel que présenté à l'utilisateur :

Date : janvier, 20 av. J.-C.
Date : 11-juin/345 apr. J.-C.
Date : 19 juillet 1955, à 1 p.m. TUC
Date : 20 sept. 1995, à 13h HNP

C.3.8 Nombres moins importants que les lettres

Il faut souvent considérer des nombres précédant des lettres comme moins importants que ces lettres. Toutefois, la table-modèle commune trie les chiffres au niveau 1. Pour réduire l'importance des chiffres, on peut adapter la table de façon à ce que les chiffres soit ignorés au niveau 1 (mais pris en compte au niveau 2 ou 3) ou encore prétraiter les chaînes en déplaçant les numéraux à la fin de la chaîne ou vers un champ moins important. Lors d'un tel déplacement, il faut prendre garde à ne pas rendre identiques des chaînes différentes (et de même pour des champs), de manière à préserver le déterminisme (cf. C.3.9).

Quelques cas où il est utile de diminuer l'importance des chiffres : les adresses postales, lorsqu'un nombre précède un nom de rue ou d'îlot (comme aux États-Unis ou au Japon) ; les formules chimiques avec préfixes numériques, par ex. 1,2-dichlorobenzol.

C.3.9 Maintien du déterminisme

On a vu plus haut que dans plusieurs cas, une partie d'une chaîne est reproduite pour préserver le déterminisme, quand le prétraitement risque de rendre identiques des chaînes originalement différentes.

Cette méthode de préservation du déterminisme par reproduction peut être généralisée : si différentes parties de la chaîne doivent subir différents prétraitements, on peut simplement reproduire la chaîne originale au complet, effectuer le prétraitement (sans autre reproduction) sur l'original et enfin ajouter la copie non modifiée.

Inconvénient de cette méthode : les lettres de base de la seconde moitié de la chaîne « dédoublée » comptent plus que les accents et la casse de la première moitié. La présente Norme internationale ne prévoit pas de mécanisme pour éviter cet écueil, où la seconde moitié compterait moins que l'entièreté de la première moitié. On pourrait améliorer le processus en plaçant les deux moitiés dans des champs différents et en construisant la clé de tri en combinant les clés de chaque champ. Ce traitement n'est toutefois pas du ressort de la présente Norme internationale.

La préservation du déterminisme dans les cas où certaines des chaînes à trier sont identiques n'est pas définie par la présente Norme internationale. Une application de tri doit indiquer si elle est « stable » (si elle préserve l'ordre original de chaînes identiques) ou non. Cette indication est utile lors du tri de données séparées en plusieurs champs.

Annexe D
(informative)

Annexe didactique sur les solutions apportées par la présente Norme internationale aux problèmes de tri lexical

Pourquoi une simple comparaison caractère par caractère avec les standards de codage de caractères existants ne donnent-ils pas des résultats de tri corrects ? Que doit-on faire pour obtenir un tri correct ? Cette annexe illustre le problème au moyen d'exemples en écriture latine.

D.1 Problèmes

1. Le tri en toute langue écrite en alphabet latin (y compris en anglais) codé en, par exemple, ISO/CEI 646, ne suit pas l'ordre traditionnel des dictionnaires, qui est l'exigence minimale de l'utilisateur moyen.

   Par exemple le tri de la liste de mots « auguste », « Auguste », « contenant », « coop », « co-op », « Vice-amiral », « Vice versa » donne l'ordre suivant, si le codage ISO/CEI 646 est utilisé et le tri effectué selon l'ordre binaire :

   ```
   Auguste
   Vice versa
   Vice-amiral
   auguste
   co-op
   contenant
   coop
   ```

   Cet ordre est évidemment incorrect.

2. La transformation des majuscules en minuscules et la suppression des caractères spéciaux donne une liste triée acceptable pour les utilisateurs, mais les résultats ne sont plus déterminés par les seules chaînes triées.

   Exemple: le tri de la liste « Auguste », « auguste », « coop », « co-op » donne l'ordre suivant :

   ```
   Auguste
   auguste
   coop
   co-op
   ```

   Un tri de la même liste se présentant dans un ordre initial différent, disons « auguste », « co-op », « Auguste », « coop », peut donner se trier différemment :

   ```
   auguste
   Auguste
   co-op
   coop
   ```

3. L'ajout de caractères accentués, au moyen par exemple de n'importe quel jeu de caractères 8 bits ISO/CEI, ne fait qu'aggraver les problèmes, les causes étant les mêmes.

4. On peut penser qu'une réorganisation des tables de numéros de caractères, rendant tous les caractères apparentés voisins, pourrait simplifier le tri d'un seul caractère ; ce n'est pas le cas. Considérons la casse. Si on attribue à « a » le numéro 01, à « A » le numéro 02, à « b » le numéro 03, à « B » le numéro 04 et ainsi de suite, une liste triée directement en fonction de ces numéros de code donnera ceci :

Liste Triée	**Valeurs internes**
aaaa	01010101

```
abbb    01030303
Aaaa    02010101
Abbb    02030303
```

Ce tri est déterministe, mais évidemment incorrect pour n'importe quel pays au vu de ses usages culturels.

D.2 Solution

La seule solution à ces problèmes est de considérer les données initiales à des niveaux multiples, de façon à respecter l'ordre lexical traditionnel tout en conservant le déterminisme. Pour l'écriture latine, on doit considérer quatre niveaux (au moins) :

1. Le premier niveau traite les textes à trier sans égard à la casse, aux diacritiques (accents) et aux caractères spéciaux (qui n'ont pas d'ordre traditionnel pré-établi).

   Un exemple en anglais :

   « résumé » ('curriculum vitæ') devient « resume » ('recommencer'), sans accent.

   Un exemple en français :

   « Vice-légation » devient « vicelegation », sans accent, ni majuscule, ni trait d'union.

   Un exemple en allemand :

   « groß » devient « gross », où le eszet (ß) est converti en double s pour fins de tri.

   Certaines langues, comme l'espagnol et les langues scandinaves, ajoutent des lettres aux 26 lettres de base des alphabets anglais, français et allemand. Ces lettres supplémentaires ne sont pas triées de la manière habituelle aux autres langues, ce qui démontre le besoin d'adaptation.

2. Le deuxième niveau résout les égalités entre quasi-homographes qui ne diffèrent que par des diacritiques.

   En anglais, « resumé » et « résumé » sont des quasi-homographes. L'ordre lexical traditionnel en anglais exige que « resume » précède « résumé » (ce que le seul tri au premier niveau ne peut garantir). Dans ce cas, la tradition ne précise pas si « resumé » doit précéder « résumé », mais la logique semble l'indiquer : la plupart des dictionnaires anglais et allemands précisent seulement que les mots sans accents précèdent les mots avec accents. Toutefois, les dictionnaires allemands suivent généralement la norme allemande DIN 5007 aux règles plus précises.

   En français, à cause de nombreux groupes de plus de deux quasi-homographes, les dictionnaires les plus importants suivent la règle suivante : les accents ne sont généralement pas pris en compte, mais en cas d'égalité homographique, la *dernière* différence d'accent détermine l'ordre de deux mots. Les différents accents ont un ordre de priorité déterminé. Selon cette règle, « coté » suit « côte » mais précède « côté ». La règle se met en œuvre facilement avec l'adaptation « backwards » : un nombre est attribué à chaque caractère, représentant soit une lettre avec accent, soit une lettre sans accent ; ces nombres sont accumulés en ordre inverse, c'est-à-dire ajouté du début plutôt que de la fin. En d'autres mots, la chaîne est construite à partir de la fin des données originales, le traitement se faisant à reculons.

   Ainsi, pour trier selon cette règle la liste « cote », « côte », « coté », « côté », on attribuera les nombres comme suit : « **** », « **C* », « A*** », « A*C* », où « * » indique l'absence d'accent, « A » indique l'accent aigu et « C » l'accent circonflexe. Cette méthode suffit pour résoudre correctement l'égalité à ce second niveau.

3. Le troisième niveau résout les égalités entre quasi-homographes qui ne diffèrent que par la casse.

La tradition est ici bien établie dans les dictionnaires allemands, où les minuscules précèdent systématiquement les majuscules dans les homographes. En français, la tradition est mal établie puisque généralement les dictionnaires utilisent des capitales pour toutes les vedettes ; certains dictionnaires utilisent aussi des minuscules, qui suivent alors les majuscules, mais cette règle n'est pas explicite et souffre de nombreuses exceptions. L'anglais, tout comme le français, n'a pas vraiment de pratique bien établie à cet égard. Pour la table-modèle commune, il convient donc d'adopter la tradition allemande bien établie, de façon à regrouper le plus grand nombre possible de langues sans affecter les autres. On notera qu'au Danemark, les majuscules précèdent les minuscules, une règle différente mais bien établie. Il s'agit d'un deuxième cas démontrant le besoin d'adaptation du modèle utilisé dans la présente Norme internationale.

Par exemple : pour obtenir l'ordre « auguste », « Auguste », des nombres pourraient être attribués indiquant respectivement « mmmmmmm », « Mmmmmmm », où « m » signifie minuscule et « M » majuscule.

4. Le quatrième niveau résout les égalités restantes qui, en général, ne correspondent à aucune tradition forte. Il s'agit de quasi-homographes ne se distinguant que par la présence de caractères spéciaux.

   La résolution de ces égalités est nécessaire pour préserver le déterminisme, ainsi que pour permettre le tri de chaînes ne comprenant que des caractères spéciaux. Puisque toute trace de ces caractères spéciaux est éliminée aux trois premiers niveaux, le simple fait de les restaurer séquentiellement au quatrième niveau signifierait que leurs positions seraient perdues. Ces positions sont nécessaires pour résoudre les égalités restantes : deux quasi-homographes peuvent contenir un même caractère spécial en positions différentes et ainsi être strictement différents (exemple : « ab*cd » est distinct de « a*bcd » bien que partageant le même caractère spécial « * »).

   Par exemple : pour obtenir l'ordre « coop », « co-op », « coop- », des nombres pourraient être attribués de la façon suivante : « », « 3- », et « 5- » ; où « 3- » dénote un trait d'union en position 3 (relative) dans la chaîne originale, « 5- » un trait d'union en position 5 et ainsi de suite. On notera que « coop. », « co-op. », « coop- » (avec un point à la fin dans chaque cas) donne les nombres « 5. », « 3-3. » et « 5-1. » et donc l'ordre « co-op. », « coop. », « coop-. ».

Ces quatre niveaux peuvent être composés en une clé à quatre niveaux en concaténant les sous-clés de la plus importante à la moindre et en insérant la plus faible valeur possible comme délimiteur entre chaque paire de sous-clés. L'ordre résultant est alors l'ordre numérique des clés.

Si l'attribution des poids est faite correctement, on peut éliminer certains des délimiteurs entre les paires de sous-clés. Pour éliminer le délimiteur entre les sous-clés de premier et deuxième niveau, on fera en sorte que les poids de deuxième niveau soient tous inférieurs aux poids de premier niveau. On peut procéder de même pour éliminer le délimiteur entre deuxième et troisième niveaux, en choisissant des poids de troisième niveau tous inférieurs aux poids de deuxième niveau.

Des techniques de réduction de sous-clés permettent de réduire considérablement les besoins en mémoire. Puisqu'on n'exige des mises en œuvre ni utilisation de poids particuliers ni réduction, ces techniques ne sont pas du ressort de la présente Norme internationale. Il n'en reste pas moins intéressant de noter que la mise en œuvre peut être optimisée. Les optimisations connues se sont améliorées avec le temps et sont assez faciles à réaliser ; certaines méthodes sont plus efficaces que d'autres. Cette méthode de tri de chaînes est décrite, avec des tables, dans *Règles du classement alphabétique en langue française et procédure informatisée pour le tri*, Alain LaBonté, Ministère des Communications du Québec, 1988-08-19, ISBN 2-550-19046-7. Une technique de réduction de sous-clés – du domaine public – est décrite (avec plusieurs exemples) dans *Technique de réduction - Tris informatiques à quatre clés,* Alain LaBonté, Ministère des Communications du Québec, 1989-06, ISBN 2-550-19965-0. Voir aussi l'article de Rolf Gavare *Alphabetic ordering in a lexicological perspective*, Studies in Computer-Aided Lexicology, 1988, pp. 63–102, qui décrit aussi une technique de tri à plusieurs niveaux avec compression de sous-clés.

D.3 Adaptation

Pour de nombreuses langues, la table-modèle commune présentée ici devra être adaptée. L'adaptation peut s'avérer nécessaire aussi bien aux quatre niveaux de sous-clés de la table (par redéfinition de poids de caractères ou par ajout d'élément de tri à plusieurs caractères) que dans l'analyse contextuelle nécessaire à l'obtention de résultats corrects pour les utilisateurs de ces langues.

Laissons de côté l'analyse contextuelle accessoire dans ce qui suit, considérons des exemples de suites dans des dictionnaires pour deux langues dont les ordres de tri ne sont pas donnés par la table-modèle commune :

En espagnol traditionnel, où « ch » suit « cu » et « ña » suit « no » :

cuneo < cúneo < chapeo < nodo < ñaco

Tri comparatif français/anglais/allemand des mêmes chaînes :

chapeo < cuneo < cúneo < ñaco< nodo

En danois, où « a » précède « c », « cz » précède « cæ » et « cø », et « aa » est équivalent à « å », lequel suit « z », même dans les cas de prononciations distinctes :

Alzheimer < czar < cæsium < cølibat < Aachen < Aalborg < Århus

Tri comparatif français/anglais/allemand des mêmes chaînes :

Aachen < Aalborg < Alzheimer < Århus < cæsium < cølibat < czar

De même, le tri japonais entraîne une adaptation pour traiter correctement la marque de son prolongé. Pour nombre d'autres écritures, un certain degré d'adaptation sera nécessaire.

Bibliographie

En plus des références normatives, les normes et documents suivants sont pertinents à la présente Norme internationale.

- CAN/CSA Z243.230-1998 – *Conventions canadiennes minimales de localisation des logiciels, une norme nationale du Canada*.
- CAN/CSA Z243.4.1-1998 – *Norme canadienne de classement alphanumérique, une norme nationale du Canada, Association canadienne de normalisation*.
- DS 377:1980 – *Alfabetiseringsregler*, Dansk Standard.
- Gavare, Rolf, *Alphabetic ordering in a lexicological perspective, Studies in Computer-Aided Lexicology*, 1988, pp. 63–102.
- ISO/CEI 10646-1:1993/Amd.9:1997 Technologies de l'information – Jeu universel de caractères codés sur plusieurs octets (JUC) – Partie 1 : Architecture et plan multilingue de base. Amendement 9 : Identifiants pour les caractères.
- ISO/CEI 2022, *Technologies de l'information – Structure de code de caractères et techniques d'extension*.
- ISO/CEI 646, *Technologies de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'informations*.
- ISO/CEI 6937, *Technologies de l'information – Jeu de caractères graphiques codés pour la transmission de texte – Alphabet latin*.
- ISO/CEI 8859-1, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 1 : Alphabet latin nº. 1*.
- ISO/CEI 8859-2, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 2 : Alphabet latin nº 2*.
- ISO/CEI 8859-3, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 3 : Alphabet latin nº 3*.
- ISO/CEI 8859-4, *Technologies de l'information – Technologies de l'information -- Jeux de caractères graphiques codés sur un seul octet – Partie 4 : Alphabet latin nº 4*.
- ISO/CEI 8859-5, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 5 : Alphabet latin/cyrillique.*
- ISO/CEI 8859-6, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 6 : Alphabet latin/arabe*.
- ISO/CEI 8859-7, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 7 : Alphabet latin/grec*.
- ISO/CEI 8859-8, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 8 : Alphabet latin/hébreu*.
- ISO/CEI 8859-9, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 9 : Alphabet latin nº 5*.
- ISO/CEI 8859-10, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 10 : Alphabet latin nº 6*.
- ISO/CEI 8859-13, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 13 : Alphabet latin nº 7*.

- ISO/CEI 8859-14, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 14 : Alphabet latin nº 8 (celte)*.
- ISO/CEI 8859-15, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 15 : Alphabet latin nº 9*.
- ISO/CEI 9945-2, *Technologies de l'information – Interface pour la portabilité des systèmes (POSIX) – Partie 2 : Enveloppe et services*.
- ISO/CEI TR 14652:2004, *Technologies de l'information – Méthode de modélisation des conventions culturelles*.
- LaBonté, Alain, *Règles du classement alphabétique en langue française et procédure informatisée pour le tri, Ministère des Services gouvernementaux du Québec,*
  URL: http://www.msg.gouv.qc.ca/archivage/alphabet_classement.pdf.
- *LaBonté, Alain, Technique de réduction – Tris informatiques à quatre clés, Ministère des Services gouvernementaux du Québec,*
  *URL :*http://www.msg.gouv.qc.ca/archivage/technique_reduction.pdf.
- *Retskrivningsordbogen – 2ème édition 1996*, Dansk Sprognævn & Aschehoug Dansk Forlag A/S.
- Teknisk norm nr. 34, Swedish Alphanumeric Sorting, Statskontoret, 1992. (comprend l'article de Gavare en annexe.)
- The Unicode Standard, Version 5.0, The Unicode Consortium, Addison-Wesley, 2007. ISBN 0-321-48091-0.
- *Unicode Technical Report nº 10, Unicode Collation Algorithm*, The Unicode Consortium, URL :<http://www.unicode.org/unicode/reports/tr10/>.
- *Unicode Technical Report nº 15, Unicode Normalization Forms*, The Unicode Consortium, URL :<http://www.unicode.org/unicode/reports/tr15/>.