3iamatx : manuel de référence. v2.4

Synopsis.

```
3iamatx –o
```

Description.

Le programme `3iamatx` produit un fichier au format `.3ia` à partir d'une matrice exprimée comme décrit ci-après. La matrice est lue sur `stdin` et le fichier `.3ia` est produit sur `stdout`.

Auteur : J. Ducasse, juin 2005.

Options.

- o Les arbres produits sont nettoyés des singletons (nœuds non-terminaux ne possédant qu'un seul fils, lui-même de terminal). Par défaut, les singletons ne sont pas supprimés.

Exemples.

```
% 3iamatx < matrice.txt > fichier.3ia
% 3ia fichier.3ia out
```

La première commande produit le fichier `fichier.3ia` à partir de la matrice décrite dans le fichier `matrice.txt`. La seconde procède à l'analyse 3ia.

Format de la matrice en entrée.

Le flux d'entrée (matrice) a le format suivant. Il comprend :

- Une ligne de type 1.
- Puis une ligne de type 2.
- Puis une ou plusieurs lignes de type 3.

Les lignes vides sont ignorées, ainsi que celles dont le premier caractère est '#' (commentaires). Pour les autres lignes, dans la suite, l'expression « un séparateur » représente un nombre quelconque de blancs et/ou tabulations. Les séparateurs en début et fin de ligne sont ignorés.

Dans la suite, les contrôles mentionnés entraînent l'abandon immédiat du programme, à l'exception de ceux marqués d'un astérisque, qui sont signalés par un message mais n'entraînent pas l'abandon du programme.

Ligne de type 1.

Elle comprend N termes séparés par le séparateur. Chaque terme constitue le code d'un caractère. Le nombre de termes spécifie le nombre de caractères N pour le reste du fichier.

Contrôle : le programme contrôle que tous les codes sont différents.

Ligne de type 2.

Elle comprend N termes, chacun représentant le *pattern* de la hiérarchie associée à un caractère. L'ordre des *pattern* correspond à celui des codes dans la première ligne.

Chaque *pattern* est constitué d'un jeu de parenthèses incluant des spécificateurs de rang. Les spécificateurs de rang et les parenthèses peuvent être collés ou séparés par le séparateur ; deux spécificateurs successifs sont séparés par un séparateur. Les rangs sont désignés par des numéros (entiers ≥ 0) quelconques, et doivent tous être différents dans un même *pattern*. Par contre, il n'y a aucune contrainte entre les différents *pattern*.

Contrôles : le programme contrôle :

- que le nombre de *pattern* égale le nombre de caractères nommés en première ligne.
- que, dans chaque *pattern*, tous les numéros de rangs sont différents.

- que, dans chaque *pattern*, les parenthèses ouvrantes et fermantes sont bien balancées.
- que chaque *pattern* commence par une parenthèse ouvrante et se termine par une parenthèse fermante.
- * que, dans chaque *pattern*, il n'existe pas plus d'un spécificateur de rang pour chaque niveau hiérarchique. Note : contrôle non implémenté dans la version actuelle.

Note : Il n'y a pas de séparateur spécifique entre les *pattern*. Un *pattern* se termine avec la parenthèse fermante balançant la première parenthèse ouvrante du même *pattern*. Le *pattern* suivant commence au premier caractère qui suit. Ceci peut entraîner quelque confusion dans les messages d'erreur en cas de parenthésage mal balancé.

Lignes de type 3.

Il y a une ligne par taxon. Chaque ligne est constituée par le nom du taxon suivi de N spécificateurs de rang, N étant égal au nombre de caractères nommés dans la première ligne. Le nom du taxon et les spécificateurs de rangs sont séparés par un séparateur, ainsi que les spécificateurs entre eux. Le spécificateur de rang n° X indique quelle est la position du taxon dans la hiérarchie du caractère n° X, conformément au *pattern* donné sur la deuxième ligne. Le spécificateur de rang peut être remplacé par un « ? » qui indique que le taxon ne doit pas apparaître dans la hiérarchie de ce caractère.

Contrôles : le programme contrôle :
- que le nombre de spécificateurs de rang égale le nombre de caractères nommés en première ligne.
- que chaque spécificateur de rang existe dans le *pattern* du caractère correspondant.
- * que, pour chaque *pattern*, chaque spécificateur de rang est représenté par au moins un taxon.

Format du fichier en sortie.

Le fichier produit constitue un fichier au format pris en charge par le programme `3ia` renseigné au minimum, c'est-à-dire que seules y figurent les informations calculées à partir de la matrice en entrée. Ce sont :

- La section 4 : liste des taxons. Pour chacun, un code arbitraire est calculé.
- La section 5 : liste des hiérarchies. Les codes des caractères sont ceux fournis par la première ligne de la matrice en entrée. Les codes des taxons sont ceux générés à la section 4. Les parenthésages sont ceux générés par les *pattern* en entrée.

Les arbres produits sont nettoyés des niveaux inutiles :
- Les parenthésages vides « ( ) » sont supprimés ; un tel parenthésage peut soit dériver de la même forme dans le *pattern*, soit survenir à partir d'un *pattern* « (xx) » si le spécificateur de rang xx n'est représenté par aucun taxon.
- Les parenthésages multiples sont simplifiés.
- Les parenthèses encadrant un unique taxon sont supprimées. Ce nettoyage n'est effectif qu'avec l'option `-o`.

L'arbre nettoyé est équivalent à l'arbre brut.

Ainsi, à partir du *pattern* : `(((1 (2))) 3 (4))`
peut être généré l'arbre : `(((A B ())) C (D))`
si 1 → A, B ; 2 vide ; 3 → C ; 4 → D.
Cet arbre sera finalement nettoyé en : `((A B) C D)`
en appliquant les trois règles ci-dessus, avec l'option `-o`,
ou : `((A B) C (D))`
sans l'option `-o`.

Différences de contraintes entre `3ia` et `3iamatx`.

Il existe quelques différences entre les contraintes que les deux programmes imposent. Celles-ci ne posent par de problème en général, mais peuvent entraîner l'impossibilité de traiter le fichier par `3ia` dans certains cas extrêmes. Ces différences devraient être corrigées dans les versions futures. Elles concernent :

- La longueur maximale des codes de caractère.
- La syntaxe des codes de caractère : alphanumérique pour `3iamatx`, numérique seulement pour `3ia`.

- Les noms des taxons peuvent comprendre des blancs pour `3ia` ; ils doivent former un seul mot, sans blanc, pour `3iamatx`.

Historique.

version 2.4 : décembre 2009
Changement de nom → `3iamatx`.

version 2.3 : mars 2007
Introduction de l'option `-o`. Le fonctionnement avec l'option `-o` correspond au fonctionnement auparavant. Le fonctionnement par défaut est nouveau.

version 2.2 : septembre 2006
Nettoyage des arbres produits, suppression des niveaux hiérarchiques inutiles.
Introduction des commentaires dans le fichier d'entrée.

version 2 : janvier 2006
Abandon du format binaire en entrée et traitement du nouveau format généralisé avec *pattern*.

version 1 : juin 2005