3ia : manuel de référence. v3.7

Synopsis.

```
3ia -0234 [-t nb1[,nb2]] [-o maxc,minic,bb] [-o a1,a2] [-o txb,txt] [-o h1]
    -C -v -V -I filein fileout
```

Description.

Procède à une analyse cladistique basée sur la méthode de l'analyse à trois éléments. Le principe de compatibilité est utilisé. L'ensemble de l'univers à traiter est donné dans le fichier d'entrée *filein*.

3ia utilise un fichier de configuration tts.cfg, permettant de redéfinir certains paramètres. Ce fichier est facultatif, tous les paramètres ayant une valeur par défaut. Noter aussi que les options de la ligne de commandes peuvent aussi être fournies dans ce fichier, sous la clef TTS_OPT.

3ia réalise les tranches de travaux suivantes. Le contenu de chaque tranche est détaillé plus bas :

0 - Lecture du fichier d'entrée et calcul des 3is et de leur pondération fractionnaire (PF). Cette tranche est obligatoire.
1 - (obsolète).
2 - Calcul des arbres de compatibilité.
3 - Calcul de l'arbre d'intersection par l'utilitaire 3iapr.
4 - Calcul des RI (Indices de Rétention) par caractère pour tous les arbres de compatibilité et pour l'arbre d'intersection.

Par défaut, les tranches sont toutes effectuées. On peut restreindre le traitement a certaines d'entre elles avec les options -234.

Les résultats d'analyse sont produits dans plusieurs fichiers de sortie de racine *fileout*, différant par l'extension :

fileout.3iz Description détaillée de l'analyse, étape par étape.
fileout.3ri Matrice des RI par caractère (tranche 4 seulement). Les RI pour l'arbre d'intersection n'y figurent que si la tranche 3 a été effectuée.

Note : on peut donner *fileout* avec ou sans l'extension .3iz. Si elle est présente, elle sera remplacée par .3ri pour l'autre fichier.

Auteur : J. Ducasse, juin 2005.

Options.

0234 Effectue seulement les tranches spécifiées. Il existe des dépendances entre les tranches :
- L'option -3 est ignorée si l'option -2 n'est pas fournie.
- *Idem* pour l'option -4.
- L'option -0 est implicite dès qu'une autre tranche est spécifiée.

Par défaut, toutes les tranches sont réalisées (équivalent a -234).

t L'argument est formé de deux entiers séparés par une virgule, le second étant facultatif :

-t *n1*[,*n2*]

n1 spécifie le nombre de *thread* utilisés pour le traitement de la tranche 2. Par défaut, un seul *thread* est utilisé. *n2* spécifie la profondeur maximale (nombre de UTOs introduites) à laquelle un nouveau *thread* peut être créé (5 par défaut). La valeur 0 signifie l'absence de limitation ; la valeur 1 est équivalente au mode *monothread*. Voir plus bas le paragraphe consacré à la description détaillée de la tranche 2. Option ignorée avec l'option -oa1.

- `o` Spécifie un certain nombre d'alternatives concernant l'algorithme utilisé pour la tranche 2. Dans chaque groupe, les modes sont exclusifs.
  `maxc/minic/bb`
  - `maxc` Algorithme de maximisation des compatibilités par exploration exhaustive des arbres.
  - `minic` Algorithme de minimisation des incompatibilités par exploration exhaustive des arbres.
  - `bb` Algorithme de minimisation des incompatibilités avec technique du *branch and bound*.

  `a1/a2`
  - `a1` Technique intuitive d'exploration des arbres.
  - `a2` Technique optimisée d'exploration des arbres.

  `txb/txt`
  - `txb` Lors de l'exploration de l'arbre des arbres, ajoute les UTOs dans l'ordre où elles figurent dans le fichier *`filein`*.
  - `txt` Trie au préalable les UTOs pour placer en premier celles susceptibles d'amener le meilleur élagage.

  `h1` Optimisation par tri des sous-arbres. Cette option n'est significative qu'avec le mode *branch and bound*.

  Les différents choix peuvent être fournis chacun derrière une option `-o`, ou regroupés en une liste et séparés par une virgule. Exemple : `"-obb -otxb"` ou `"-obb,txb"`.
  Les résultats sont identiques quelles que soient les options choisies ; seul le temps de calcul varie. Par défaut, la configuration `"-obb,a2,txt"`, *a priori* la plus efficace, est utilisée.
- `C` A la tranche 3, l'utilitaire `3iapr` est lancé avec l'option `-C`. Par défaut, `3iapr` est lancé sans cette option et calcule donc aussi la liste des arbres compatibles.
- `v` Produit des informations supplémentaires dans *`fileout`*`.3iz`. Dans la version actuelle, cette option a pour effet de générer la section 3 du fichier, qui sans cela n'est pas produite.
- `V` A propos. Peut être utilisée seule, sans autre argument ; dans ce cas, le programme se termine immédiatement après avoir affiché l'à propos.
- `I` L'entrée `TTS_OPT` du fichier de configuration est ignorée. Seules les options données sur la ligne de commande sont prises en compte.

Exemple.

`% 3ia data out`

Réalise l'analyse 3ia complète sur l'univers décrit dans le fichier `data`, avec les options optimales implicites `"-obb,a2,txt"`, et produit deux fichiers : `out.3iz` détaillant l'analyse et `out.3ri` contenant la matrice des RI par caractère.

Clefs de configuration.

Le programme utilise le fichier de configuration `tts.cfg`. Ce fichier est recherché en priorité dans le répertoire courant ou, à défaut, dans le répertoire référencé par la variable d'environnement TTS_PKG. Les entrées suivantes sont utilisées :

- `MAXFPF` Spécifie la valeur maximale des PF flottantes. Autrement dit, les PF flottantes sont calculées de façon que la plus élevée ait la valeur `MAXFPF`. Par défaut, `MAXFPF`=10000.
- `MAXIPF` Si l'une des PF vraies dépasse la valeur `MAXIPF`, les PF flottantes sont utilisées. Si `MAXIPF` vaut 0, il n'y a pas de telle limite. Par défaut, `MAXIPF`=0.
- `SED_PROG` Exécutable `sed` ou équivalent utilisé pour filtrer les fichiers modèles. Par défaut, `SED_PROG="sed"`. Le programme sera lancée avec les conventions habituelles : on peut donc soit spécifier son chemin complet, soit donner seulement son nom et laisser le système utiliser la variable d'environnement `PATH`. Note : sur Windows, si `sed` (utilitaire Unix) n'est pas disponible, on pourra utiliser l'utilitaire `minised` livré.
- `DIR_FILE` Répertoire contenant les filtres `sed` et les modèles.

FORMAT — La valeur est une série de mots clefs spécifiant des options de format en sortie. Les valeurs suivantes sont acceptées :

- p2/p4 Format des 3is : avec p2, les 3is seront produits au format "a(b c)" ; avec p4, ils seront produits au format "(a(b c))". Par défaut : p2.
- vg/bl Format des arbres produits : les nœuds frères sont séparés par une virgule si vg est spécifié, un blanc si bl est spécifié. A défaut, le mode est celui du premier arbre trouvé dans le fichier en entrée.

TTS_OPT — La valeur a la même syntaxe que la partie « options » de la ligne de commande, et fournit des options qui seront prises en compte par le programme comme si elles avaient été fournies sur la ligne de commande. Les options fournies sur la ligne de commande sont cependant prioritaires sur celles fournies sous TTS_OPT. Ainsi, avec la clef :

```
TTS_OPT   -t4 -omaxc
```

la commande :

```
3ia -t2 filein fileout
```

sera équivalente à :

```
3ia -t2 -omaxc filein fileout
```

Pour les options de tranche -0234, celles fournies sous TTS_OPT sont ignorées dès que l'une d'entre elles est fournie sur la ligne de commande. La clef TTS_OPT permet de fournir des options toujours utilisées, tout en permettant de les ignorer pour une exécution particulière en les écrasant par la ligne de commande.
Cette entrée est ignorée avec l'option -I.

RI_DEC — Nombre de décimales utilisées pour afficher la valeur de RI. Par défaut : 3.

PROGRES — La valeur est un couple permettant de paramétrer l'affichage du pourcentage de progression de la recherche des arbres de compatibilité. La première valeur (0 – 9 ; 3 par défaut) est le nombre de décimales affichées du pourcentage. La seconde est le nombre minimal d'arbres traités donnant lieu à la mise à jour du pourcentage ; le nombre est donné en milliers (100 par défaut, représentant 100 000 arbres). Le nombre de décimales affichées peut être automatiquement réduit selon le rapport entre ce nombre et la profondeur de l'arbre des arbres exploré.
Le nombre de décimales n'influe pas seulement sur la précision de la progression affichée. Il peut aussi avoir un impact sur la durée du traitement, notamment en traitement *multithread*.

Le fichier d'entrée *filein*.

Le fichier ce compose de 5 sections, chacune terminée par un *";"* :

1- Le titre de la matrice.
2- Les dimensions : nombre de UTOs et nombre de caractères.
3- Le référentiel : le non du caractère précédé de sont numéro entre crochets, la liste de ses différents états et l'arbre du caractère avec la correspondance aux états. Exemple :

```
[3] Membre
    1 absence de membres
    2 pattes
    3 ailes
(1,(2,(3)))
```

Cette notation signifie que dans l'espace de discussion (1), certains organismes possèdent des pattes (2), elles-mêmes parfois différenciées en ailes (3).
4- Les UTOs : liste des UTOs et leur code associé.
5- Description : arbre élémentaire pour chaque caractère.

Exemple de fichier d'entrée :

```
Betes a plumes, a poils et a ailes ;
Dimensions
 ntax=4 ncar=3;
```

```
Referentiel
 [1] Ailes
     1 absente
     2 présente
     (1,(2))
 [2] Pattes
     1 non
     2 deux
     3 quatre
     (1,(2,(3)))
 [3] Corne
     1 absente
     2 présente
     (1,(2))
 ;
Taxons
 A  = Licorne
 B  = Yeti
 C  = L'Eole
 DD = Serpent a plume
 E  = Griffon
 FF = Pegase (incertae sedis)
 G  = Ange
;
Descriptions
 [1] (A B (C E FF G) DD)
 [2] (C DD (B E G (A FF)))
 [3] ((A) B C DD E FF G)
;
```

Ci-dessous sont données des précisions sur le format de ce fichier tel qu'il est entendu dans ce programme.

Définitions :

- blanc : regroupe les caractères blanc et tabulation.
- ligne blanche : formée de 0 ou plusieurs caractères blancs seulement.

Format général :

- Les lignes blanches sont ignorées.
- Les lignes commençant par ″#″ sont des commentaires et sont ignorées. Le ″#″ doit être le premier caractère de la ligne.
- Le ″;″ terminant chaque section peut être sur une ligne propre, ou bien terminer la dernière ligne utile de la section. La ligne doit être blanche derrière le ″;″.
- Les sections peuvent se présenter dans n'importe quel ordre, à l'exception du titre qui doit être le premier.

- Chaque section (sauf le titre) doit commencer par son titre propre, *case sensitive*, éventuellement précédé ou suivi d'un nombre quelconque de blancs : `Dimensions`, `Referentiel`, `Taxons`, `Descriptions`.
- Dans la version actuelle de `3ia`, les sections `Dimensions` et `Referentiel` sont inutilisées et facultatives.

Pour la section `Taxons` :

- Chaque ligne est formée de trois termes : le code, le ″=″, le nom. Chaque terme peut être précédé ou suivi d'un nombre quelconque de blancs.
- Le code est formé de 1 à 23 caractères (les caractères surnuméraires sont ignorés) quelconques, sauf le blanc.

- Le nom commence au 1er caractère non blanc suivant le "=", et se termine au dernier caractère non blanc précédant la fin de ligne (ou le ";"de fin de section). Le nom est formé d'un nombre quelconque de caractères quelconques.
- 3ia contrôle qu'un même code n'est pas utilisé deux fois.

Pour la section Descriptions :

- Chaque ligne est formée de 4 termes : le "[" , le numéro de caractère, le "]", l'arbre. Chaque terme peut être précédé ou suivi d'un nombre quelconque de blancs.
- Le numéro de caractère est un entier, sur 23 chiffres maximum.
- L'arbre commence au 1er caractère non blanc suivant le "]", et se termine au dernier caractère non blanc précédent la fin de ligne (ou le ";" de fin de section).
- L'arbre est formé de parenthèses ouvrantes et fermantes, et de codes.
- Pour les séparateurs, deux syntaxes peuvent être utilisées :
  - Syntaxe sans séparateur spécifique. Deux codes successifs doivent être séparés par au moins un blanc.
  - Syntaxe avec le séparateur virgule ",". Les éléments frères, qu'il s'agisse d'une simple UTO ou d'un sous-arbre encadré par des parenthèses, doivent être séparés par une virgule.

  Dans les deux cas, il peut y avoir un nombre quelconque de blancs entre les différents termes : UTOs, parenthèses ou virgules.

  L'arbre donné pour un caractère doit être entièrement dans une syntaxe ou dans l'autre. Par contre, les différents caractères peuvent adopter des syntaxes différentes.
- L'arbre commence par une "(" et se termine par une ")" ; les parenthèses doivent être correctement balancées. La ligne doit être blanche après la ")" balançant la "(" initiale (à l'exception du ";" de fin de section éventuel).
- Les noeuds vide "()" sont interdits.
- 3ia ne contrôle pas qu'un même numéro de caractère n'est pas utilisé deux fois.
- Comme la section Referentiel est ignorée, 3ia ne contrôle pas que les caractères utilisés sont référencés dans cette section.
- Pour l'arbre, 3ia contrôle que les codes utilisés ont été déclarés dans la section Taxons.
- 3ia contrôle qu'une même UTO (code) ne figure pas deux fois dans un même arbre.
- Les UTOs peuvent intervenir dans n'importe quel ordre ; des UTOs déclarées dans la section Taxons peuvent manquer dans certains arbres.

Le fichier de sortie *fileout*.3iz.

Le fichier de sortie est composé de sections introduites par l'en-tête "-<D*nn*>--------------- *Titre* : " et terminées par "-<F*nn*>-". Les sections sont présentes ou non selon les tranches de travaux effectuées.

Sections toujours présentes.

Section 1 "Configuration" :

Reproduit les paramétrages extraits du fichier de configuration, ainsi que les modalités de l'option -o activées (explicitement ou implicitement).

Section 2 "Entree" :

On y trouve des informations extraites de l'analyse du fichier d'entrée :

- Le titre de la matrice (première section du fichier d'entrée).
- Le nombre de UTOs définies dans la section 4 du fichier.
- Le nombre de 3is différents calculé à partir des arbres décrits dans la section 5 (le même que le nombre affiché à la section "3is" plus bas).
- La somme des PF de ces 3is.

Ensuite est donnée la liste des UTOs prises en compte et leur code.

Section 3 "Caracteres" :

Cette section présente l'analyse des arbres décrits dans la section 5 du fichier d'entrée : représentation indentée de l'arbre, paramètres calculés. Elle n'est produite que si l'option -v est fournie.

Pour chaque caractère, introduit par le terme "caractere", on trouve :
- Son numéro.
- L'arbre, repris textuellement du fichier d'entrée, entre chevrons.
- Le nombre de nœuds non terminaux (ou nœuds internes, noeuds), et le nombre de nœuds terminaux (feuilles).

Puis vient ensuite la représentation indentée de l'arbre. Chaque nœud est représenté par un X (non terminal) ou un F (terminal). L'indentation représente la filiation : les nœuds frères ont le même degré d'indentation, le fils est indenté de deux caractères par rapport à son père.

Chaque nœud terminal (F) est suivi du code de l'UTO associée. Pour chaque nœud non terminal (X), on trouve :
- La liste des UTOs absentes, puis celle des UTOs présentes après le séparateur "/". L'ensemble de la liste est entre chevrons.
- Le nombre total de 3is associés à ce nœud.
- Le nombre de 3is indépendants.
- Sa pondération fractionnaire exprimée sous forme fractionnaire.
- La liste des 3is associés à ce nœud. Chaque 3is est sous la forme "(ABS(PRES1 PRES2))", où ABS est la UTO absente et PRES1 et PRES2 les deux UTOs présentes. La liste est présentée sur quatre colonnes. Seuls les 3is non redondants sont présentés (c'est-à-dire ceux non générés par un nœud descendant) : ce sont eux qui sont pris en compte dans le calcul de la pondération fractionnaire.

Section 4 "3is" :

Cette section donne la liste exhaustive non redondante de tous les 3is de l'univers traité et leur pondération fractionnaire.

Derrière le titre 3is, on trouve :
- Le nombre de 3is de la liste.
- Une indication sur la représentation des pondérations fractionnaires :
  PF exactes : les PF ont été calculées sur les valeurs entières, donc exactes.
  PF flottantes : les PF ont été calculées en virgule flottante. Entre parenthèses figure la cause de ce choix :
    debordement : valeurs entières générées par la réduction des fractions trop grandes.
    limite=*max* : limite spécifiée par la clef de configuration MAXIPF dépassée. Cette limite est rappelée par *max*.

Vient ensuite la liste de tous les 3is. Pour chacun d'eux, on trouve :
- Le 3is sous la forme "(ABS(PRES1 PRES2))" ou "ABS(PRES1 PRES2)" selon la clef de configuration FORMAT.
- Le nombre de nœuds non terminaux (noeuds) auquel il appartient.
- Sa pondération fractionnaire prise en compte dans les calculs (PF).
  - Si la représentation (voir ci-dessus) est "PF exactes", il s'agit de la pondération fractionnaire entière, et elle est suivie de son expression fractionnaire avant réduction puis de la pondération flottante (fPF).
  - Sinon, il s'agit de la pondération fractionnaire flottante (fPF).

Sections présentes si la tranche 2 a été exécutée.

Section 5 "Reorganisation des taxons" :

Cette section n'est présente que si l'option -otxt était activée. Elle donne l'ordre de prise en compte des UTOs dans la génération de l'arbre des arbres.

Si l'option -otxb était activée, cette section manque et les UTOs sont prises dans l'ordre de leur déclaration dans la section 4 du fichier d'entrée.

Section 6 "Recherche des arbres compatibles" :

On trouve d'abord l'indication de l'algorithme utilisé :

- Maximisation des 3is compatibles (option -omaxc), avec la somme maximale des PF des 3is compatibles. Les arbres retenus sont ceux pour lesquels la somme des PF des 3is compatibles égale cette valeur. Suit la valeur du RI = sPFc_max / SS (SS = somme des PF des 3is de l'univers), affiché avec le nombre de décimales fourni par l'entrée RI_DEC du fichier de configuration.
- Minimisation des 3is incompatibles (option -ominic ou -obb), avec la somme maximale des PF des 3is compatibles (calculée comme la différence entre, d'une part la somme des PF de tous les 3is, et d'autre part la somme minimale des PF des 3is incompatibles, valeur effectivement considérée dans le traitement). Les arbres retenus sont ceux pour lesquels la somme des PF des 3is incompatibles égale cette valeur. Suit la valeur du RI = (SS - sPFi_min) / SS (SS = somme des PF des 3is de l'univers).

La durée du traitement est ensuite affichée derrière le signe =>.

Si l'option *multithread* était active, on trouve ensuite une table statistique donnant l'activité des *thread* à chaque niveau de l'arbre des arbres. Pour chaque niveau, on trouve :

- Le rang (rang), c'est-à-dire le nombre de UTOs.
- Le nombre (nb) de *thread* créés pour initialiser le traitement d'un sous-arbre ayant pour racine un arbre de nb UTOs.
- La durée moyenne d'existence de ces *thread* (duree), au format hh:mm:ss.

On trouve ensuite l'option d'élagage utilisée :

- Exploration exhaustive (pas d'élagage) (option -omaxc ou -ominic).
- Exploration *branch and bound* (option -obb). Dans ce cas, la liste qui suit donne le nombre de sous-arbres élagués par l'algorithme du *branch and bound* à chaque niveau de l'arbre des arbres (c'est-à-dire le nombre de UTOs prises en comptes). Pour chaque niveau, on trouve :
  - Le rang (rang), c'est-à-dire le nombre de UTOs prises en compte dans le nœud racine du sous-arbre élagué.
  - Le nombre (nb) de sous-arbres élagués à ce niveau.
  - La UTO correspondant au niveau (taxon).

La liste des arbres retenus est donnée ensuite, en notation parenthésée. La notation avec le séparateur virgule est utilisée si au moins un arbre utilisait cette notation dans le fichier d'entrée, la notation sans séparateur sinon. Le nombre d'arbres est indiqué en tête. Chaque arbre est précédé d'un numéro, qui fera notamment la correspondance avec la matrice des RI par caractères.

Sections présentes si la tranche 3 a été exécutée.

Section 7 "Reconstruction de l'arbre d'intersection par 3iapr" :

La sortie du programme 3iapr est recopiée ici. On y trouve :

- La liste A des 3is communs à tous les arbres retenus présentés ci-dessus.
- La liste B des 3is communs entre la liste A et la liste des 3is de base présentés à la section « 3is ».

- La liste des arbres reconstruits avec tous les UTOs présentes dans les 3is de la liste B et incluant ces mêmes 3is. Le nombre qui précède chaque arbre donne le nombre de 3is que génère l'arbre en plus de ceux de la liste B. La liste est classée par ordre croissant de ce nombre.
- Enfin est donné l'arbre d'intersection reconstruit.

Voir le manuel de `3iapr` pour plus d'information sur les informations figurant dans cette section.

Le fichier de sortie *`fileout`*`.3ri`.

La matrice présente dans ce fichier a la forme suivante :

- Chaque ligne correspond à un arbre compatible donné à la section « Recherche des arbres compatibles » du fichier *`fileout`*`.3iz`. Dans la première colonne, chaque arbre est identifié par son numéro tel qu'il figure dans cette section.
- Chaque colonne correspond à un caractère défini à la section 5 « Description » du fichier d'entrée *`filein`* (ceci est rappelé par la flèche `"car ->"`). Sur la première ligne, chaque caractère est identifié par son code tel qu'il figure dans cette section.
- A chaque intersection, on trouve le RI de cet arbre pour ce caractère. Les RI sont des rapports (<= 1) correspondant à la part de l'arbre dans la pondération fractionnaire du caractère ; la valeur affichée dans la matrice est la partie entière de RI×10^*N*, *N* étant le nombre de décimales donné sous l'entrée RI_DEC du fichier de configuration. Le RI est remplacé par `"-"` pour un caractère ne générant aucun 3is.

L'ouverture de fichier dans MS Excel en facilite la lecture.

Le fichier de configuration `tts.cfg`.

`3ia`, ainsi que les autres utilitaires du paquetage, lit au cours de son initialisation un fichier de configuration `tts.cfg`. Ce fichier a la syntaxe standard des fichiers de configuration DCFG, rappelée succinctement ci-après :

- Il est formé de lignes « clef valeur ».
- Les clefs sont prédéfinies par le programme, et leur signification est détaillée dans le manuel de chacun. La valeur est spécifiée par l'utilisateur.
- Les lignes commençant par ″#″ sont des commentaires qui sont ignorés. De même, les lignes vides ou blanches sont ignorées.
- Les termes de la forme $XXX sont substitués par la valeur de la variable d'environnement XXX. Le nom peut être délimité à droite par un ″]″ si nécessaire.

Détail des opérations réalisées et des valeurs calculées.

Tranche 0 : chargement de l'univers.

Les données suivantes sont mises en place à partir du contenu du fichier *`filein`* :

- La liste des UTOs définies dans la section 4 du fichier.
- La liste des caractères utilisés dans la section 5 du fichier. A chacun est associé un arbre.
- Les arbres spécifiés à la section 5, chacun associé à un caractère. Chaque arbre est constitué :
  - de nœuds terminaux, chacun associé à une UTO.
  - de nœuds non terminaux, chacun racine d'un sous-arbre.

A partir de ces données de base, constituant l'univers de travail, sont calculées pour chaque arbre (chaque caractère) les données suivantes :

- La liste des UTOs incluses dans l'arbre (`tt` = leur nombre). Toutes les UTOs de l'univers ne sont pas forcément présentes pour tous les caractères.
- Pour chaque nœud non terminal `NnT` :

- La liste des UTOs incluses dans le sous-arbre descendant de ce nœud, dites UTOs « présentes » (pp = leur nombre). Les autres UTOs de l'arbre sont dites « absentes » (aa = tt – pp = leur nombre).
- La liste des 3is associés à ce nœud. Les 3is sont construits en prenant toutes les combinaisons entre 1 « absent » et 2 « présents ». Chaque 3is comprend donc une UTO ABS et deux UTOs PRES (PRES1 et PRES2).
- Le nombre de 3is de ce nœud = (aa × pp × (pp – 1)) / 2. On voit qu'un nœud ne comprend aucun 3is s'il n'a aucun « absent » ou aucun « présent » ou un seul « présent ».
- Le nombre de 3is indépendants = Itts = (pp – 1) × aa.
- Pour le nœud NnT, chaque 3is est marqué comme « redondant » ou « non redondant ». Un 3is est « redondant » si ce même 3is existe aussi dans l'un des nœuds non terminaux du sous-arbre descendant de NnT ; il est « non redondant » sinon. Dans la suite, seuls les 3is « non redondants » sont considérés (Ntts = nombre de 3is « non redondants »). Par construction, chaque 3is n'est « non redondant » que pour un seul nœud non terminal de l'arbre.
- On calcule le rapport nPF = Itts / Ntts = pondération fractionnaire du nœud.

L'analyse des arbres des caractères, avec les valeurs de Itts et de nPF pour leurs nœuds non terminaux, ainsi que la liste des 3is qu'ils génèrent, est produite dans la section 3 du fichier de sortie.

L'ensemble des 3is calculés pour tous les nœuds de tous les arbres de l'univers constitue la liste exhaustive non redondante de tous les 3is de l'univers. Chaque 3is a été généré par un ou plusieurs caractères ; pour chacun de ces caractères, il a été généré (en tant que « non redondant ») par un seul nœud non terminal : à chaque 3is est donc associée une liste de nœuds non terminaux, dont on a calculé ci-dessus pour chacun la pondération fractionnaire nPF.

Pour chaque 3is, on calcule sa pondération fractionnaire, somme de toutes les pondérations fractionnaires nPF des nœuds de cette liste. La valeur absolue de cette somme n'a pas de signification ; ce qui est significatif, c'est la valeur relative des PF entre les 3is. Un processus de renormalisation est appliqué, qui permet d'obtenir une valeur entière de pondération fractionnaire ttsPF comparables entres les différents 3is.

Si, à la suite de cette renormalisation, la PF d'au moins un 3is est égale à zéro, un message d'avertissement est affiché à l'écran. Ceci peut se produire avec un très grand nombre de caractères pour les 3is générés par quelques uns d'entre eux seulement. Pour éviter cela, il est conseillé d'augmenter la valeur maximale de renormalisation MAXFP, qui doit être du même ordre de grandeur que le nombre de caractères.

La liste des 3is, avec leur ttsPF, est produite dans la section 4 du fichier de sortie.

Note : dans le programme, la valeur ttsPF est calculée ainsi. La somme des nPF est calculée de deux manières :

- Calcul flottant : somme de toutes les nPF exprimées en flottant. Les valeurs obtenues pour tous les 3is sont ensuite renormalisées de façon à ce que la PF maximale égale la valeur MAXFPF spécifiée par l'utilisateur dans le fichier de configuration. La pondération fractionnaire du 3is est la partie entière du résultat : fPF.
- Calcul fractionnaire : somme fractionnaire de toutes les nPF exprimées sous forme fractionnaire, après réduction au dénominateur commun. Les valeurs obtenues pour tous les 3is sont ensuite renormalisées en les réduisant à leur dénominateur commun. La pondération fractionnaire du 3is est le numérateur après renormalisation : iPF.

Le calcul fractionnaire présente l'avantage de conserver des valeurs exactes dans les calculs, car ceux-ci sont faits en entiers ; les calculs flottants, eux, sont toujours sujets à arrondis. Cependant, le calcul fractionnaire peut devenir impossible si les réductions à un dénominateur commun font apparaître de trop grands nombres. La validité du calcul fractionnaire est déterminée comme suit :

- Débordement informatique : les entiers trop grands ne peuvent être représentés.
- Comparaison avec la limite MAXIPF fournie par l'utilisateur dans le fichier de configuration : si une des valeurs iPF dépasse cette limite, le calcul fractionnaire est invalidé.

La valeur ttsPF retenue pour la suite des calculs est :

- iPF si le calcul fractionnaire est valide.
- fPF s'il a été invalidé.

Exemples :

- Un 3is donné est représenté dans trois nœuds, dont les nPF valent respectivement 1/13, 1/6 et 1/3. Pour ce 3is, la somme flottante des nPF égale 0,577, la somme fractionnaire égale 45/78.
- Si l'univers comprend trois 3is, dont les sommes des nPF égalent respectivement 15/46 – 45/78 – 8/55, soit en notation flottante : 0,326 – 0,577 – 0,145 :
  - Si MAXFPF égale 1000, les fPF valent 565 – 1000 – 251 après renormalisation, c'est-à-dire multiplication par le facteur 1733 (= 1000/0,577).
  - Les iPF des trois 3is égalent, après renormalisation, 32175 – 56925 – 14352, suite à la réduction au dénominateur commun 98670.

  Si MAXIPF n'est pas fourni, ou égale par exemple 100000, les iPF seront validés. Si MAXIPF égale 50000, les iPF sont abandonnés car l'un d'entre eux dépasse cette limite.

Optimisation du chargement.

Le chargement des caractères fournis dans le fichier d'entrée est instantané si celui-ci contient un nombre raisonnable de caractères, mais s'allonge lorsque ce nombre atteint quelques millions. D'autre part, l'espace mémoire nécessaire pour stocker les informations utiles peut devenir prohibitif.

En fait, cette consommation de ressources n'est nécessaire que si la matrice des RI doit être produite en fin de traitement dans le fichier *fileout*.3ri. Si ce fichier n'est pas attendu, il est conseillé d'utiliser 3ia sans l'option -4, ce qui aura pour effet de diminuer drastiquement le temps de chargement et l'espace mémoire consommé.

Tranche 2 : calcul des arbres de compatibilité.

Les arbres mis en place dans la tranche 0, ainsi que les 3is qui en ont été déduits, seront appelés dans la suite arbres de base et 3is de base, respectivement, pour les distinguer des arbres dichotomiques explorés dans cette tranche 2 et de leurs 3is associés.

Dans la tranche 2, le programme calcule tous les arbres dichotomiques (c'est-à-dire où chaque nœud, terminal ou non, a un et un seul frère) possibles incluant toutes les UTOs de l'univers. Pour N UTOs, le nombre NAB de tels arbres dichotomiques égale :

```
NAB = (fact(2 × N)) / (2^(N – 1) × fact(N – 1))
```

ce qui est égal à :

```
NAB = 1 × 3 × 5 × 7 × … × (2×N–3)
```

(fact() est la fonction factorielle, ^ est l'élévation à la puissance).

Pour chaque arbre est calculée la liste de ses 3is. Dans la liste des 3is de base, chaque 3is est alors repéré comme présent ou absent de la liste des 3is de l'arbre dichotomique :

- Les 3is de base présents dans l'arbre dichotomique sont dits compatibles.
- Les 3is de base absents de l'arbre dichotomique sont dits incompatibles.

Noter que les 3is de l'arbre dichotomique absents de la liste des 3is de base sont simplement ignorés.

On calcule alors :

- La somme sPFc des pondérations fractionnaires ttsPF des 3is compatibles.
- La somme sPFi des pondérations fractionnaires ttsPF des 3is incompatibles.

Parmi les NAB arbres possibles, seuls sont retenus ceux correspondant au critère de sélection, à savoir :

- Dans la méthode de maximisation des compatibilités (-omaxc), on calcule la valeur maximale des sPFc de tous les arbres (max_sPFc), et on ne conserve que les arbres dont la sPFc égale cette valeur maximale.

- Dans la méthode de minimisation des incompatibilités (`-ominic` ou `-obb`), on calcule la valeur minimale des `sPFi` de tous les arbres (`min_sPFi`), et on ne conserve que les arbres dont la `sPFi` égale cette valeur minimale (éventuellement zéro).

La liste des arbres retenus est produite dans la section 6 du fichier de sortie.

Optimisation.

La tranche 2 pouvant être très longue, plusieurs méthodes d'optimisation sont proposées.

branch and bound
A faire.

Tri des UTOs.
A faire.

Tri des sous-arbres (option `-oh1`).
Dans cette méthode, lors de l'exploration de l'arbre des arbres, pour un nœud donné, parmi tous les sous-arbres qui en dérivent par ajout d'une UTO à toutes les positions possibles, on explore en priorité ceux dont la `sPFi` (partielle) de la racine est la plus faible. On estime qu'il y a ainsi plus de chance pour que ces sous-arbres génèrent des arbres terminaux à faible `sPFi`. Ce gain, hypothétique, est à mettre en balance avec le coût certain du tri nécessaire. Cette optimisation n'a de signification qu'avec la méthode *branch and bound*.

Implémentation de la recherche *multithread*.

La recherche des arbres dichotomiques compatibles revient à l'exploration de l'arbre de toutes les combinaisons possibles. Il est possible de paralléliser l'exploration des différents sous-arbres grâce à l'option `-t` : le nombre de *thread* indiqué représente le nombre de traitements parallèles qui tourneront simultanément. La parallélisation est destinée à réduire le temps de calcul par les deux mécanismes suivants :

- D'une part, sur une machine multi-processeurs, elle permet d'exploiter toutes les ressources disponibles. Noter que le nombre de *thread* peut être supérieur au nombre de processeurs : le parallélisme est alors implémenté par le principe du temps partagé.
- D'autre part, si l'algorithme de *branch and bound* est utilisé (option `-obb`), en explorant simultanément plusieurs branches de l'arbre des arbres, elle permettra peut-être de trouver plus rapidement des valeurs de `sPFi` qui permettront d'élaguer plus rapidement certaines branches.

Le premier mécanisme est certain et immédiatement observable ; le second est hypothétique. Le nombre de *thread* optimal doit être déterminé par l'expérience, en tenant compte du fait qu'un trop grand nombre de *thread* (fonctionnant en temps partagé) diminue les performances, d'une part à cause de la charge système de bascule de contexte, d'autre part à cause des attentes de synchronisation nécessaires entre eux.

Par ailleurs, l'option `-t` permet de spécifier un autre paramètre : le niveau maximal de génération de *thread*. Par défaut, pour traiter un sous-arbre, le système tente systématiquement de générer un nouveau *thread* si le nombre maximal spécifié ci-dessus n'est pas atteint (suite à la terminaison de *thread*). Or, les *thread* créés pour des sous-arbres proches des feuilles ont une durée de vie très courte ; sans limitation, on risque d'assister à une effervescence de créations/terminaisons de *thread* proches des feuilles, préjudiciable aux performances. Pour éviter ce phénomène, il est préférable de limiter le niveau (nombre de UTOs) où le système autorise la création de *thread* ; ce niveau est donné par le deuxième terme de l'option `-t`. Par exemple, pour un niveau égal à 5 (valeur par défaut), chaque *thread* traitera lui-même les sous-arbres comportant 6 UTOs ou plus, sans tenter de créer de nouveaux *thread*.

Noter que l'exploration parallèle n'est pas implémentée avec l'option `-oa1` ; dans ce cas, l'exploration est toujours séquentielle.

Tranche 3 : calcul de l'arbre d'intersection.

L'arbre d'intersection est calculé par 3iapr. La sortie de ce programme est produite dans la section 7 du fichier de sortie.

Se reporter au manuel de 3iapr pour le détail de cette section.

Tranche 4 : calcul des Indices de Rétention par caractère.

Pour chaque caractère est calculée la somme pondérée des pondérations fractionnaires de ses nœuds. Plus précisément, cette valeur carPF est calculée comme suit :

- Pour chacun des nœuds non terminaux dans l'arbre du caractère est calculé le produit nPF × Ntts (nPF = pondération fractionnaire du nœud, Ntts = nombre de 3is « non redondants » du nœud). Noter que cette valeur égale le nombre de 3is indépendants Itts (voir tranche 0).
- La somme carPF du caractère est la somme des (nPF × Ntts) de tous ses nœuds.

Pour un arbre donné, on calcule pour chaque caractère la valeur carPFi de la même façon que ci-dessus, mais en ne tenant compte que des 3is présents dans l'arbre (et non de tous les 3is de base comme pour carPF).

Pour un arbre donné et pour un caractère donné, l'Indice de Rétention RI est égal au rapport carPFi/carPF.

3ia calcule ce rapport par caractère pour tous les arbres retenus dans la tranche 2. Dans le fichier en sortie *filout*.3ri, chaque arbre est identifié par le numéro (>0) qu'il porte dans la liste produite à la section 6 du fichier de sortie. Le rapport est donné multiplié par 1000. Pour les caractères ne générant pas de 3is, le rapport, sans signification, est remplacé par le signe ″−″.

Si la tranche 3 a été exécutée et si un arbre d'intersection a été trouvé, 3ia calcule aussi ces RI pour l'arbre d'intersection, identifié par le numéro zéro.

Historique.

version 3.7 : décembre 2009
Changement de nom → 3ia.

version 3.6 : janvier 2009
Affichage du pourcentage de progression lors de la recherche des arbres de compatibilité. Entrée PROGRES du fichier de configuration.

version 3.5 : septembre 2007
Remplacement des extension .inf et .ri par .3iz et .3ri.
Affichage de la valeur de RI_DEC dans le fichier .3ri.

version 3.4 : mai 2007
Ajout et prise en compte de l'entrée RI_DEC du fichier de configuration.

version 3.3 : octobre 2006
Prise en charge des pondérations fractionnaires sur 64 bits (au lieu de 32 bits auparavant). Tous les calculs de PF (sur les nœuds et sur les 3is) sont effectués en entiers non signés sur 64 bits : dans ces conditions, le risque de débordement devient très faible, et les PF sont donc en général calculées en valeurs fractionnaires, donc exactes (voir la Note du § « Tranche 0 : chargement de l'univers »).

version 3.2 : septembre 2006
Modification de la méthode de lecture du fichier d'entrée : les caractères ne sont plus stockés, mais traités au fur et à mesure de la lecture du fichier. Ceci amène un gain considérable de rapidité de lecture pour les très

gros fichiers (plusieurs millions de caractères), ainsi qu'un gain d'occupation d'espace mémoire. Voir le § « Optimisation du chargement ». En contrepartie, quelques régressions sont à noter :

- La section 3 du fichier *fileout*.inf n'affiche plus les 3is redondants.
- L'unicité des numéros des caractères n'est plus contrôlée.

Introduction de l'option -I.
Modification de la signification de l'option -v.
Pour le calcul du RI (tranche 4), la valeur produite est ramenée à un entier par arrondi et non plus par troncature.
De même, pour la renormalisation de la PF flottante des 3is par rapport au maximum MAXPPF, la valeur produite est ramenée à un entier par arrondi et non plus par troncature
Affichage d'un message si la PF de certains 3is est nulle.

version 3.1 : août 2006

Implémentation de l'optimisation par tri des sous-arbres. Introduction de l'option -oh1.
Choix possible du niveau maximal de génération de *thread*. Ajout du second terme de l'option -t.
Correction du calcul des RI par caractères pour l'arbre d'intersection.

version 3 : mai 2006

Implémentation *multithread*. Introduction de l'option -t.
Introduction de la clef de configuration TTS_OPT, et de l'option -C.

version 2.2 : avril 2006

Assouplissement du format du fichier d'entrée : seules les sections utiles sont prises en comptes. Les sections non utilisées peuvent manquer.
Modification importante du code de lecture du fichier d'entrée en corrélation avec la version 2.1 de prtts et l'introduction de ttsbiog.

version 2.1 : mars 2006

Multilingue.
Ajout de l'entrée FORMAT du fichier de configuration. Introduction du double format pour les 3is.
Lors du calcul des arbres compatibles, ajout de la nouvelle UTO à la racine avant l'ajout aux autres positions (auparavant, l'ajout à la racine se faisait après).
Affichage systématique de la somme des PF des 3is compatibles, même lors de la recherche par minimisation des incompatibilités (section 6 du fichier de sortie).
Nomenclature 3is.
Option –V.

version 2 : février 2006.

Intégration du calcul de l'arbre d'intersection par prtts.
Reformatage du fichier de sortie, pour filtrage par scripts sed.
Suppression de la génération du fichier Nexus, reportée dans ttsout.

version 1 : juin 2005.