

Cours d'Informatique de Base  
2ème Candidature en Sciences Economiques et  
Sociales – Option Informatique  
Travaux Pratiques:  
Système de Gestion de Bases de Données  
Bibliographiques Hétérogènes  
Solution de l'étape de conceptualisation et de  
structuration  
Année Académique 2000-2001

Michaël Petit

15 mars 2001

## Table des matières

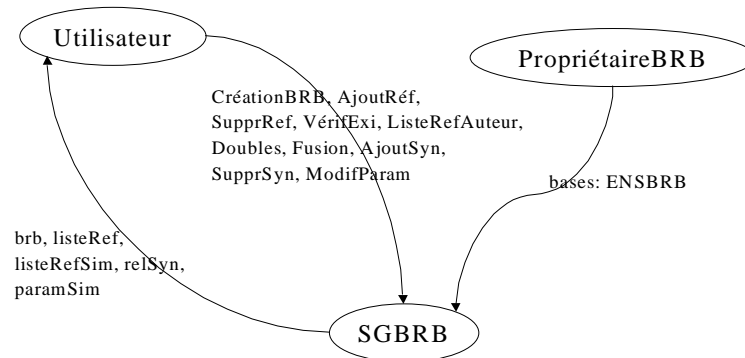
<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Diagramme de contexte</b>	<b>3</b>
2.1	Acteurs . . . . .	3
2.2	Interactions . . . . .	3
<b>3</b>	<b>Types de données</b>	<b>4</b>
3.1	ENSBRB . . . . .	4
3.2	BRB . . . . .	4
3.3	REF . . . . .	4
3.4	LISTEPAIREREF . . . . .	4
3.5	RELSYN . . . . .	4
3.6	STATSBRB . . . . .	5
3.7	PARAM . . . . .	5
<b>4</b>	<b>Fonctionnalités principales</b>	<b>5</b>
4.1	CréationBRB . . . . .	5
4.2	AjoutRéf . . . . .	6
4.3	SupprRéf . . . . .	7

4.4	VérifExi	8
4.5	Doubles	10
4.6	ListeRefAuteur	11
4.7	Fusion	11
4.8	ModifParam	12
4.9	AjoutSyn	13
4.10	SupprSyn	13
<b>5</b>	<b>Sous-Fonctions</b>	<b>14</b>
5.1	VérifExiBRB	14
5.2	AfficherErreur	14
5.3	CréerBRBVide	14
5.4	TrouverRefSimilaire	15
5.5	CalculStatsRef	16
5.6	AjoutChampRef	16
5.7	AjoutRefFin	16
5.8	SupprRefPosition	17
5.9	AfficherContenuBRB	17
5.10	SupprBRB	18
5.11	AfficherPairesRef	18
5.12	AjouterPaireRef	19
5.13	MotAppartientEns	19
5.14	EgalitéEns	19
5.15	UnionEns	19
5.16	AjoutEnsSyn	19
5.17	RetraitEnsSyn	20
5.18	AfficherRef	20
5.19	ComparerRefs	20
5.20	ComparerChamps	21
5.21	ComparerPhrases	21
5.22	ComparerMots	22
<b>6</b>	<b>Modules</b>	<b>22</b>
6.1	Module 1 : Principal	22
6.2	Module 2 : BRB	22
6.3	Module 3 : Similitude	22
6.4	Module 4 : ListePaireRef	23
6.5	Module 5 : RelSyn	23

# 1 Introduction

Ce document décrit une solution pour la première étape du TP du cours d'Informatique de Base de l'année académique 2000-2001. Il présente la description de la conceptualisation et de la structuration du problème à résoudre dans le TP.

## 2 Diagramme de contexte



### 2.1 Acteurs

Les trois acteurs représentés sont :

**SGBRB** Le programme à réaliser (gestionnaire de bases de références bibliographiques hétérogènes. Il gère un ensemble de BRBs (en partie fournies par les acteurs *PropriétaireBRB*), une liste de mots synonymes et les paramètres de similitude que l'acteur *Utilisateur* veut utiliser pour calculer la similitude entre références bibliographiques.

**PropriétaireBRB** Ils fournissent un ensemble de BRB au programme (*bases*).

**Utilisateur** L'utilisateur du programme. Il active les différentes fonctionnalités du programme et obtient en retour des informations.

### 2.2 Interactions

L'interaction du *PropriétaireBRB* vers le *SGBRB* indique la transmission de la donnée *bases* qui représente l'ensemble des BRBs mises à disposition du programme.

Les interactions de l'*Utilisateur* vers le *SGBRB* consistent en l'appel des fonctionnalités. Les interactions ont donc le même nom que les fonctionnalités. Additionnellement, pour certaines fonctionnalités, l'utilisateur doit fournir certaines données (par exemple, un clé et une référence dans le cas de *AjoutRéf*). Ces interactions ne sont pas montrées explicitement.

Les interactions du *SGBRB* vers l'*Utilisateur* représentent les réponses du programme à l'utilisateur. Ces réponses sont des données qui sont affichées par le programme à l'utilisateur en fonction des fonctionnalités que celui-ci a invoquées. Par exemple, lorsque l'utilisateur envoie l'interaction *VérifExi* au programme, il recevra en retour l'affichage de la donnée *listeRef*, l'ensemble des références similaires à la référence qu'il a donnée.

## 3 Types de données

### 3.1 ENSBRB

ENSBRB=ENSEMBLE[BRB]

Un ensemble de BRBs.

### 3.2 BRB

BRB=RECORD[nom: STRING, contenu: SUITEREF, stats: STATSBRB]

SUITEREF=SUIE[REF]

Une BRB possède un nom, un contenu et des statistiques. Le contenu est une suite (séquence) de références.

### 3.3 REF

REF=SUIE[CHAMP]

CHAMP=RECORD[nom: SUITEMOTS, val: SUITEMOTS]

SUITEMOTS=SUIE[MOT]

MOT=SUIE[CHAR]

Une référence est une suite de champs. Chaque champ possède un nom et une valeur. Chacun de ceux-ci sont des phrases (suites de mots). Un mot est une suite de caractères (STRING).

### 3.4 LISTEPAIREREF

LISTEPAIREREF=SUIE[RECORD[ref1: REF, ref2: REF]]

Une liste de paires de références est une suite donc chaque élément contient deux éléments qui sont des références.

### 3.5 RELSYN

RELSYN=ENSEMBLE[ENSEMBLE[MOT]]

ENSMOTS=ENSEMBLE[MOT]

Une relation de synonymie définit quels mots sont synonymes entre eux. Tous les mots qui sont synonymes sont regroupés dans un ensemble. Une relation de synonymie sera donc un ensemble d'ensembles.

### 3.6 STATSBRB

STATSBRB=SUITE[STATSREF]  
STATSREF=...

Les statistiques d'une BRB sont une suite de statistiques de références. Le premier élément de cette suite contient les statistiques de la première référence de la BRB, le deuxième contient celles de la deuxième référence,...

Les statistiques d'une référence (type à définir) contiennent notamment, par exemple, la longueur de la référence en mots, en champs et en caractères, mais contiennent aussi les statistiques de chacun des champs qui sont présents à l'intérieur de la référence. Les statistiques d'un champ contiennent à la fois des statistiques pour le nom de champ et des statistiques pour la valeur de champ.  
...

### 3.7 PARAM

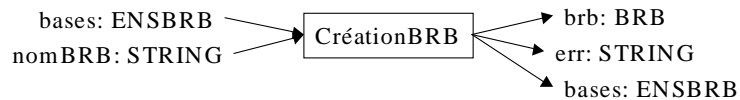
PARAM=...

Les paramètres de similitude (type à définir) représentent les préférences de l'utilisateur pour la vérification de similitude. Ils sont bien détaillés dans l'énoncé.

## 4 Fonctionnalités principales

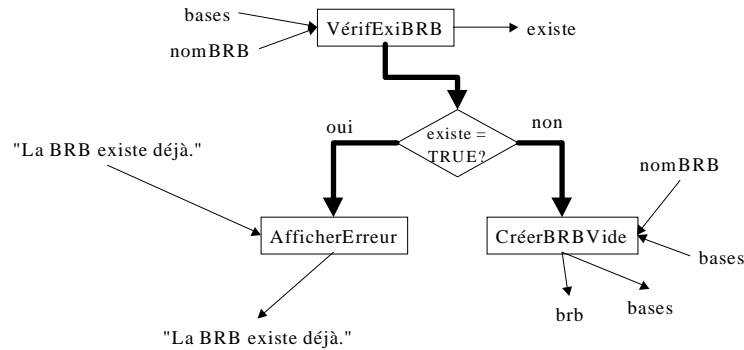
### 4.1 CréationBRB

#### 4.1.1 Description



A partir d'un nom de BRB (*nomBRB*) fourni par l'utilisateur, création d'une BRB vide (*brb*) portant ce nom. Si le nom fourni par l'utilisateur n'est pas le nom d'une BRB existante, la nouvelle BRB est ajouté à l'ensemble des BRBs existantes (*bases*). Sinon, le message d'erreur (*err*) "La BRB existe déjà" est affiché à l'écran.

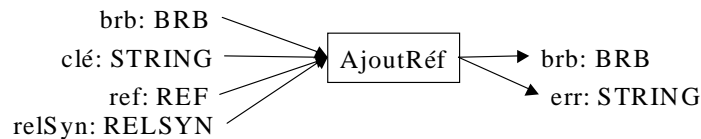
### 4.1.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



On utilise d'abord la fonction *VérifExiBRB* pour vérifier l'existence d'une BRB du nom donné en entrée (*nomBRB*) dans l'ensemble des BRB connues (*bases*). Si il en existe une, on utilise la fonction *AfficherErreur* pour afficher le message "La BRB existe déjà." à l'écran. Sinon, on utilise la fonction *CréerBRBVide* pour créer une BRB vide qui porte le nom donné et qui est ajoutée à l'ensemble des BRB connue.

## 4.2 AjoutRéf

### 4.2.1 Description

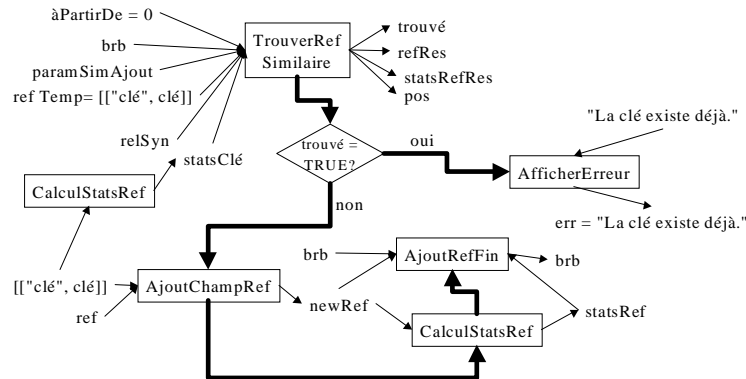


A partir d'une référence sans clé (*ref*) et d'une clé (*clé*), ajouter cette référence (avec un champ supplémentaire qui contient sa clé) à une BRB donnée (*brb*).

Si la BRB en entrée ne contient pas de référence ayant comme valeur de clé la même valeur que celle données en entrée, la référence est ajoutée à la BRB. Sinon, la BRB est inchangée et un message d'erreur (*err*) le signale.

Pour déterminer si une référence de la BRB possède la même clé, on vérifiera que cette référence ne possède pas un champ de clé similaire à la clé fournie en entrée. Un champ de clé similaire sera un champ dont le nom est "clé" (on admettra éventuellement un faute d'orthographe) ou un synonyme de "clé" et dont la valeur sera exactement la même que la clé recherchée.

## 4.2.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



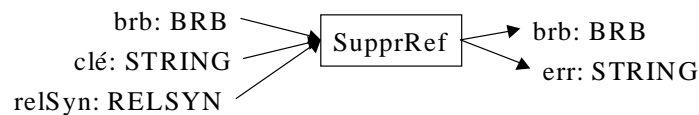
On commence par vérifier qu'il n'existe pas une référence de même clé dans la BRB en utilisant la fonction *TrouverRefSimilaire* (voir la définition de cette fonction). On recherche avec cette fonction la première référence de la BRB qui a au moins un champ similaire à la référence constituée du seul champ *["clé", clé]* en fonction des critères *paramSimAjout* et de la liste des synonymes (*relSyn*). La fonction *TrouverRefSimilaire*, pour vérifier la similitude, a besoin de connaître les statistiques de la référence recherchée. On doit donc d'abord calculer ces statistiques au moyen de la fonction *CalculStatRef* pour la référence constituée du seul champ *["clé", clé]*.

Si on trouve une telle référence similaire, on utilise la fonction *AfficherErreur* pour le dire à l'utilisateur. Sinon, on agrège la référence à ajouter donnée en entrée (*ref*) et le champ qui donne sa clé au moyen de la fonction *AjoutChampRef*.

Ensuite, on calcule les statistiques de la référence obtenue avant d'ajouter la référence et ses statistiques à la fin de la BRB (fonction *AjoutRefFin*).

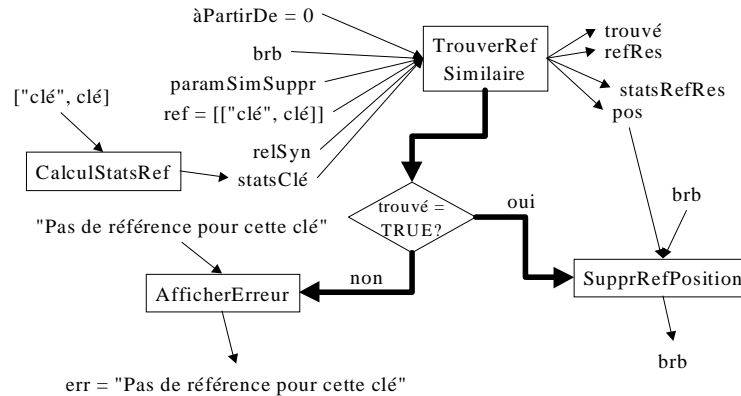
## 4.3 SupprRéf

### 4.3.1 Description



A partir d'une valeur de clé donnée (*clé*) par l'utilisateur, supprimer une référence d'une BRB (*brb*). Si la BRB en entrée contient une référence ayant comme valeur de clé la même valeur que celle données en entrée, la référence correspondante est supprimée. Sinon, un message d'erreur indique qu'aucune référence possédant cette clé n'a été trouvée.

### 4.3.2 Décomposition en sous-fonctions et Diagramme d'enchaînement

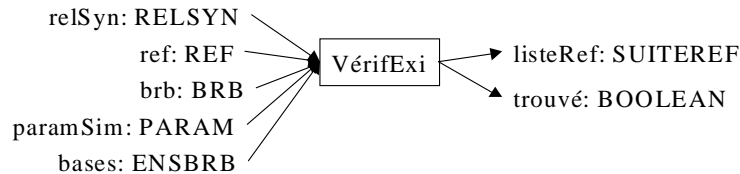


On commence par vérifier qu'il existe une référence de même clé dans la BRB en utilisant la fonction *TrouverRefSimilaire* (voir la définition de cette fonction). On recherche avec cette fonction la première référence de la BRB qui a au moins un champ similaire à la référence constituée du seul champ `["clé", clé]` en fonction des critères *paramSimSuppr* et de la liste des synonymes (*relSyn*). La fonction *TrouverRefSimilaire*, pour vérifier la similitude, a besoin de connaître les statistiques de la référence recherchée. On doit donc d'abord calculer ces statistiques au moyen de la fonction *CalculStatRef* pour la référence constituée du seul champ `["clé", clé]`.

Si l'on ne trouve pas une telle référence similaire, on utilise la fonction *AfficherErreur* pour le dire à l'utilisateur. Sinon, on utilise la position *pos* renvoyée par la fonction *TrouverRefSimilaire* pour supprimer la référence avec la fonction *SupprRefPosition*.

## 4.4 VérifExi

### 4.4.1 Description

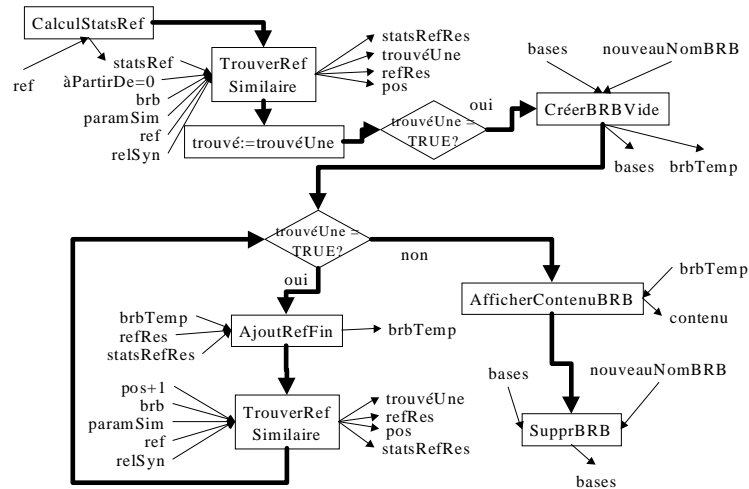


Vérification de l'existence d'une référence similaire à une référence (*ref*) donnée par l'utilisateur dans une BRB (*brb*). La fonction renvoie la liste (*listeRef*) de toutes les références similaires à la référence donnée en fonction de la relation de synonymie (*relSyn*), des paramètres définissant le degré de similitude désiré des mots, phrases, champs et références (*paramSim*).

Le résultat booléen *trouvé* est vrai si et seulement si au moins une référence

similaire a été trouvée.

#### 4.4.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



On commence par rechercher la première référence similaire dans la BRB en utilisant la fonction *TrouverRefSimilaire*. Les paramètres utilisés pour cette fonction correspondent donc aux paramètres en entrée, sauf pour *àPartirDe* qui prend la valeur zéro pour indiquer que l'on recherche à partir du début de la BRB. Pour utiliser cette fonction, on a également besoin de connaître les statistiques de la référence recherchée. Celle-ci sont donc d'abord calculées via la fonction *CalculStatsRef*. Les résultats de cette fonction sont un booléen *trouvéUne* qui indique si une référence similaire a été trouvée, et si cela est le cas, la référence trouvée *refRes*, ses statistiques *statsRefRes* et la position *pos* à laquelle elle a été trouvée.

Le résultat *trouvé* de *VérifExi* prend la valeur renvoyée par la fonction *TrouverRefSimilaire* qui indique si une référence similaire a été trouvée (*trouvéUne*). Si cette valeur est *FALSE*, la fonction est terminée (sans rien afficher à l'écran). Sinon, on continue en créant une BRB temporaire vide *brbTemp* qui servira à stocker toutes les références similaires avant de les afficher à l'écran. Pour cela, on utilise la fonction *CréerBRBVide* en lui passant comme paramètre un nom de BRB inexistant *nouveauNomBRB*.

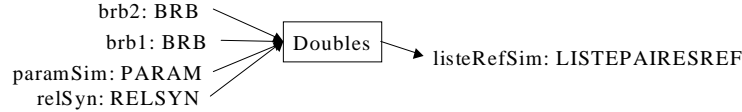
On entre ensuite dans une boucle dont la condition de sortie est (*trouvéUne = FALSE*). Dans cette boucle, on ajoute la référence similaire que l'on vient de trouver et ses statistiques dans la BRB temporaire avec la fonction *AjoutRefFin*. On recherche ensuite la référence similaire suivante avec la fonction *TrouverRefSimilaire* mais à partir de la position qui suit celle où l'on a trouvé la référence qui vient d'être ajoutée à la BRB temporaire.

Lorsque l'on sort de la boucle, il ne reste plus qu'à afficher le contenu de la BRB temporaire avec la fonction *AfficherContenuBRB* et à effacer cette BRB

temporaire avec la fonction *SupprBRB*.

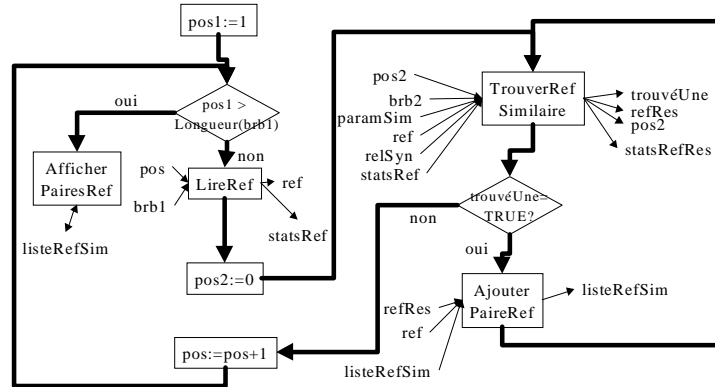
## 4.5 Doubles

### 4.5.1 Description



Rechercher les doubles dans deux BRB données *brb1* et *brb2*. Des doubles sont des références qui sont similaires en fonction d'une liste de synonymes *relSyn* et des paramètres définissant le degré de similitude désiré des mots, phrases, champs et références *paramSim*. Le résultat est une liste de toutes les paires de références similaires *listeRefSim* qui est affiché à l'écran. Chaque paire contient une référence de la première BRB et une de la deuxième.

### 4.5.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



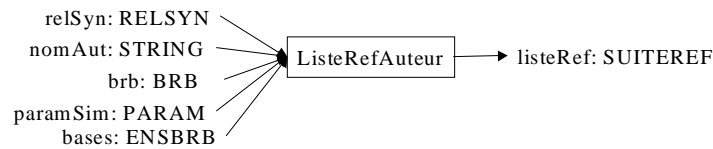
La fonctionnalité consiste en une boucle dans laquelle on parcourt la première BRB en lisant une à une ses références avec la fonction *LireRef*. Pour chaque référence lue, on recherche toutes les références similaires dans la deuxième BRB.

Pour ce faire, on exécute une boucle imbriquée dans laquelle sont exécutées les fonctions *TrouvéRefSim* (qui recherche dans la seconde BRB une référence similaire à celle lue dans la première) et *AjoutPaireRef* qui ajoute ces deux références similaires au résultat. On sort de cette boucle quand on ne trouve plus dans la seconde BRB de référence similaire à celle lue dans la première. On passe alors à la référence suivante de la première BRB.

On sort de la boucle principale lorsque l'on a parcouru toute la première BRB. Il ne reste plus alors qu'à afficher le résultat avec la fonction *AfficherPairesRef*.

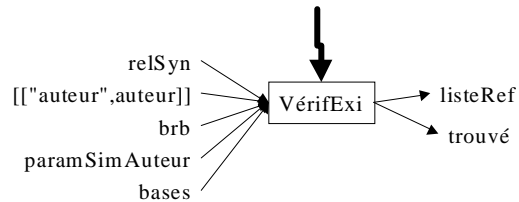
## 4.6 ListeRefAuteur

### 4.6.1 Description



Recherche de tous les documents d'un auteur (*nomAut*) dans une BRB donnée (*brb*). Le résultat affiché est la liste de toutes les références dont le champ "auteur" ou un champ synonyme (au vu de la liste de synonymes *relSyn* donnée en entrée) est similaire avec le nom d'auteur spécifié en entrée, au vu des paramètres définissant le degré de similitude désiré du nom d'auteur (qui est un mot).

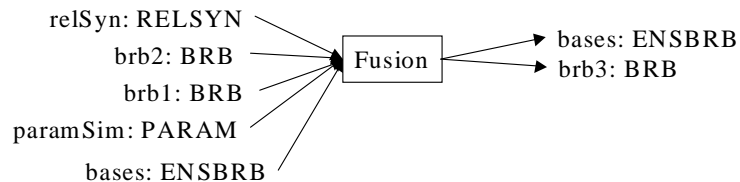
### 4.6.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



On utilise simplement la fonction *VérifExi* en lui passant les bons paramètres.

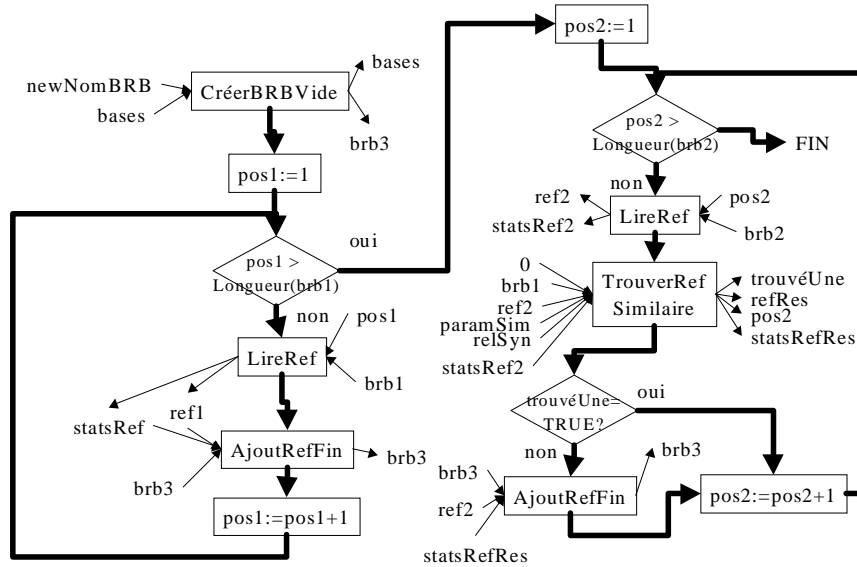
## 4.7 Fusion

### 4.7.1 Description



Fusion de deux BRB données en entrée (*brb1* et *brb2*). Le résultat est une nouvelle BRB (*brb3*) ajoutée à l'ensemble des bases existantes (*bases*) et contenant toutes les références des deux BRB dans laquelle aucune référence n'est similaire à une autre en fonction d'une liste de synonymes (*relSyn*) et de paramètres définissant le degré de similitude désiré des mots, phrases, champs et références (*paramSim*).

#### 4.7.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



La fonctionnalité commence par la création de la BRB résultat qui est initialement vide, avec la fonction *CréerBRBVide*. Ensuite, on entre dans une boucle dans laquelle on parcourt la première BRB en lisant une à une ses références avec la fonction *LireRef*. Pour chaque référence lue, on l'ajoute à la BRB résultat avec *AjoutRefFin*.

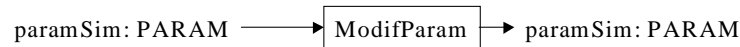
Lorsque toutes les références de la première BRB ont été lues et ajoutées au résultat, on sort de la boucle et on entre dans une seconde boucle qui parcourt la seconde BRB en lisant une à une ses références avec la fonction *LireRef*.

Pour chaque référence lue, on vérifie que cette référence n'a pas déjà une référence similaire qui se trouve dans la première BRB. Pour cela, on utilise la fonction *TrouverRefSimilaire*. Si cette fonction renvoie FALSE comme valeur pour *trouvéUne*, il n'y a pas de référence similaire et on peut alors ajouter la référence de la deuxième BRB à la BRB résultat avec *AjoutRefFin*.

La fonction se termine lorsque l'on a traité toutes les références de la seconde BRB.

## 4.8 ModifParam

### 4.8.1 Description



L'utilisateur introduit les nouveaux paramètres pour le similitude. Ceux-ci sont enregistrés par le programme afin d'être utilisés pour les autres fonctions.

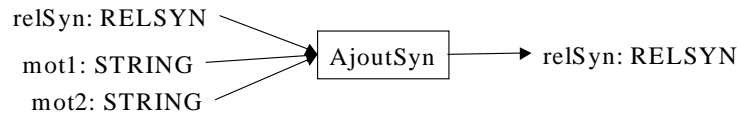
NB : On peut réutiliser les paramètres anciens pour ne pas obliger l'utilisateur à réentrer des valeurs pour tous les paramètres, mais seulement ceux qu'il veut modifier.

#### 4.8.2 Décomposition en sous-fonctions et Diagramme d'enchaînement

Cette fonction est suffisamment simple et ne nécessite pas de décomposition.

### 4.9 AjoutSyn

#### 4.9.1 Description



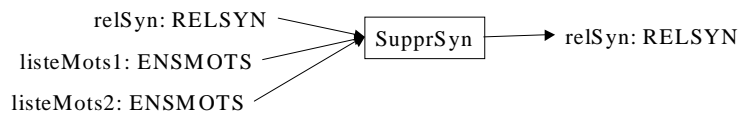
Ajout d'une paire de mots (*mot1* et *mot2*) à la relation de synonymie (*relSyn*). La relation de synonymie est mise à jour pour que les deux mots de la paire donnée en entrée apparaisse comme synonymes.

#### 4.9.2 Décomposition en sous-fonctions et Diagramme d'enchaînement

Pour spécifier et implémenter cette fonction, les sous-fonctions suivantes seront utiles : *MotAppartientEns*, *EgalitéEns*, *UnionEns*, *AjoutEnsSyn*, *RetraitEnsSyn*. Leur enchaînement reste à définir.

### 4.10 SupprSyn

#### 4.10.1 Description



Suppression d'une relation de synonymie entre deux ensembles de mots. La relation de synonymie contient actuellement un ensemble de mots qui est partitionné en deux par les deux listes de mots données en entrées.

La relation de synonymie est mise à jour pour que les mots qui apparaisse dans chacune des listes restent synonymes entre eux mais que les mots qui sont dans des listes différentes ne soient plus synonymes.

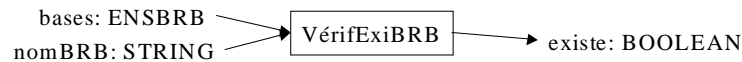
#### 4.10.2 Décomposition en sous-fonctions et Diagramme d'enchaînement

Pour spécifier et implémenter cette fonction, les sous-fonctions suivantes, qui sont les mêmes que pour le fonction *AjoutSyn* seront utiles : *MotAppartientEns*, *EgalitéEns*, *UnionEns*, *AjoutEnsSyn*, *RetraitEnsSyn*. Leur enchaînement reste à définir.

## 5 Sous-Fonctions

### 5.1 VérifExiBRB

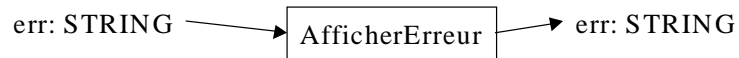
#### 5.1.1 Description



Vérifie l'existence d'une BRB d'un nom donné (*nomBRB*) parmi l'ensemble des BRBs existantes (*bases*). Renvoie TRUE si une telle BRB existe, FALSE sinon.

### 5.2 AfficherErreur

#### 5.2.1 Description



Affiche le message d'erreur *err* à l'écran.

### 5.3 CréerBRBVide

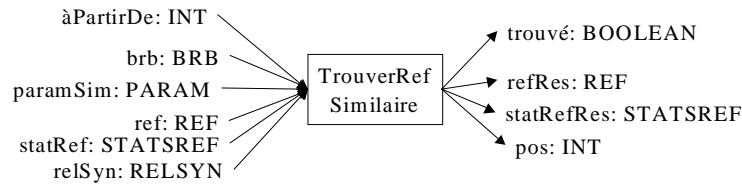
#### 5.3.1 Description



Crée une nouvelle BRB dont le nom est celui donné en entrée (*nomBRB*) et dont le contenu est vide. Cette BRB est ajoutée à l'ensemble des BRB existantes (*bases*).

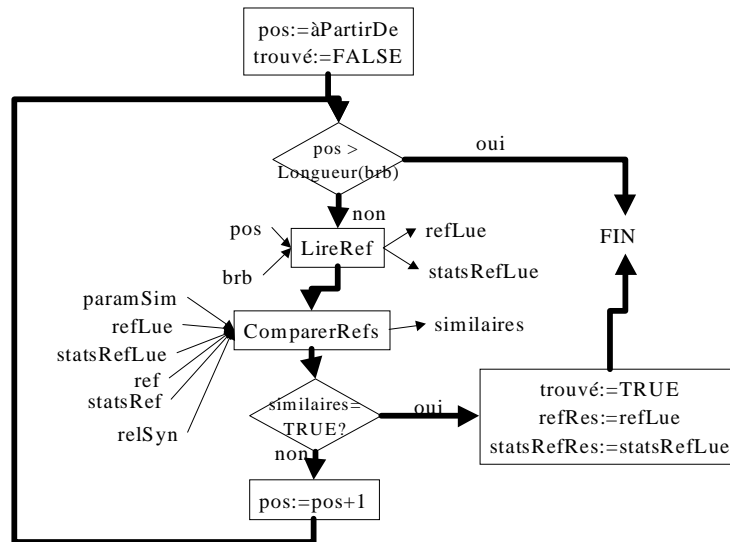
## 5.4 TrouverRefSimilaire

### 5.4.1 Description



Trouve dans une BRB donnée en entrée (*brb*) une référence (*refRes*) similaire à une référence passée en entrée (*ref*) et qui ne se trouve pas avant la position *àPartirDe*) donnée en entrée. Cette référence est similaire au vu de la relation de synonymie *relSyn* et des paramètres de similitude fournis par l'utilisateur (*paramSim*). Les résultats renvoyés sont le booléen *trouvé* qui indique si une référence a effectivement été trouvée, et dans ce cas, la référence trouvée (*refRes*), ses statistiques (*statsRefRes*) et sa position dans la BRB (*pos*).

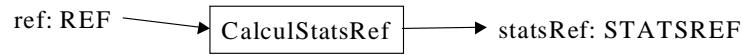
### 5.4.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



On parcourt la BRB *brb* à partir de la position *àPartirDe* en lisant les références une à une avec *LireRef* et on les compare avec la référence donnée en entrée via la fonction *ComparerRefs*. Si les références sont similaires, on sort de la boucle et on renvoie la référence, ses statistiques et sa position comme résultats. Sinon, on passe à la référence suivante dans la BRB. Si on arrive à la fin de la BRB sans avoir trouvé de référence similaire, on renvoie **FALSE** comme valeur pour *trouvé*.

## 5.5 CalculStatsRef

### 5.5.1 Description



Calculer les statistiques *statsRef* d'une référence donnée en entrée (*ref*). Les statistiques sont toutes les mesures quantitatives qui peuvent être calculées une fois pour toutes sur une référence et qui seront utiles pour évaluer la similitude entre références. Elles incluent, par exemple, la longueur de la référence en champs, en mots et en caractères, mais aussi les statistiques propres à chaque champ de la référence.

### 5.5.2 Décomposition en sous-fonctions et Diagramme d'enchaînement

Il peut être utile de décomposer cette fonction en une sous-fonction *CalculStatsChamp* qui se décompose elle-même en une sous-fonction *CalculStatsPhrase*, elle-même utilisant une sous-fonction *CalculStatsMot*. Ces sous-fonctions ne sont pas décrites dans ce document.

## 5.6 AjoutChampRef

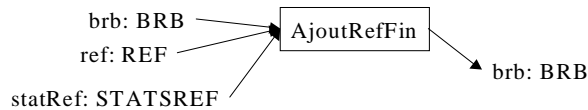
### 5.6.1 Description



Ajout d'un champ donné (*champ*) à une référence (*ref*).

## 5.7 AjoutRefFin

### 5.7.1 Description



Ajout d'une référence (*ref*) et de ses statistiques (*statsRef*) à la fin d'une BRB (*brb*).

## 5.8 SupprRefPosition

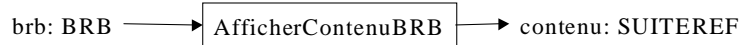
### 5.8.1 Description



Suppression de la référence se trouvant à une position donnée (*pos*) dans une BRB (*brb*).

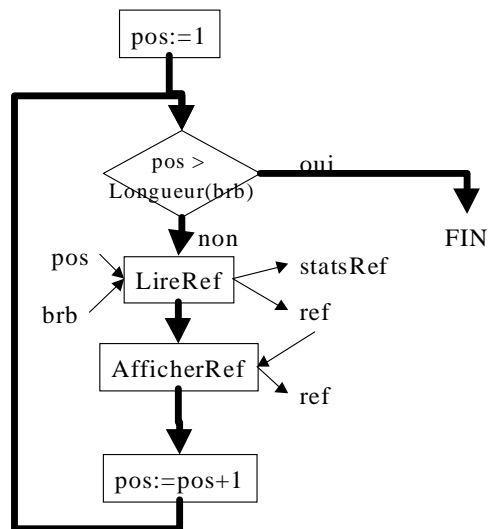
## 5.9 AfficherContenuBRB

### 5.9.1 Description



Afficher le contenu d'une BRB donnée (*brb*) qui est une suite de références (*contenu*).

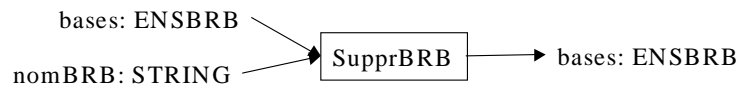
### 5.9.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



On parcourt la BRB avec la fonction *LireRef* et on affiche chacune des références lues avec *AfficherRef*.

## 5.10 SupprBRB

### 5.10.1 Description



Supprimer une BRB de nom donné (*nomBRB*) de l'ensemble des BRB existantes (bases).

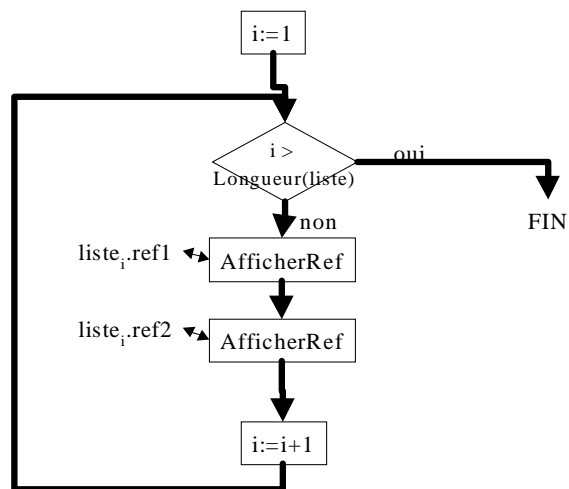
## 5.11 AfficherPairesRef

### 5.11.1 Description



Afficher une suite de paires de références (*liste*).

### 5.11.2 Décomposition en sous-fonctions et Diagramme d'enchaînement



Parcours de la suite des paires jusqu'à la fin. Pour chaque paire, affichage de chacun de ses éléments avec *AfficherRef*.

## 5.12 AjouterPaireRef

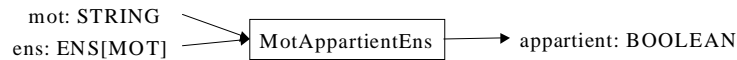
### 5.12.1 Description



Ajout d'une paire de références (*ref1* et *ref2*) à la fin d'une suite de références (*liste*).

## 5.13 MotAppartientEns

### 5.13.1 Description



Détermine si un mot donné *mot* appartient à une ensemble de mots donné *ens*. Si c'est le cas, le résultat *appartient* vaut TRUE, sinon il vaut FALSE.

## 5.14 EgalitéEns

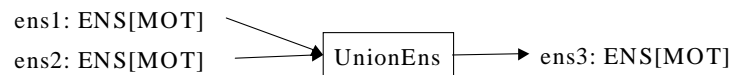
### 5.14.1 Description



Détermine si deux ensembles (*ens1* et *ens2*) sont égaux. Si c'est le cas, le résultat *égaux* vaut TRUE, sinon il vaut FALSE.

## 5.15 UnionEns

### 5.15.1 Description



Calcule un ensemble *ens3* comme étant l'union de deux ensembles données *ens1* et *ens2*.

## 5.16 AjoutEnsSyn

### 5.16.1 Description



Ajoute un ensemble de mots qui sont tous synonymes entre eux (*ens*) à une relation de synonymie. Aucun mot de *ens* n'appartient à la relation de synonymie *rel* avant d'y être ajouté.

## 5.17 RetraitEnsSyn

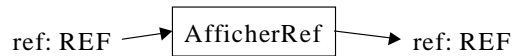
### 5.17.1 Description



Enlève un ensemble de mots *ens* qui est présent dans la relation de synonymie *rel*. La relation *rel* est modifiée pour que l'ensemble n'apparaisse plus dans la relation.

## 5.18 AfficherRef

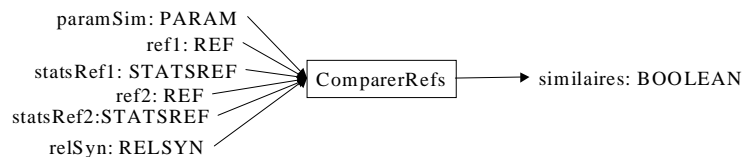
### 5.18.1 Description



Afficher une référence *ref* donnée à l'écran.

## 5.19 ComparerRefs

### 5.19.1 Description



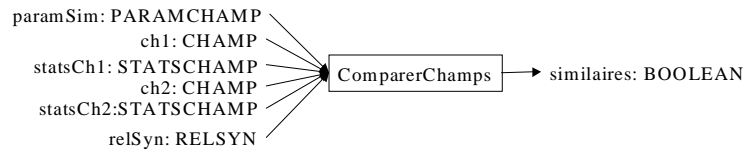
Comparer deux références données en entrée (*ref1* et *ref2*) en se basant notamment sur leurs statistiques (*statsRef1* et *statsRef2*) pour déterminer si elles sont similaires en fonction des paramètres de similitude données (*paramSim* et d'une relation de synonymie donnée (*relSyn*). Si c'est le cas, le résultat renvoyé *similaires* vaut TRUE, sinon il vaut FALSE.

### 5.19.2 Décomposition en sous-fonctions et Diagramme d'enchaînement

Pour déterminer si deux références sont similaires, on aura besoin, notamment de déterminer combien de leurs champs sont similaires en comparant leurs champs deux à deux via la fonction *ComparerChamps*.

## 5.20 ComparerChamps

### 5.20.1 Description



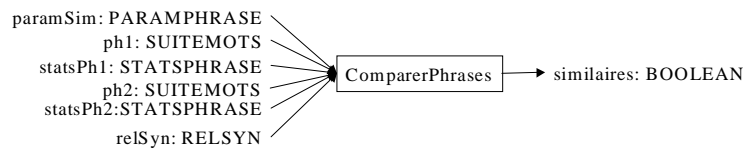
Déterminer si deux champs *ch1* et *ch2* sont similaires en se basant Comparer deux champs *ch1* et *ch2* donnés en entrée en se basant notamment sur leurs statistiques (*statsCh1* et *statsCh2*) pour déterminer s'ils sont similaires en fonction des paramètres de similitude de champs donnés (*paramSim* et d'une relation de synonymie donnée (*relSyn*). Si c'est le cas, le résultat renvoyé *similaires* vaut TRUE, sinon il vaut FALSE.

### 5.20.2 Décomposition en sous-fonctions et Diagramme d'enchaînement

Pour déterminer si deux champs sont similaires, on aura besoin de déterminer si leur noms et valeurs (qui sont tous deux des phrases) sont similaires. On utilisera pour cela la fonction *ComparerPhrases*.

## 5.21 ComparerPhrases

### 5.21.1 Description



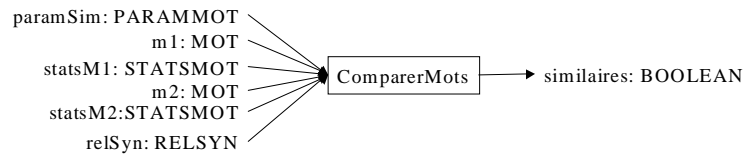
Comparer deux phrases données en entrée (*ph1* et *ph2*) en se basant notamment sur leurs statistiques (*statsPh1* et *statsPh2*) pour déterminer si elles sont similaires en fonction des paramètres de similitude de phrases donnés (*paramSim* et d'une relation de synonymie donnée (*relSyn*). Si c'est le cas, le résultat renvoyé *similaires* vaut TRUE, sinon il vaut FALSE.

### 5.21.2 Décomposition en sous-fonctions et Diagramme d'enchaînement

Pour déterminer si deux phrases sont similaires, on aura besoin, notamment de déterminer combien de leurs mots sont similaires en comparant leurs mots deux à deux via la fonction *ComparerMots*.

## 5.22 ComparerMots

### 5.22.1 Description



Comparer deux mots donnés en entrée ( $m1$  et  $m2$ ) en se basant notamment sur leurs statistiques ( $statsM1$  et  $statsM2$ ) pour déterminer s'ils sont similaires en fonction des paramètres de similitude de mots donnés ( $paramSim$  et d'une relation de synonymie donnée ( $relSyn$ )). Si c'est le cas, le résultat renvoyé *similaires* vaut TRUE, sinon il vaut FALSE.

## 6 Modules

### 6.1 Module 1 : Principal

Le module 1 regroupe les fonctionnalités principales du programme qui peuvent être invoquées par l'utilisateur à l'exception de celles relatives à la modification des paramètres de similitude et à la gestion de la relation de synonymie.

**Fonctions :** CréationBRB, AjoutRéf, SupprRéf, VérifExi, Doubles, ListeRefAuteur, Fusion, AfficherErreur, AjouterChampRef AfficheRef

### 6.2 Module 2 : BRB

Le module 2 regroupe toutes les opérations qui portent sur l'ensemble des BRB existantes ou sur le contenu de celles-ci, ainsi que les types de données associés à ces objets. Lors de la répartition des modules entre groupes, ce module devra être attribué au même groupe que le module 4.

**Types de données :** ENSBRB, BRB

**Fonctions :** VérifExiBRB, CréerBRBVide, AjoutRefFin, SupprRefPosition, SupprBRB, LireRef, AfficherContenuBRB

### 6.3 Module 3 : Similitude

Le module 3 regroupe les opérations relatives à la comparaison de références et de leurs éléments (champs, phrases et mots) ainsi que les types de données associés.

**Types de données :** PARAM, STATSBRB

**Fonctions :** TrouverRefSimilaire, ComparerRefs, ComparerChamps, ComparerPhrases, ComparerMots, ModifParam, CalculStatsRef

#### **6.4 Module 4 : ListePaireRef**

Le module 4 regroupe les opérations relatives aux listes de paires de références. Lors de la répartition des modules entre groupes, ce module devra être attribué au même groupe que le module 2.

**Type de Données :** LISTEPAIREREF

**Fonctions :** AjouterPaireRef, AfficherPairesRef

#### **6.5 Module 5 : RelSyn**

Le module 5 regroupe les opérations et le type de données relatifs à la gestion de la relation de synonymie.

**Types de Données :** RELSYN

**Fonctions :** AjoutSyn, SupprSyn, MotAppartientEns, EgalitéEns, UnionEns, AjoutEnsSyn, RetraitEnsSyn