|  |
| --- |
|  |
|  |
|  |
| **Projet réalisé par** |
| * Belkhous Redha Nabil * Zebouchi Mohammed |
|  |
|  |
|  |

***Projet de compilation***

***MyBib***

***Utilitaire de manipulation de Bibtex***



Sommaire

1. *Description du projet*
2. *Présentation de la grammaire*
3. *Rapport de développement*
4. *Manuel d’utilisation*
5. *jeu d’exemple*



Description du projet

1

**Description du projet**

Les fichiers de bases de données bibliographiques peuvent rapidement prendre une taille importante, et devenir, au fil des années, mal entretenus : par exemple, des doublons peuvent apparaitre, certaines conventions peuvent être respectées par endroits et pas d’autres, etc. Le sujet du projet est d’écrire un petit outil de manipulation de fichiers BibTEX, mybib, pour aider a résoudre ces problèmes.



2

Présentation de la grammaire

**Présentation de la grammaire**

Dans notre projet nous avons choisi de manipuler quatre types d’entités : *article*, *book*, *manual* et *misc.*

Un fichier Bibtex contient un ensemble d’entités, qui contiennent elles mêmes des attributs et il peut contenir des champs d’informations (Comment et String)

L’axiome S nous permet de donner soit des *Comments* suivies par une liste d’entités ou bien simplement une liste d’entités.

Une entité se compose d’un mot clef se référant à un des quatre types cités une accolade ouvrante et fermente puis entre les deux une clef et la liste des champs qui sont spécifiques à chaque types (champs obligatoires et optionnels)

Chaque champs est sous la forme d’un couple ***( Clef, Valeur )***.

S : LIST\_COMMENT LIST\_ENTITY |

LIST\_ENTITY

;

COMMENT: comment mc\_par\_our chaine mc\_par\_fer

;

STRING: string mc\_par\_our mot mc\_egal chaine mc\_par\_fer

;

LIST\_COMMENT: LIST\_COMMENT COMMENT STRING |

COMMENT STRING

;

HEADING: nom\_entity mc\_virg

;

ENTITY: article mc\_acc\_ouv HEADING LIST\_CHAMPS\_article mc\_acc\_fer |

book mc\_acc\_ouv HEADING LIST\_CHAMPS\_book mc\_acc\_fer |

manual mc\_acc\_ouv HEADING LIST\_CHAMPS\_manual mc\_acc\_fer |

misc mc\_acc\_ouv HEADING LIST\_CHAMPS\_misc mc\_acc\_fer

;

LIST\_ENTITY: LIST\_ENTITY ENTITY |

ENTITY

;

CHAMPS\_OBLIGATOIRES\_article: author mc\_egal valeur\_champs |

title mc\_egal valeur\_champs |

journal mc\_egal valeur\_champs |

year mc\_egal valeur\_champs

;

CHAMPS\_OPTIONNELS\_article: volume mc\_egal valeur\_champs |

number mc\_egal valeur\_champs |

pages mc\_egal valeur\_champs |

month mc\_egal valeur\_champs |

note mc\_egal valeur\_champs |

key mc\_egal valeur\_champs |

;

CHAMPS\_OBLIGATOIRES\_book: author mc\_egal valeur\_champs |

title mc\_egal valeur\_champs |

publisher mc\_egal valeur\_champs |

year mc\_egal valeur\_champs

;

CHAMPS\_OPTIONNELS\_book: volume mc\_egal valeur\_champs |

series mc\_egal valeur\_champs |

address mc\_egal valeur\_champs |

edition mc\_egal valeur\_champs |

note mc\_egal valeur\_champs |

key mc\_egal valeur\_champs |

month mc\_egal valeur\_champs |

;

CHAMPS\_OBLIGATOIRES\_manual: title mc\_egal valeur\_champs

;

CHAMPS\_OPTIONNELS\_manual: author mc\_egal valeur\_champs |

organization mc\_egal valeur\_champs |

address mc\_egal valeur\_champs |

edition mc\_egal valeur\_champs |

note mc\_egal valeur\_champs |

key mc\_egal valeur\_champs |

month mc\_egal valeur\_champs |

year mc\_egal valeur\_champs |

;

CHAMPS\_OPTIONNELS\_misc: author mc\_egal valeur\_champs |

title mc\_egal valeur\_champs |

howpublished mc\_egal valeur\_champs |

note mc\_egal valeur\_champs |

key mc\_egal valeur\_champs |

month mc\_egal valeur\_champs |

year mc\_egal valeur\_champs |

;

LIST\_CHAMPS\_article : LIST\_CHAMPS\_article mc\_virg

CHAMPS\_OBLIGATOIRES\_article |

LIST\_CHAMPS\_article mc\_virg CHAMPS\_OPTIONNELS\_article |

CHAMPS\_OBLIGATOIRES\_article |

CHAMPS\_OPTIONNELS\_article

;

LIST\_CHAMPS\_book: LIST\_CHAMPS\_book mc\_virg CHAMPS\_OBLIGATOIRES\_book |

LIST\_CHAMPS\_book mc\_virg CHAMPS\_OPTIONNELS\_book |

CHAMPS\_OBLIGATOIRES\_book |

CHAMPS\_OPTIONNELS\_book

;

LIST\_CHAMPS\_manual: LIST\_CHAMPS\_manual mc\_virg CHAMPS\_OBLIGATOIRES\_manual |

LIST\_CHAMPS\_manual mc\_virg CHAMPS\_OPTIONNELS\_manual |

CHAMPS\_OBLIGATOIRES\_manual |

CHAMPS\_OPTIONNELS\_manual

;

LIST\_CHAMPS\_misc : LIST\_CHAMPS\_misc mc\_virg CHAMPS\_OPTIONNELS\_misc |

CHAMPS\_OPTIONNELS\_misc

;

**Remarque : Plus de détails sur la partie Flex, Bison et routines sémantiques seront données.**



Rapport de développement

3

**Rapport de développement**

Nous nous basons tout au long du projet sur une structure en liste chainée, qui reprend le concept d’ensemble d’entités qui eux-mêmes regroupes des champs spécifiques.

Au lancement de la compilation, un fichier *.bib* est donné en entrée afinl’analyseur lexical Flex lise chaque entité et l’identifie comme étant propre au langage.

Par la suite, chaque entité est envoyée pour une analyse syntaxico-sementique faite par Bison en comparent avec la grammaire les entités reçues et leurs sens à travers les routines sémentiques.

Pour chaque entité lue par Flex et Bison, elle est stockée dans la structure en pensent à sauvegarder avec, quelques informations importantes comme indiquer si l’entité existe déjà dans la structure ou encore l’appartenance de chaque champs à son entité, chaque valeur à son champs et chaque clef à son type d’entité.

L’ajout des entités se fait grâce à une fonction « *addEntity*  » et celui des champs et leurs valeurs avec une autre fonction  «  *addParamToCurrentEntity*  » (*Routines sémentiques*)

Pour les fonctionnalités que nous avons développées, elle sont lancé en suffixant l’executable avec une otion ( Ex : ./lunch.exe  **-c** ).

Les fonctionnalités que nous avons réalisé sont :

1. mybib –c toto.bib
2. mybib –k toto.bib
3. mybib –s toto.bib
4. mybib –u toto.bib
5. mybib – t type toto.bib



Manuel d’utilisation

4

**Manuel d’utilisation**

L’utilisation de l’application est faite de manière a être très simple d’utilisation et intuitif.

On lance le makefile pour compiler et générer l’exécutable, ensuite il vous suffit de lancer l’exécutable avec l’option que vous désirez.

Si l’utilisateur veut :

* vérifie que chaque entrée a une clef qui lui est propre : il ajout à la fin de l’exécutable «  -c  »

Exemple : ./lunch.exe –c

* Remplace les clefs de toto.bib par des clefs normalisées : il ajoute le suffixe «  -k  »

Exemple : ./lunch.exe –k

* Pour tous les enregistrements, extraire (s’ils existent) les champs *publisher*, *organization*, *series*, *journal*, et les définir à partir d’une chaıne de caracteres BibTEX : il lance l’option « -s »

Exemple : ./lunch.exe –s

* supprime les doublons : pour le même type, les doublons sont retirés et stockés dans un fichier annexe de nom *bib.doublons*

Exemple : ./lunch.exe –u

* Ne garde que les entrées de type *type* (article, book, misc et manual)

Exemple : ./lunch.exe –t



5

Jeu d’exemple

**Jeu d’exemple**

Pour vous guider et vous permettre de voir toutes les fonctionnalités développées nous vous suggérons de suivre ce scénario :

* + Lancer le ***Command.sh***
  + Lancer l’exécutable sans aucune option pour bien voir que le fichier exemple.bib se compile correctement.
  + Faire des modifications dans ce dernier en faisant délibérément des erreurs et relancer l’exécutable pour distinguer la ligne ou se trouve l’erreur.
  + Revenir au fichier tel qu’il été initialement.
  + Lancer l’exécutable avec l’option «  -c  » pour pouvoir constater les éléments qui ont la même clef.
    - ./mybib.out –c *filename.bib*
  + Lancer ensuite l’exécutable avec l’option «  -k  » pour normaliser les clefs.
    - ./ mybib. out –k *filename.bib*
  + Lancer l’exécution avec l’option «  -s  » pour générer les *Comments.*
    - ./ mybib. out –s *filename.bib*
  + Lancer l’exécutable avec l’option «  -u  » pour supprimer les doublons et vérifier dans le fichier généré.
    - ./ mybib. out –u *filename.bib*
  + Lancer une dernière exécution avec l’option «  -t  ».
    - ./ mybib. out –t article *filename.bib* pour ne prendre que les entités de type article et les mettrent dans un autres fichier *Bibtex\_Selection.bib*