Compte Rendu
Compilation

January 11, 2016

Version :	0.1
Date :	January 11, 2016
Rédigé par :	Thomas Capet Yohann Henry

Contents

1 Introduction

Le projet consiste en la création d'un exécutable permettant la gestion de références contenues dans des fichiers `bibtex` ou `LaTeX`. L'exécutable devait notament extraire les clés cités d'un fichier d'extension `.tex`, ainsi que les données contenues dans un fichier `.bib`. L'application effectuerait ensuite selon les options différents traitement et afficherait le résultat sur la sortie standard ou un fichier.

2 Notice d'utilisation

2.1 Option -t

2.2 Option -o

2.3 Option -b

2.4 Option -s

2.5 Option -o

3 Fonctionnement global

4 Structures de données utilisées

4.1 Collections

Pour travailler, nous avons d'abord développé plusieurs structures de données génériques de type collection :

- Les listes ordonnées
- Les sets ordonnées (ainsi que son homologue non ordonné)
- Les maps sous la forme d'une liste ordonnée
- Les tables de hachage

4.2 Références

Nous avons ensuite créer un type Référence. Ce dernier représente toutes les données nécessaires pour enregistrer une référence d'un bibtex.

```
struct _ref {
    TypeReference type;
    char* id;
    char* champs[NB_CHAMP_REF];
};

typedef struct _ref * Reference;
```

Les attributs `type` et `id` sont trivialement le type et l'id de la référence. L'attribut `champs` est un tableau qui attribut à un type de champ une valeur en chaîne de caractères. Par défaut, cette valeur est initialisée avec une chaine de caractères vide. Plusieurs fonctions sont présentes dans `references.h` pour simplifier les différentes options demandées. Notamment, une des fonctions permet d'écrire sur un flux, le contenu d'une référence avec tous les champs obligatoires ou optionnels de ce type.

4.3 Les gestionnaire de références

Nous avons finalement créer un type RefManager, un gestionnaire de références. Ce dernier nous permait de stocker toutes les références contenus dans les `.tex` et les `.bib`. Une fois le gestionnaire remplis, on traite les données selon les options demandées.

```
struct _ref_manager {
        HashMap map;
        int onlyUpdateMode;
};

typedef struct _ref_manager* RefManager;
```

Le gestionnaire de références est une simple table de hachage avec pour clé, l'id d'une référence et en valeur, la référence correspondante. Comme pour référence, plusieurs fonctions utilitaires ont permis de simplifier les différentes options.

5 La grammaire

5.1 Les tokens

La grammaire est lancée en contenant le mode dans lequel elle est lancée. Elle reconnait donc soit un fichier Bibtex, soit un fichier Latex, mais jamais les deux en même temps.

5.1.1 Les tokens pour parser les .tex

- `CITE : \cite{[^}]+}`
- `NOCITE : \nocite{[^}]+}`
- `BIBNAME : \bibname{[^}]+}`
- `INCLUDE : \include{[^}]+}`
- `INPUT : \input{[^}]+}`

5.1.2 Les tokens pour parser les .bib

- `TYPEREF : @Article|@Book|@Booklet|@Conference|@Inbook|@Incollection |@Inproceedings|@Manual|@Mastersthesis|@Misc|@Phdthesis |@Proceedings|@Techreport|@Unpublished`
- `TYPECHAMP: address|abstract|annote|author|booktitle|chapter|crossref |edition|editor|eprint|howpublished|institution|isbn|journal|key|month|note |number|organization|pages|publisher|school|series|title|type|url|volume|year`
- `KEY : {[a-zA-Z0-9]+(:[a-zA-Z0-9]+)*,`
- `VAL : {.*},|{.*}\n`

5.2 Définition de la grammaire

```
file: bloc file
    |;

bloc: TYPEREF KEY champs '}';

champs: TYPECHAMP champs
    |TYPECHAMP VAL champs
    |;
```