Polytech'Lille — GIS 3 — Structures de Données — Feuille de TP numéro 5

Ce TP se déroule sur deux séances (quatre heures). La section 1 consiste uniquement à comprendre les mécanismes de base. La section 2 montre comment adapter ces mécanismes dans le cadre d'une application (dictionnaire traducteur). Le travail réalisé en section 2 sera réutilisé lors du TP sur les tables de hachage.

1 Implantation d'un module d'ABR

On veut réaliser un module dédié aux ABR, qui permette de faire fonctionner le programme principal suivant. Les valeurs sont des entiers. Vous pouvez vous inspirer de l'extrait de code disponible sur le site du cours mais il serait préférable que vous refassiez tout vous-même.

```
#include "abr.h"
#include <stdio.h>

int main ()
{   struct abr* racine;
    int x;
    racine = NIL;
    scanf ("%d", &x);
    while (x != -1)
    {   racine = ajouter_abr (x, racine);
        afficher_abr (racine);
        scanf ("%d", &x);
    }
    printf ("la hauteur de l'ABR est %d\n", hauteur_abr (racine));
    printf ("le nombre de noeuds de l'ABR est %d\n",
                                        nombre_noeuds_abr (racine));
    clear_abr (racine);
    return 0;
}
```

Question 1. Combien de fichiers doit-on écrire ?

Question 2. On souhaite compiler séparément ce qui peut l'être. Quelles seront les commandes de compilation nécessaires ?

Question 3. Écrire le fichier d'entête. Spécifier la structure de données, dans un commentaire, placé dans le fichier.

Question 4. Écrire le fichier source. Lors des essais, commentez, dans le programme principal, les appels aux fonctions que vous n'avez pas encore réalisées.

Question 5. Écrire une fonction qui permette de visualiser un ABR avec `dot` (voir feuille de TD). Gérer le cas de l'arbre vide et celui de l'arbre réduit à une feuille. Pour les tests, sortir l'affichage de la boucle du programme principal.

2 Dictionnaire Esperanto — Français

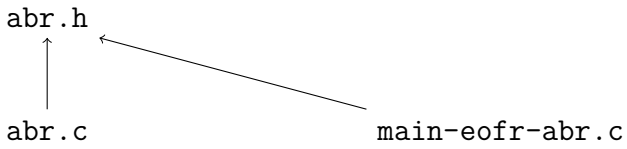

FIGURE 1 – Représentation graphique du traducteur à réaliser.

Dans cette section, on montre une utilisation des ABR dans le cadre d'une application (Figure 1), où les valeurs sont des couples (*clef*, *donnée satellite*). On en profite aussi pour étudier le comportement des ABR sur des données un peu volumineuses : un dictionnaire Esperanto-Français, obtenu à partir du site web `http://purl.org/net/panorama/vortaro/eofr.htm`, avec l'aimable autorisation de M. Godivier. La commande suivante permet de télécharger ce dictionnaire :

```
$ wget http://cristal.univ-lille.fr/~boulier/polycopies/SD/Esperanto-Francais.utf8
```

2.1 Modification du fichier d'entête

Dans cette section, les clefs et les données satellites sont des chaînes de caractères, comportant des lettres accentuées (la clef est en Esperanto, la donnée satellite est en Français). On utilise pour cela les *wide char* du langage C, décrits dans la section 2.1 de la feuille de TP 1.

Les étudiants qui le souhaitent peuvent utiliser le type `struct chaine` du TP 1 (il faut alors réinterpréter et adapter le reste de la section). Les autres doivent utiliser le type `wstring` dont la déclaration peut être insérée dans le fichier `abr.h` :

```
#include <wchar.h>
#define MAXLEN 80

typedef wchar_t wstring [MAXLEN];
```

Question 6. Modifier le type `struct abr` de telle sorte que les valeurs des nœuds soient un couple (*clef*, *donnée satellite*).

Question 7. Modifier le prototype de la fonction `ajout_abr`.

Question 8. Pour la recherche, on ne peut plus se contenter de retourner un booléen. La nouvelle fonction `recherche_abr` doit retourner une valeur de type `wchar_t*`. La valeur retournée doit être l'adresse de la donnée satellite si le mot Esperanto est trouvé, `(wchar_t*)0` sinon. Effectuer la modification.

2.2 Modification du code

Il s'agit maintenant de modifier le fichier `abr.c`. Dans un premier temps, on conseille de mettre en commentaire toutes les fonctions à l'exception de l'ajout, la recherche, l'affichage et le destructeur.

Question 9. Modifier le fichier en utilisant les fonctions de manipulation de chaînes de `wchar_t` suivantes (utiliser l'aide en ligne pour les détails) :

- `wcscmp` permet de comparer deux chaînes (analogue de `strcmp`) ;
- `wcscpy` permet de copier une chaîne (analogue de `strcpy`) ;
- `wprintf` généralise `printf`. En particulier, voici une instruction qui affiche la clef et la donnée satellite d'un `struct abr*` :

  ```
  wprintf (L"%ls:%ls\n", A->clef, A->satellite);
  ```

  Attention à bien remplacer tous les appels à `printf` par des appels à `wprintf`, dans tout le programme !

2.3 Le programme principal

Le programme principal donné Figure 2 permet de charger le dictionnaire dans l'ABR de racine `racine`. Il peut être téléchargé par la commande suivante :

```
$ wget http://cristal.univ-lille.fr/~boulier/polycopies/SD/main-eofr-abr.c
```

Question 10. La fonction `wcscmp` respecte-t-elle l'ordre alphabétique ? Même en Esperanto ?

Question 11. (optionnelle) Ajouter à ce programme des instructions permettant de rechercher des mots Esperanto et de donner leur traduction.

Question 12. Afficher la hauteur de l'ABR (utiliser `wprintf (L"%d", ...)`). Quel phénomène observe-t-on ?

Question 13. Pour atténuer le phénomène ci-dessus, il suffit de modifier le dictionnaire. Comment ? Effectuer la modification avec la commande `shuf` et vérifier expérimentalement.

```
/* main-eofr-abr.c */

#include <locale.h>
#include <wctype.h>
#include <assert.h>
#include <stdio.h>
#include "abr.h"

int main ()
{   struct abr* racine;
    wstring clef;
    wstring satellite;
    wint_t c;
    FILE* f;

    assert (setlocale (LC_ALL, "C.UTF-8") != NULL);
    f = fopen ("Esperanto-Francais.utf8", "r");
    assert (f != (FILE*)0);

    racine = NIL;
    c = fgetwc (f);
    while (c != WEOF)
    {   int i = 0;
        while (c != L':')
        {   clef [i] = c;
            i += 1;
            c = fgetwc (f);
        }
        clef [i] = L'\0';
        c = fgetwc (f);
        i = 0;
        while (c != L'\n')
        {   satellite [i] = c;
            i += 1;
            c = fgetwc (f);
        }
        satellite [i] = L'\0';
        racine = ajout_abr (racine, clef, satellite);
        c = fgetwc (f);
    }
    fclose (f);

    clear_abr (racine);
    return 0;
}
```

FIGURE 2 – Programme principal chargeant le dictionnaire `Esperanto-Francais.utf8` dans un ABR initialement vide.