Présentation

EF11 : Anas SEGHROUCHNI et Emile JEANDON

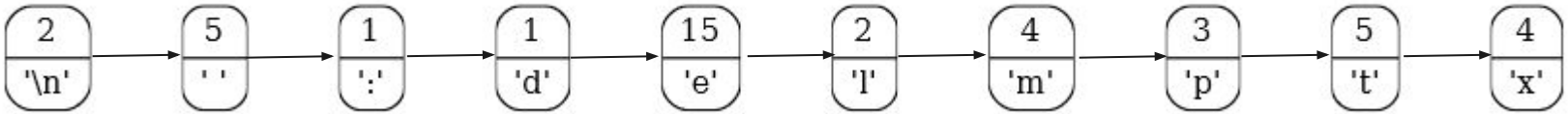
L’utilisation de Unbounded_String

- Pour pouvoir les concaténer (exemple : *010 111 01 → “010” “111” “01” → “01011101”*)
- Pour pouvoir les enregistrer **T_Octet** (exemple : *“01011101” → 01011101*), on peut alors ensuite enregistrer les octets dans le fichier compressé.
- Lors de la lecture du fichier compressé, on lit les octets codés bit par bit (exemple : *si dans la table de Huffman on a les octets compressés 0, 10 et 11 et que dans le fichier codé on a le code 01011010)*

Principales fonctions

Frequence

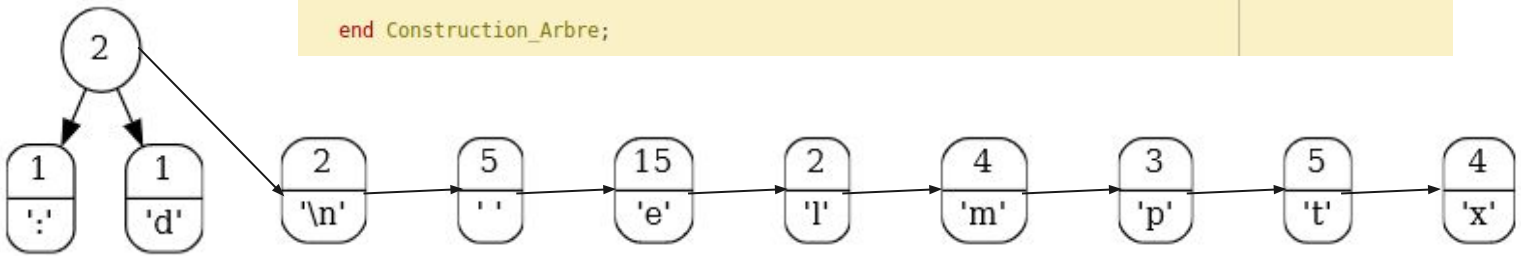
```
--Parcours le fichier à compressé et enregistre chaque octet rencontré dans la sda de noeud.
procedure Frequence(Sda : in out T_Chaine; File_Name : in String ;File : in out Ada.Streams.Stream_IO.File_Type) is
    S : Stream_Access;
    Octet : T_Octet;
    Ch:T_Chaine;
begin
    Initialiser(Sda);
    Open(File,In_File,File_Name);
    S:=Stream(File);
    while not End_Of_File(File) loop
        Octet:=T_Octet'Input(S);
        Enregistrer(Sda,Octet);
    end loop;
    Close(File);
    Ch:=new T_Cellule;
    Ch.all.Arbre:=new T_Noeud;
    Ch.all.Arbre.all.Gauche:=Null;
    Ch.all.Arbre.all.Droite:=Null;
    Ch.all.Arbre.all.Donnee:=0;
    Ch.all.Arbre.all.Cle:=-1;
    Ch.all.suivante:=Sda;
    Sda:=Ch;
end Frequence;
```



Construction Arbre

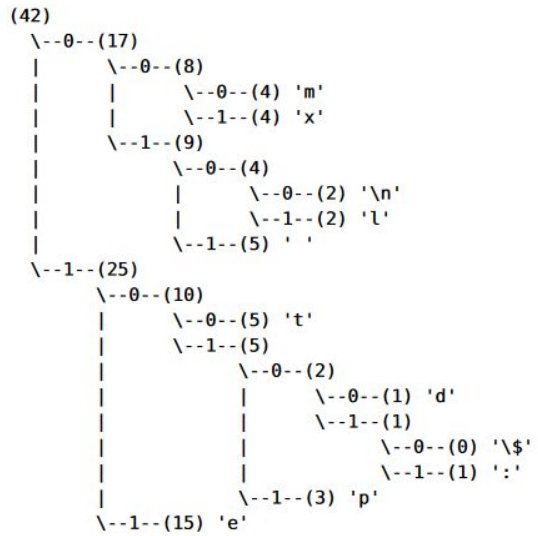
Construit un arbre à partir d'une Sda de noeud.

```
--Prend en entré un sda de noeud correspondant aux fréquences et renvoie l'ardre de Huffman.
procedure Construction_Arbre(Sda: in out T_Chaine; Arbre : out T_Arbre) is
    Ch:T_Chaine;
    Arbre_min1:T_Arbre;
    Arbre_min2:T_Arbre;
begin
    if Sda=Null then
        Put("La lca est vide");
    elsif Sda.all.suivante=Null then
        Arbre:=Sda.all.arbre;
    else
        Deux_Min(Sda,Arbre_min1,Arbre_min2);
        Ch:=new T_Cellule;
        Ch.all.Arbre:=new T_Noeud;
        Ch.all.Arbre.all.Gauche:=Arbre_min1;
        Ch.all.Arbre.all.Droite:=Arbre_min2;
        Ch.all.Arbre.all.Donnee:=Arbre_min1.all.Donnee+Arbre_min2.all.Donnee;
        Ch.all.suivante:=Sda;
        Sda:=Ch;
        Construction_Arbre(Sda,Arbre);

    end if;

end Construction_Arbre;
```



Affiche_Arbre

Affiche un arbre entré en paramètre.




```
Procedure Affiche(Arbre: in T_Arbre; Alinea: in Integer; C : in out T_Suite )is
begin
    Put("(");
    Put(Arbre.all.Donnee,1);
    Put(")");
    if Est_Feuille(Arbre) then
        Affiche_Caractere(Arbre.all.Cle);
    else
        new_Line;
        for i in 1..Alinea loop
            if Est_Present(C,i) then
                put(" ");
            else
                put("|");
            end if;
            put(7*" ");
        end loop;
        Put("\--0--");
        Supprimer(C,Alinea+1);
        Affiche(Arbre.all.Gauche,Alinea+1,C);
        New_Line;
        for i in 1..Alinea loop
            if Est_Present(C,i) then
                put(" ");
            else
                put("|");
            end if;
            put(7*" ");
        end loop;
        Put("\--1--");
        Enregistrer(C,Alinea+1);
        Affiche(Arbre.all.Droite,Alinea+1,C);
    end if;
end Affiche;

procedure Afficher_Arbre(Arbre:in T_Arbre) is
    C:T_Suite;
begin
    Initialiser(C);
    Affiche(Arbre,0,C);
    Vider(C);
end Afficher_Arbre;
```

Construction_Table

```
--Permet de construire la table à partir de l'arbre.
procedure Construction_Table(Arbre: in T_Arbre;Table:out T_Table) is
    procedure Construire_table(Arbre: in T_Arbre;Table:out T_Table;Chaine:in Unbounded_String) is
    begin
        if Est_Feuille(Arbre) then
            Enregistrer(Table, Arbre.all.Cle, Chaine);
        else
            Construire_table(arbre.all.gauche,Table,Chaine & "0");
            Construire_table(arbre.all.droite,Table,Chaine & "1");
        end if;
    end construire_table;
begin
    Construire_table(Arbre,Table,To_Unbounded_String("")) ;
end Construction_Table;
```

CONCLUSION

Notre approche du problème ainsi que les solutions que nous avons apportées nous ont permis d'obtenir des programmes capables de compresser et décompresser de simples fichiers (ex : *.txt*). Une amélioration éventuelle permettrait d'appliquer ces programmes à d'autres types de fichier (ex : *.jpeg* ou *.mp3*).