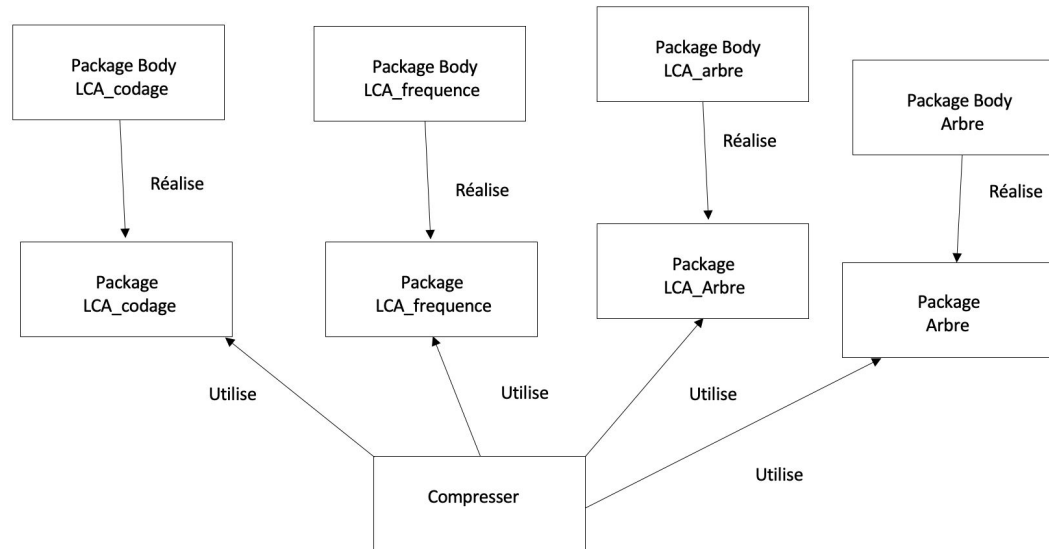




Présentation Projet 2021-2021

Leo Meissner et Steven Zheng

Architecture en modules: Compresser





Type de données

```
package Arbre is
  new arbre_bin_projet (T_Cle => Unbounded_String);
use Arbre;

package LCA_frequence is
  new LCA_projet (T_Cle => Unbounded_String, T_Donnee => Integer);
use LCA_frequence;

package LCA_arbre is
  new LCA_projet (T_Cle => Unbounded_String, T_Donnee => T_Abr);
use LCA_arbre;

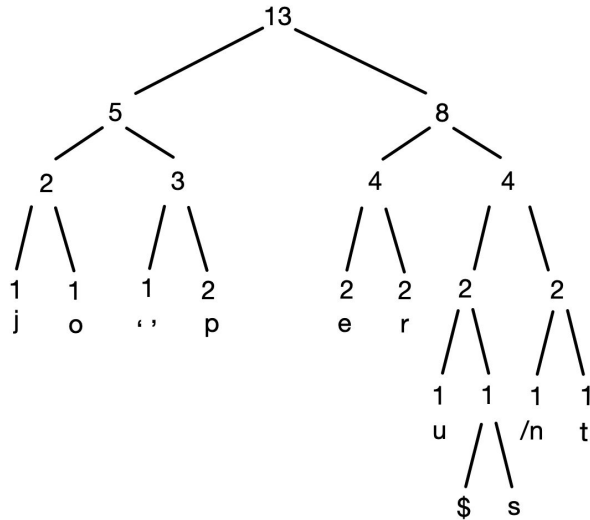
package LCA_codage is
  new LCA_projet (T_Cle => Unbounded_String, T_Donnee => Unbounded_String);
use LCA_arbre;
```



Table de fréquence

\$	s	u	p	e	r	“ ”	o	j	t	/n
1	1	1	2	2	2	1	1	1	1	1

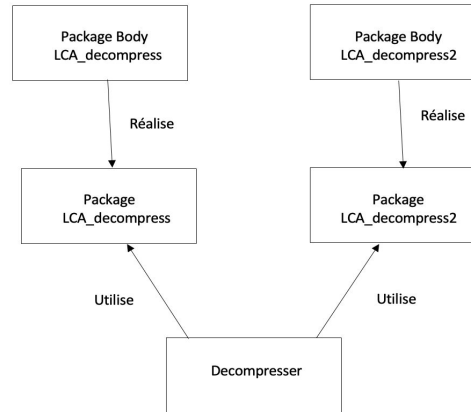
Arbre et table de Huffman



\$	s	u	p	e	r	“ ”	o	j	t	/n
11010	11011	1100	011	100	101	010	001	000	1111	1110

Parcours infixe : 000110110011001011011

Architecture en modules: Décompresser





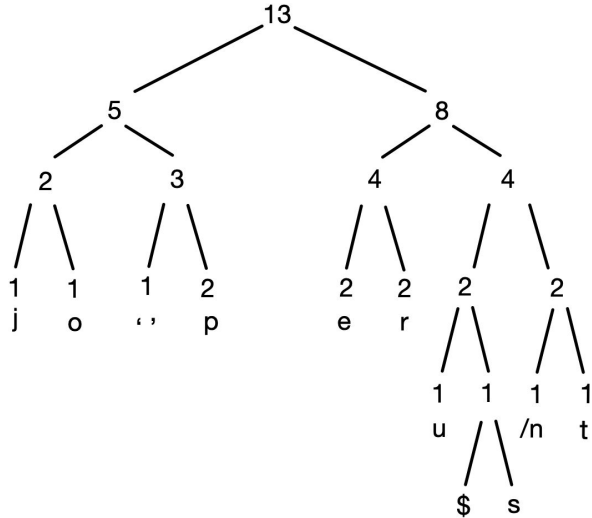
Types de données :

```
package LCA_decompress is
    new LCA_projet (T_Cle => Integer, T_Donnee => Unbounded_String);
use LCA_decompress;

package LCA_decompress2 is
    new LCA_projet (T_Cle => Unbounded_String, T_Donnee => T_Octet);
use LCA_decompress2;
```

Recréation de la table de codage à partir du parcours infixé:

Parcours infixé de l'arbre : 00011011001100101101



j	o	“ ”	p	e	r	u	/\$	s	/n	t
000	001	010	011	100	101	1100	11010	11011	1110	1111



Insertion des symboles dans le fichier décompressé :

Exemple avec 4 octet : 000110110011001011101000

j	o	“ ”	p	e	r	u	/\$	s	/n	t
000	001	010	011	100	101	1100	11010	11011	1110	1111