

Projet d'ALGAV

Adan Bougherara Vivien Demeulenaere

16 Décembre 2021

- 1 Structure de données
- 2 Compression_bdd
 - Implémentation
 - Analyse de complexité
- 3 Expérimentations
 - ROBDD de 1 à 4 variables
 - ROBDD de 5 à 10 variables
- 4 Fusion de deux ROBDD

```
class abd:

    def __init__ (self, etiquette=None, faux=None,
vrai=None):

        self.etiquette = etiquette
        self.faux = faux
        self.vrai = vrai
```

Compression_bdd

Implémentation

```
def compression_bdd (arbre):  
    def aux (arbre, noeuds, racine=False):  
        if arbre == None:  
            return None  
        if arbre.faux == None and arbre.vrai == None:  
            feuille = trouver (arbre.etiquette, noeuds)  
            if not feuille:  
                nouv_feuille = abd (arbre.etiquette)  
                noeuds.append (nouv_feuille)  
                return nouv_feuille  
            return feuille  
        noeud = trouver (arbre.etiquette, noeuds)  
        if not noeud:  
            faux = aux (arbre.faux, noeuds)  
            vrai = aux (arbre.vrai, noeuds)  
            nouv_faux = est_inutile (faux)  
            nouv_vrai = est_inutile (vrai)
```

Compression_bdd

Implémentation

```
    if nouv_faux:
        faux = nouv_faux
    if nouv_vrai:
        vrai = nouv_vrai
    nouv_arbre = abd (arbre.etiquette, faux, vrai)
    if racine and est_inutile(nouv_arbre):
        return faux
    noeuds.append (nouv_arbre)
    return nouv_arbre
return noeud
```

```
res = aux (arbre, [], True)
renommer (res)
return res
```

- Longueur d'un mot de *Lukasiewicz* l_h :

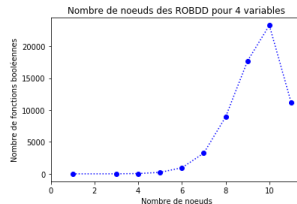
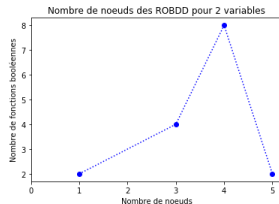
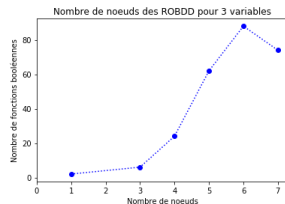
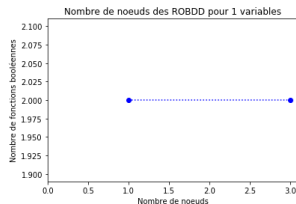
$$\begin{aligned} l_h &= 2.(2^h - 1) + 5.2^h + 4.(2^h - 1) \\ &= 11.2^h - 6 \end{aligned}$$

- Complexité au pire cas en nombre de comparaisons de caractères :

$$C = \sum_{i=0}^{h-1} (l_i \cdot 2^{h-i}) + 5 \cdot 2^h = (11.h - 1).2^h + 6$$

Expérimentations

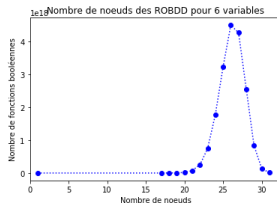
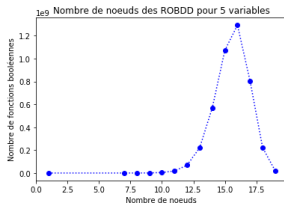
ROBDD de 1 à 4 variables



Expérimentations

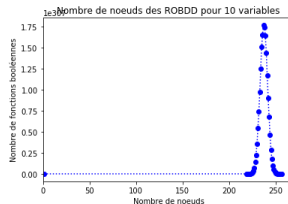
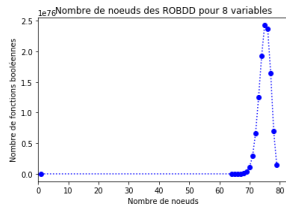
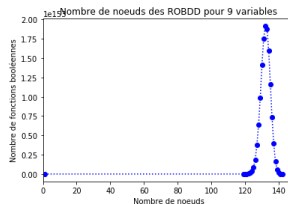
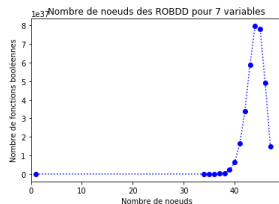
ROBDD de 5 à 10 variables

$$c = \frac{\text{nombre de ROBDD possibles}}{\text{taille de l'échantillon}}$$



Expérimentations

ROBDD de 5 à 10 variables



Expérimentations

ROBDD de 5 à 10 variables

Nb. variables	Nb. arbres	Nb. tailles uniques	Temps de calcul (en s)	Temps de calcul par ROBDD (en s)
5	100000	14	81.85602951	0.00081856
6	100000	16	87.1204493	0.000871204
7	100000	15	114.6158905	0.001146159
8	100000	17	194.0332394	0.001940332
9	100000	25	384.8936143	0.003848936
10	100000	38	821.0664122	0.008210664

- Fusion :

$$\alpha \diamond \alpha' = \begin{cases} (v, l \diamond l', h \diamond h') & \text{si } v = v' \\ (v, l \diamond \alpha', h \diamond \alpha') & \text{si } v < v' \\ (v', \alpha \diamond l', \alpha \diamond h') & \text{si } v > v' \end{cases}$$

- Simplification et réduction :

- Réunification des nombres binaires à partir de l'opérateur booléen
- Suppression des noeuds doublons