

Devoir de Programmation



1 Présentation

2 Arbre de décision

- Compression
- ROBDD
- Complexité

3 Étude expérimentale

4 Pour aller plus loin...

Langage de programmation utilisé: Python

Plateforme de travail: Replit

Diaporama, Rapport: Overleaf

Algorithm 1 Pseudo-code compression

```
compression(arbre, dict) :  
if enfant  $\neq$  null then  
  if  $\neg$ (enfant in dict) then  
    dict[étiquette de l'enfant]  $\leftarrow$  enfant  
    compression(enfant, dict)  
  else  
    enfant  $\leftarrow$  dict[étiquette de l'enfant]  
  end if  
end if
```

Arbre de décision - Compression

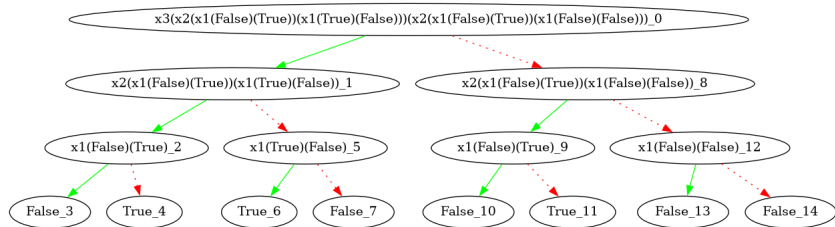


Figure 1: L'arbre avant compression

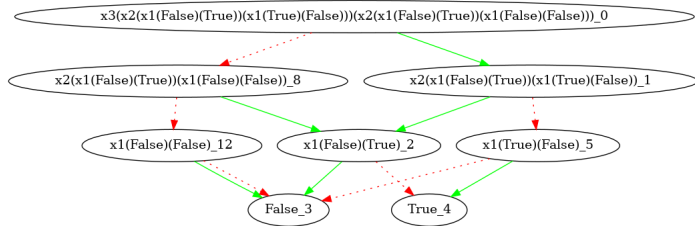


Figure 2: L'arbre après compression

Algorithm 2 Pseudo-code `compression_bdd`

```
compression_bdd(arbre) :
if enfant  $\neq$  null then
  if (enfant_gauche  $\neq$  null) and (enfant_droite  $\neq$  null) then
    if étiquette_enfant_gauche = étiquette_enfant_droite then
      arbre  $\leftarrow$  enfant_gauche
      compression_bdd(arbre)
    else
      compression_bdd(enfant_gauche)
      compression_bdd(enfant_droite)
    end if
  end if
end if
```

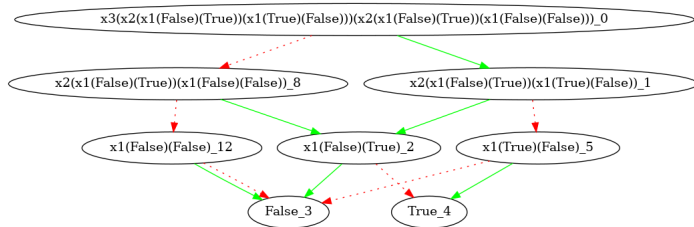


Figure 3: L'arbre avant compression_bdd

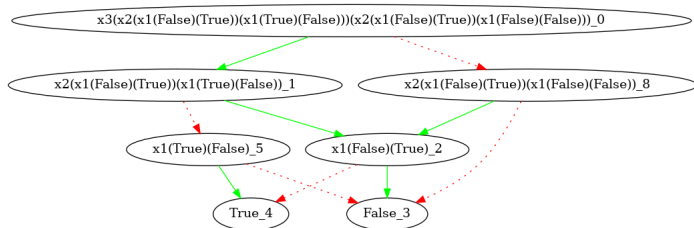


Figure 4: L'arbre après compression_bdd

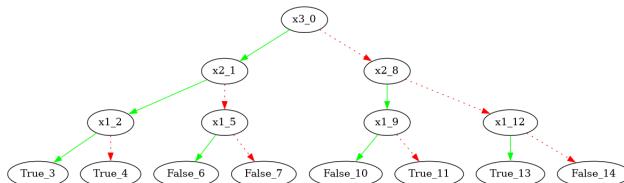


Figure 5: Arbre de décision issu de `table(int('01100011', 2), 8)`

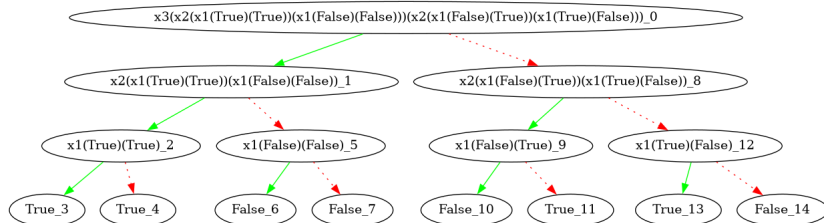


Figure 6: Arbre de décision qui à chaque nœud associe le mot de Lukasiewicz enrichi

- Passage en parcourant avec un parcours suffixe donc complexité en $O(\text{nombre noeuds}) = O(2^{h+1} - 1)$

Arbre de décision - Complexité

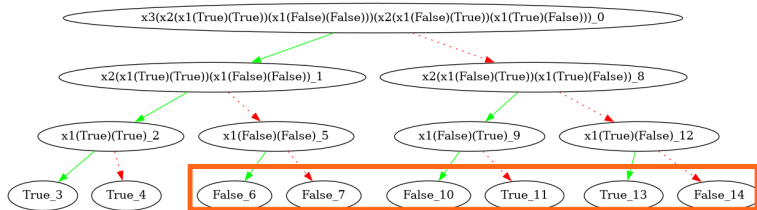


Figure 7: Arbre de décision qui à chaque nœud associe le mot de Lukasiewicz enrichi

-Supposition de l'utilisation d'un tableau avec comparaison naïve = $O(l_h) = O(2^h)$ comparaisons

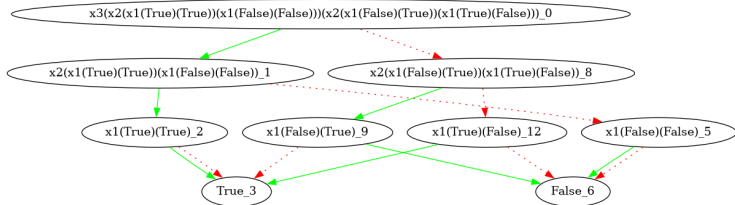


Figure 8: DAG obtenu par compression de l'arbre précédent avec Terminal et Merging Rule

$$-O((2^h - 1 + 2) * l_h) = O(2^{2h})$$

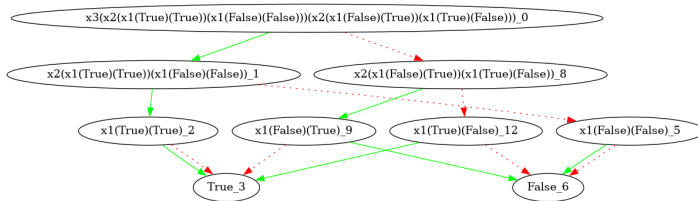


Figure 9: DAG obtenu par compression de l'arbre précédent avec Terminal et Merging Rule

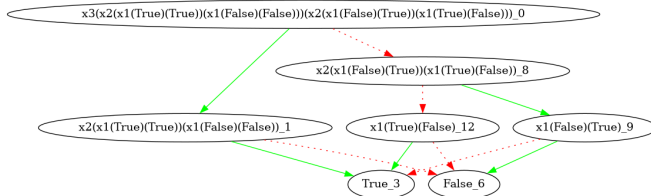
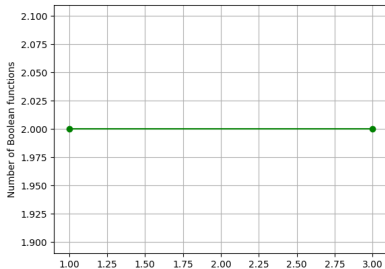


Figure 10: ROBDD associé en appliquant Deletion Rule

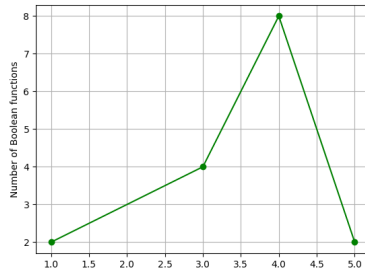
- Nombre de noeuds internes ($2^h - 1$) comparaisons
- Complexité en $O(2^h - 1) * l_h = O(2^{2h})$

- Complexité en nombre de comparaisons pour le passage de l'arbre de décision au ROBDD:
 - $O((2^{h+1} - 1) + 2^{2h} + 2^{2h}) = O(2 * 2^{2h}) = O(2^{2h})$

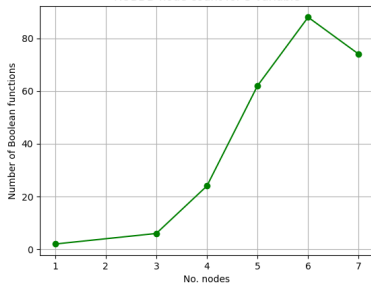
ROBDD node count for 1 variable



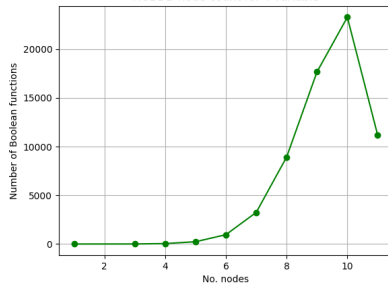
ROBDD node count for 2 variable



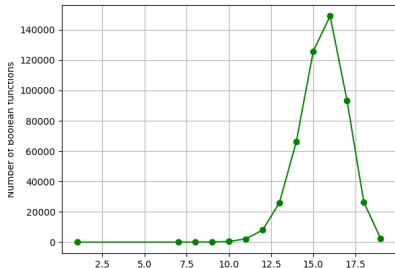
ROBDD node count for 3 variable



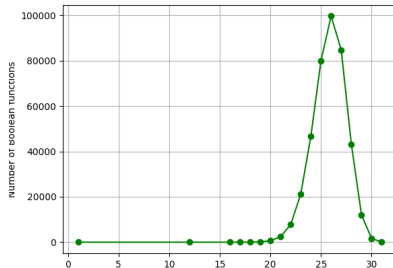
ROBDD node count for 4 variable



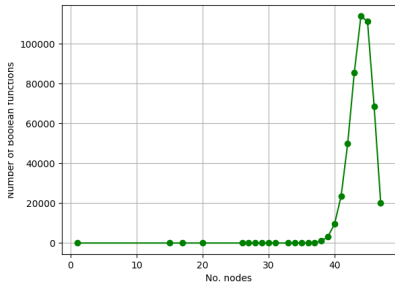
ROBDD node count for 5 variable



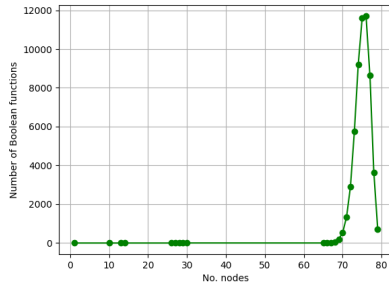
ROBDD node count for 6 variable

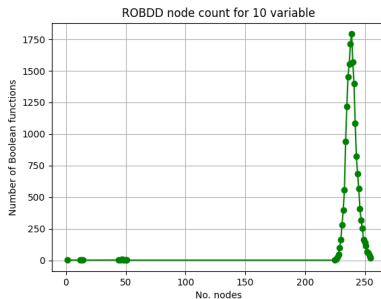
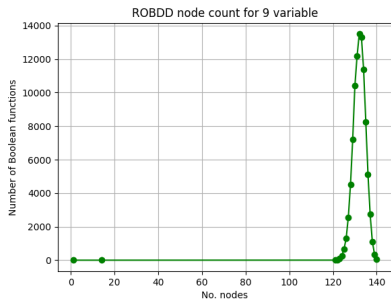


ROBDD node count for 7 variable



ROBDD node count for 8 variable





- Traçage avec un pas de $2^{2^n} \div \text{No. samples de l'article pour la variable } n$.

No. Variables(n)	No. Samples	No. Unique Sizes	Compute Time hh:mm:ss	Seconds per ROBDD
1	4	2	0:00:00	1.98483e-05
2	16	4	0:00:00	0.000135407
3	256	6	0:00:00	0.000134168
4	65536	10	0:00:04	5.57667e-05
5	500055	14	0:00:48	9.62368e-05
6	400004	18	0:01:25	0.00021372
7	486893	25	0:03:25	0.000421237
8	56344	24	0:00:52	0.000915054
9	95000	22	0:03:02	0.001913
10	17976	42	0:01:18	0.00436046

Figure 11: Informations relatives aux expérimentations

Pour aller plus loin...

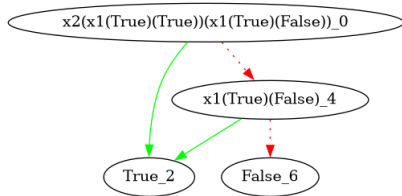


Figure 12: ROBDD de $\text{table}(\text{int}('0111', 2), 4)$

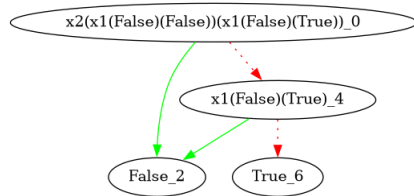


Figure 13: ROBDD de $\text{table}(\text{int}('0111', 2), 4)$

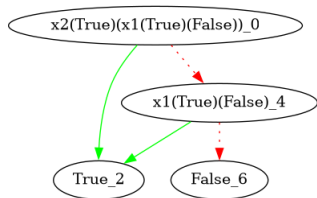


Figure 14: Mots de Lukasiewicz à jour

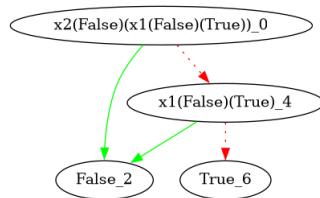


Figure 15: Mots de Lukasiewicz à jour

Pour aller plus loin...

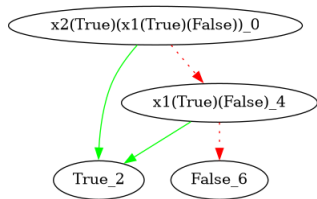


Figure 16: Mots de Lukasiewicz à jour

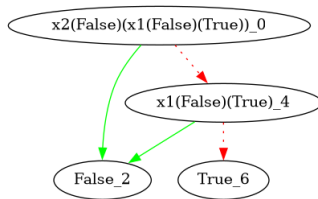


Figure 17: Mots de Lukasiewicz à jour

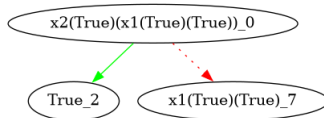


Figure 18: Fusion des deux ROBDD