

Compilation de connaissances et Bornes Inférieures pour les Systèmes de Preuve

Alexis de Colnet



Plan de la présentation

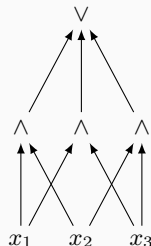
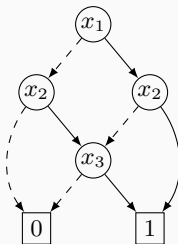
- Compilation de connaissances
- Des fonctions dures à compiler : les formules de Tseitin
- Bornes inférieures pour la résolution régulière
- Bornes inférieures pour les réfutations $\text{OBDD}(\wedge, r)$

Compilation de connaissances

La compilation de “connaissances” est un domaine de l'IA qui étudie les différentes représentations de fonctions (Booléennes).

$$F(x_1, x_2, x_3) = \text{maj}_3(x_1, x_2, x_3) \quad (x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (x_1 \vee x_3)$$

x_1	x_2	x_3	$F(x_1, x_2, x_3)$
0	0	0	0
1	0	0	0
1	1	0	1
0	1	0	0
0	1	1	1
1	1	1	1
1	0	1	1
0	0	1	0



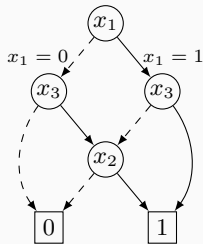
OBDD [Bryant, IEEE Trans. Comp. 1986] Ordered Binary Decision Diagrams

DAG de décision avec deux puits, 0 et 1, et tel que

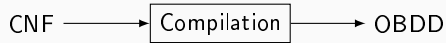
- sur tout chemin de la racine a un puit, chaque variable apparaît au plus une fois
- les variables ont le même ordre d'apparition le long de tous les chemins

L'OBDD est **structuré** par un ordre sur ses variables. Ex :

$$x_1 \prec x_3 \prec x_2$$



Compilation de connaissances



Requêtes sur des CNF :

- satisfiabilité **NP-complet**
- implication clausale **NP-complet**
- nombre de solutions **#P-complet**
- minimisation **Π_2^P -complet**

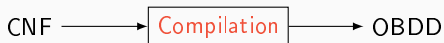
Requêtes sur des OBDD :

- satisfiabilité **linear-time***
- implication clausale **linear-time***
- nombre de solutions **linear-time***
- minimisation **NP-complet**

* : linéaire en la taille de l'OBDD (i.e., son nombre de noeuds)

On veut que la “forme compilée” soit la plus petite possible, et parfois ce n’est pas possible [Bova, Capelli, Mengel et Slivovsky, IJCAI 2016].

Compilation de connaissances



Requêtes sur des CNF :

- satisfiabilité **NP-complet**
- implication clausale **NP-complet**
- nombre de solutions **#P-complet**
- minimisation **Π_2^P -complet**

Requêtes sur des OBDD :

- satisfiabilité **linear-time***
- implication clausale **linear-time***
- nombre de solutions **linear-time***
- minimisation **NP-complet**

* : linéaire en la taille de l'OBDD (i.e., son nombre de noeuds)

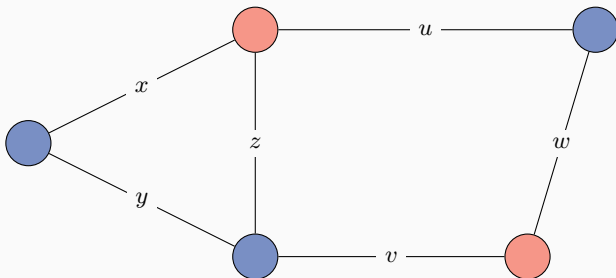
On veut que la “forme compilée” soit la plus petite possible, et parfois ce n’est pas possible [Bova, Capelli, Mengel et Slivovsky, IJCAI 2016].

Plan de la présentation

- ~~Compilation de connaissances~~
- Des fonctions dures à compiler : les formules de Tseitin
- Bornes inférieures pour la résolution régulière
- Bornes inférieures pour les réfutations $\text{OBDD}(\wedge, r)$

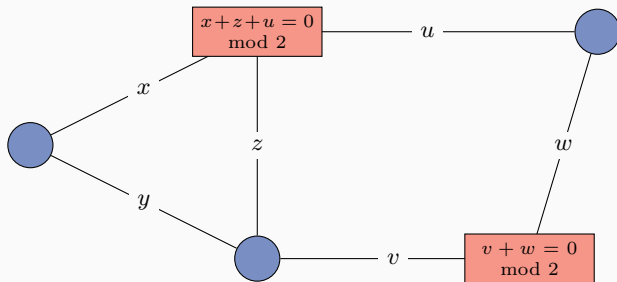
Des fonctions dures à compiler : Les formules de Tseitin

Une formule de Tseitin (nommées après Grigori S. Tseitin [Tseitin, 1968]) est une formule CNF définie avec graphe G et un coloriage c de $V(G)$ avec deux couleurs.



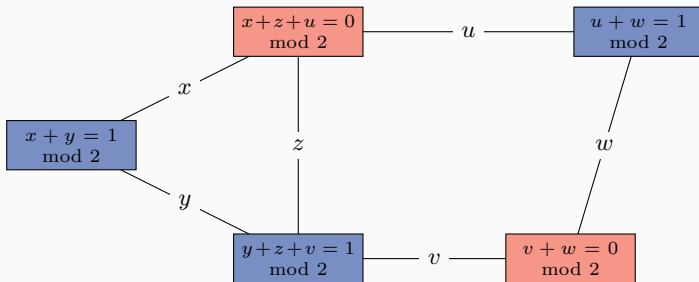
Des fonctions dures à compiler : Les formules de Tseitin

Une formule de Tseitin (nommées après Grigori S. Tseitin [Tseitin, 1968]) est une formule CNF définie avec graphe G et un coloriage c de $V(G)$ avec deux couleurs.



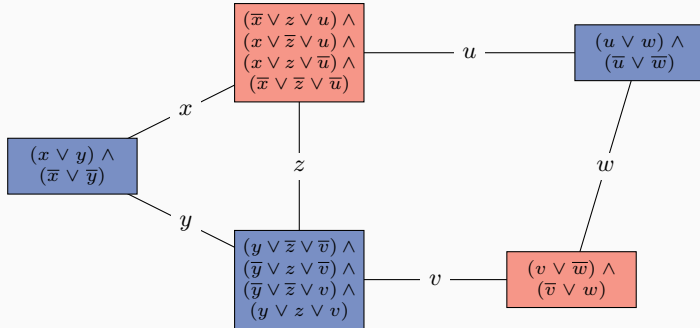
Des fonctions dures à compiler : Les formules de Tseitin

Une formule de Tseitin (nommées après Grigori S. Tseitin [Tseitin, 1968]) est une formule CNF définie avec graphe G et un coloriage c de $V(G)$ avec deux couleurs.



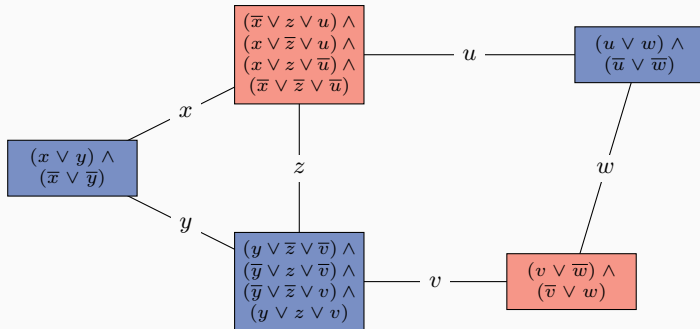
Des fonctions dures à compiler : Les formules de Tseitin

Une formule de Tseitin (nommées après Grigori S. Tseitin [Tseitin, 1968]) est une formule CNF définie avec graphe G et un coloriage c de $V(G)$ avec deux couleurs.



Des fonctions dures à compiler : Les formules de Tseitin

Une formule de Tseitin (nommées après Grigori S. Tseitin [Tseitin, 1968]) est une formule CNF définie avec graphe G et un coloriage c de $V(G)$ avec deux couleurs.



- $T(G, c)$ est la CNF obtenues en conjoignant les CNF des différents vertex.
- Il y a un critère simple pour construire des formules de Tseitin satisfiables ou non-satisfiable pour un graphe G donné.
- Si le degré de G est constant, alors $T(G, c)$ is “petite”.

Les formules de Tseitin sont dures pour les OBDD

[Itsykson, Riazanov, Sagunov and Smirnov, 2021]

Il existe une constante $\alpha > 0$ telle que toute représentation OBDD d'une formule de Tseitin $T(G, c)$ satisfiable est de taille $2^{\alpha tw(G)/\Delta(G)}$, où $tw(G)$ est la treewidth de G et $\Delta(G)$ est le degré maximum du graphe.

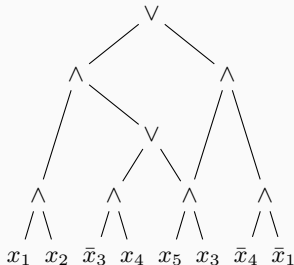
Beaucoup de formules sont “dures” pour les OBDD (i.e., requierent des OBDD de taille exponentielle). On peut regarder la taille de représentation dans d'autres langages.

Circuits DNNF [Darwiche, IJCAI 1999]

Decomposable Negation Normal Form

Circuits Booléens avec les propriétés suivantes :

- les entrées sont des littéraux x , \bar{x}
- les portes sont des \wedge (ET) et des \vee (OU) (pas de négation !)
- chaque porte \wedge est **décomposable** : $g = g_1 \wedge g_2 \wedge \dots \wedge g_m$ vérifie $\text{var}(g_i) \cap \text{var}(g_j) = \emptyset$ (pour tout $i \neq j$).



Les circuits DNNF sont généralement exponentiellement plus petits que les OBDD (et jamais beaucoup plus gros).

Les formules de Tseitin sont dures pour les DNNF

[Itsykson, Razianov, Sagunov and Smirnov, 2021]

Il existe une constante $\alpha > 0$ telle que toute représentation OBDD d'une formule de Tseitin $T(G, c)$ satisfiable est de taille $2^{\alpha tw(G)/\Delta(G)}$, où $tw(G)$ est la treewidth de G et $\Delta(G)$ est le degré maximum du graphe.

[de C et Mengel, SAT 2021, JAIR 2023]

Il existe une constante $\alpha > 0$ telle que toute représentation DNNF d'une formule de Tseitin $T(G, c)$ satisfiable est de taille $2^{\alpha tw(G)/\Delta(G)}$, où $tw(G)$ est la treewidth de G et $\Delta(G)$ est le degré maximum du graphe.

Ce résultat est utilisé pour prouver deux bornes inférieures pour les formules de Tseitin **non satisfiable** dans deux systèmes de preuve.

Plan de la présentation

- ~~Compilation de connaissances~~
- ~~Des fonctions dures à compiler : les formules de Tseitin~~
- Bornes inférieures pour la résolution régulière
- Bornes inférieures pour les réfutations $\text{OBDD}(\wedge, r)$

Réfutation par résolution

Inférence par résolution :
$$\frac{C_1 \vee x \quad C_2 \vee \bar{x}}{C_1 \vee C_2} \quad \text{résolution sur } x$$

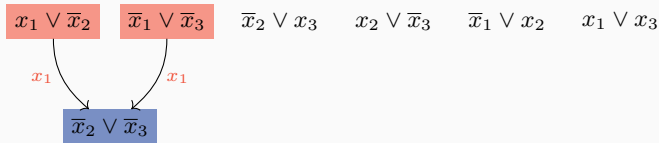
$$x_1 \vee \bar{x}_2 \quad \bar{x}_1 \vee \bar{x}_3 \quad \bar{x}_2 \vee x_3 \quad x_2 \vee \bar{x}_3 \quad \bar{x}_1 \vee x_2 \quad x_1 \vee x_3$$

Réfutation par résolution

Inférence par résolution :

$$\frac{x_1 \vee \bar{x}_2 \quad \bar{x}_1 \vee \bar{x}_3}{\bar{x}_2 \vee \bar{x}_3}$$

résolution sur x_1



Réfutation par résolution

Inférence par résolution :

$$\frac{\boxed{\bar{x}_1 \vee x_2} \quad \boxed{x_1 \vee x_3}}{\boxed{x_2 \vee x_3}}$$

résolution sur x_1

$$\begin{array}{ccc} x_1 \vee \bar{x}_2 & & \bar{x}_1 \vee \bar{x}_3 \\ \swarrow x_1 & & \searrow x_1 \\ & \bar{x}_2 \vee \bar{x}_3 \end{array}$$

$$\bar{x}_2 \vee x_3$$

$$x_2 \vee \bar{x}_3$$

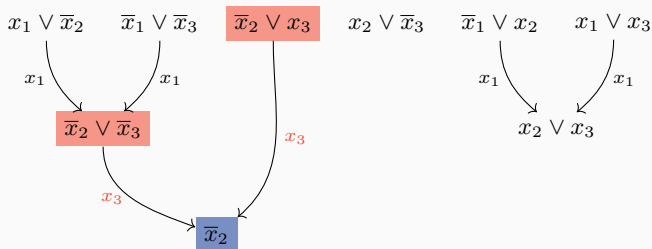
$$\begin{array}{ccc} \boxed{\bar{x}_1 \vee x_2} & & \boxed{x_1 \vee x_3} \\ \swarrow x_1 & & \searrow x_1 \\ & \boxed{x_2 \vee x_3} \end{array}$$

Réfutation par résolution

Inférence par résolution :

$$\frac{\overline{x}_2 \vee \overline{x}_3 \quad \overline{x}_2 \vee x_3}{\overline{x}_2}$$

résolution sur x_3

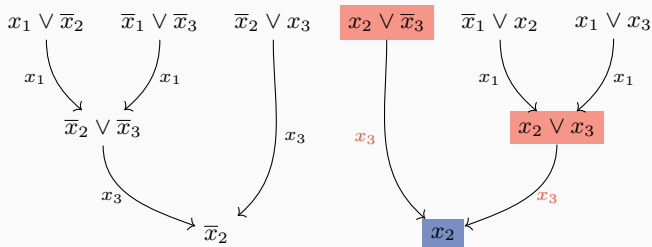


Réfutation par résolution

Inférence par résolution :

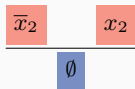
$$\frac{x_2 \vee x_3 \quad x_2 \vee \bar{x}_3}{x_2}$$

résolution sur x_3

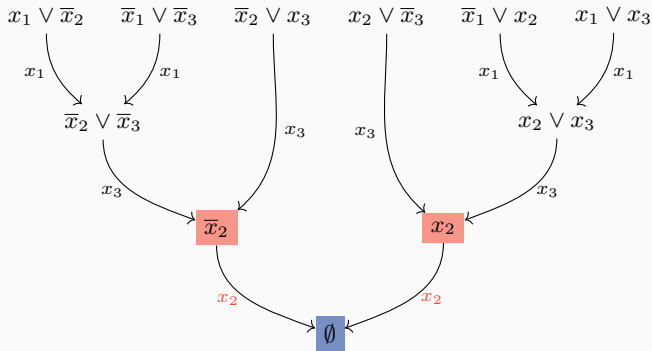


Réfutation par résolution

Inférence par résolution :



résolution sur x_2

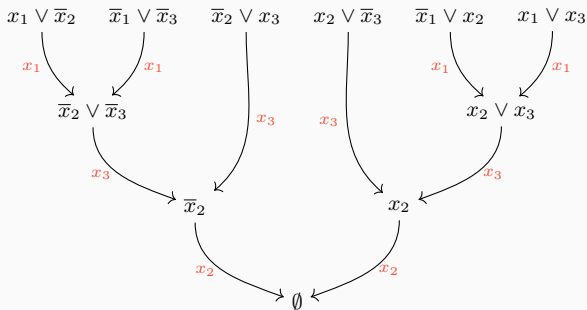


Réfutation par résolution

Le système de preuve par résolution est cohérent et complet (sound and complete).

La longueur d'une réfutation par résolution est le nombre de clauses dans la réfutation.

Réfutation par résolution régulière (RRR) de longueur 11.



La réfutation est dite régulière quand chaque **variables de résolution** apparaît au plus une fois par chemin.

Réfutation par résolution des formules de Tseitin

$T(G, c)$ est une formule de Tseitin non-satisfiable, n est son nombre de variables, Δ est le degré maximal de G .

- Quand G est un expander de degré borné on sait que les réfutations par résolution régulière (RRR) sont de longueur exponentielle en n [Urquhart, 1987]. Que peut-on dire dans le cas général ?

$$2^{\Omega(\frac{tw(G)}{\Delta \log(n)})} \Omega(\text{poly}(\frac{1}{n})) \leq \text{RRR-length of } T(G, c) \leq 2^{O(tw(G))} O(\text{poly}(n))$$

[Itsykson, Razianov, Sagunov and Smirnov 2021]

[Alekhovich and Razborov, 2011]

- La longueur des RRR de formules de Tseitin dont le graphe est de degré borné est **presque complètement** caractérisée par la treewidth.

Réfutation par résolution des formules de Tseitin

$T(G, c)$ est une formule de Tseitin non-satisfiable, n est son nombre de variables, Δ est le degré maximal de G .

- Quand G est un expander de degré borné on sait que les réfutations par résolution régulière (RRR) sont de longueur exponentielle en n [Urquhart, 1987]. Que peut-on dire dans le cas général ?

$$2^{\Omega(\frac{tw(G)}{\Delta})} \Omega(\text{poly}(\frac{1}{n})) \leq \text{RRR-length of } T(G, c) \leq 2^{O(tw(G))} O(\text{poly}(n))$$

[dC and Mengel, 2021]

[Alekhovich and Razborov, 2011]

- La longueur des RRR de formules de Tseitin dont le graphe est de degré borné est presque complètement caractérisée par la treewidth.

Réfutation par résolution des formules de Tseitin

$T(G, c)$ est une formule de Tseitin non-satisfiable, n est son nombre de variables, Δ est le degré maximal de G .

- Quand G est un expander de degré borné on sait que les réfutations par résolution régulière (RRR) sont de longueur exponentielle en n [Urquhart, 1987]. Que peut-on dire dans le cas général ?

$$2^{\Omega(tw(G))} \Omega(poly(\frac{1}{n})) \leq \text{RRR-length of } T(G, c) \leq 2^{O(tw(G))} O(poly(n))$$

[Itsykson, Razianov and Smirnov, 2022]

[Alekhovich and Razborov, 2011]

- La longueur des RRR de formules de Tseitin ~~dont le graphe est de degré borné~~ est ~~presque~~ complètement caractérisée par la treewidth.

Proof sketch

Ancienne borne
[Itsykson et al. 2021]:

$$\text{RRR-length of } T(G, c) \geq 2^{\Omega(\frac{tw(G)}{\log(n)})} \Omega(\text{poly}(\frac{1}{n}))$$

Proof sketch
($T(G, c)$ est unsat.,
 $T(G, c^*)$ est sat.)

$$\text{RRR-length of } T(G, c) = \text{FBDD-size of SearchClause}(T(G, c))$$

$$\geq \text{FBDD-size of SearchVertex}(T(G, c))$$

$$\geq \left(\text{FBDD-size of } T(G, c^*) \right)^{\frac{1}{\log(n)}}$$

$$\geq \left(2^{\Omega(tw(G))} \right)^{\frac{1}{\log(n)}}$$

Proof sketch

Nouvelle borne

[dC and Mengel, 2021]:

$$\text{RRR-length of } T(G, c) \geq 2^{\Omega(tw(G))} \Omega(\text{poly}(\frac{1}{n}))$$

Proof sketch

$(T(G, c)$ est unsat.,
 $T(G, c^*)$ est sat.)

$$\text{RRR-length of } T(G, c) = \text{FBDD-size of SearchClause}(T(G, c))$$

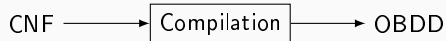
$$\geq \text{FBDD-size of SearchVertex}(T(G, c))$$

$$\geq \text{DNF-size of } T(G, c^*)$$

$$\geq 2^{\Omega(tw(G))} \times \frac{1}{n}$$

Plan de la présentation

- ~~Compilation de connaissances~~
- ~~Des fonctions dures à compiler : les formules de Tseitin~~
- ~~Bornes inférieures pour la résolution régulière~~
- Bornes inférieures pour les réfutations $\text{OBDD}(\wedge, r)$



Deux familles de compilateurs :

- **Top-down** : C2D, miniC2D, D4, DSharp, (\approx) SharpSAT-TD
- **Bottom-up** : SDD package, CUDD package, SALADD

En compilation top-down, la forme compilée est la trace d'un algorithme de recherche de type DPLL+キャッシング+components analysis.

La compilation bottom-up peut être directement décrite en utilisant des systèmes de preuve.

Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

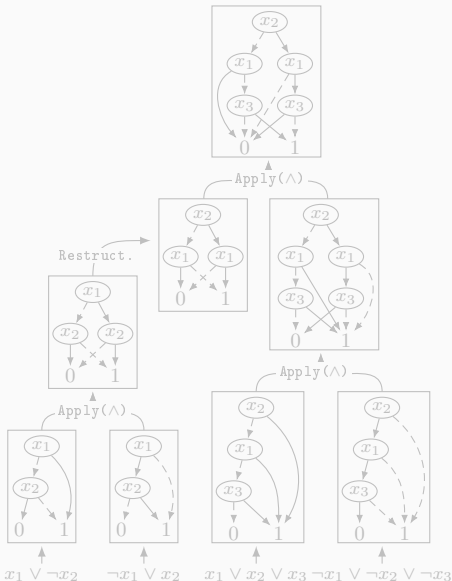
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

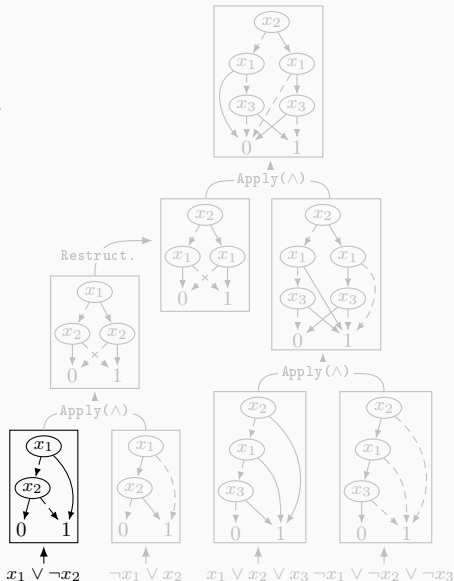
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

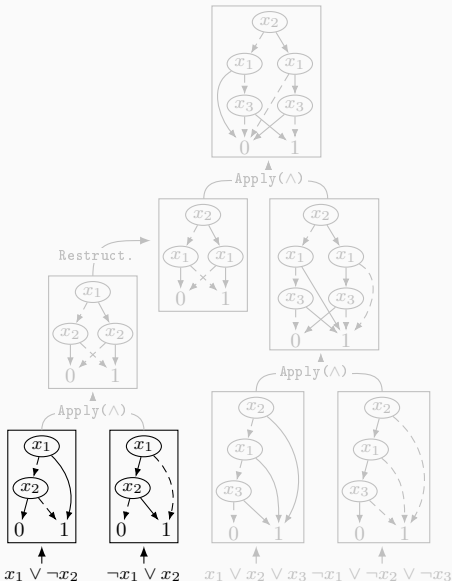
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

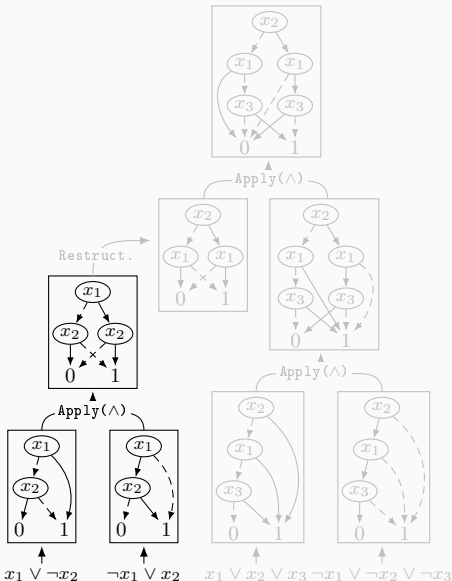
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

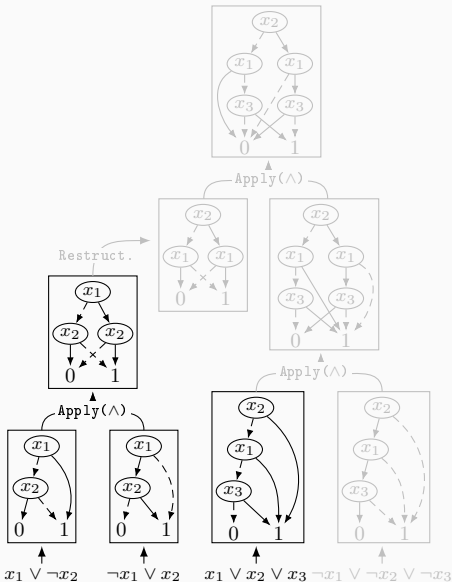
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

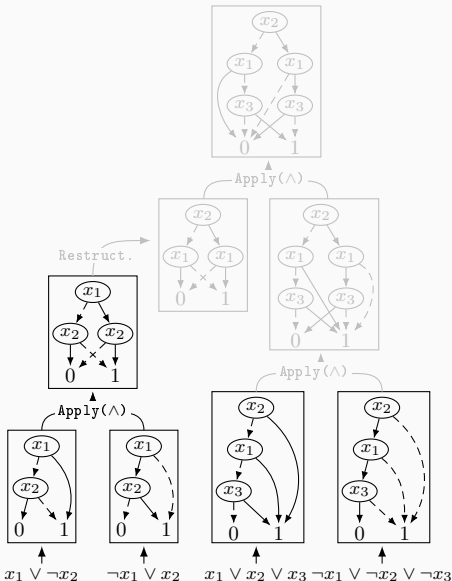
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

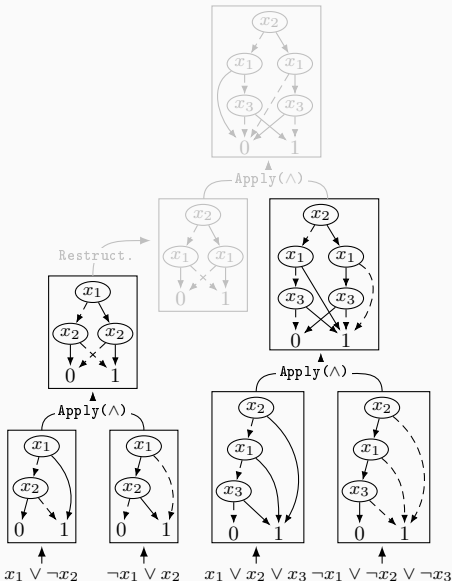
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

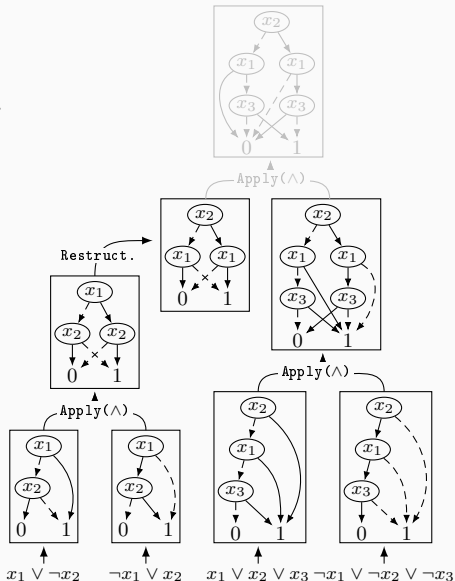
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Compilation bottom-up de la formule

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

vers OBDD.

$$B_1 = \text{Compile}(x_1 \vee \neg x_2)$$

$$B_2 = \text{Compile}(\neg x_1 \vee x_2)$$

$$B_3 = \text{Apply}(B_1, B_2, \wedge)$$

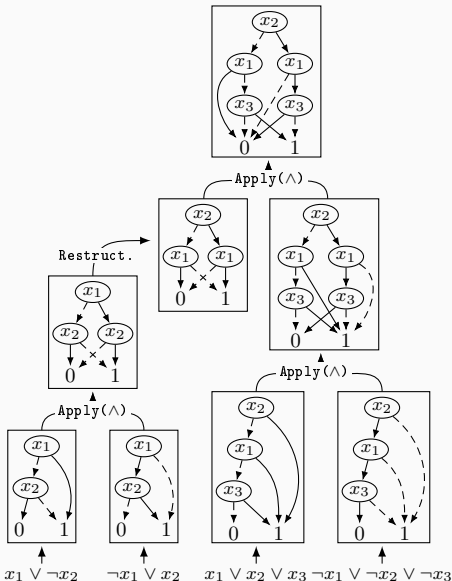
$$B_4 = \text{Compile}(x_1 \vee x_2 \vee x_3)$$

$$B_5 = \text{Compile}(\neg x_1 \vee \neg x_2 \vee \neg x_3)$$

$$B_6 = \text{Apply}(B_4, B_5, \wedge)$$

$$B_7 = \text{Restructure}(B_3)$$

$$B_8 = \text{Apply}(B_6, B_7, \wedge)$$



Compilation Bottom-Up OBDD(\wedge, r)

Une compilation Bottom-Up OBDD(\wedge, r) d'un système F de contraintes est une séquence

$$(B_1, I_1), (B_2, I_2), \dots, (B_N, I_N)$$

où les B_i sont des OBDD, où $F \equiv B_N$ et où les I_i sont des instructions telles que, pour tout $i \in [N]$

- $I_i : B_i = \text{Compile}(C)$ où C est une contrainte de F et $B_i \equiv C$; ou
- $I_i : B_i = \text{Restructure}(B_j)$ où $j < i$ et $B_i \equiv B_j$; ou
- $I_i : B_i = \text{Apply}(B_j, B_k, \wedge)$ où $j, k < i$ et $B_i \equiv B_j \wedge B_k$ et B_i, B_j, B_k ont le même ordre sur leur variables.

Compilation Bottom-Up OBDD(\wedge, r)

Une compilation Bottom-Up OBDD(\wedge, r) d'un système F de contraintes est une séquence

$$(B_1, I_1), (B_2, I_2), \dots, (B_N, I_N)$$

où les B_i sont des OBDD, où $F \equiv B_N$ et où les I_i sont des instructions telles que, pour tout $i \in [N]$

- $I_i : B_i = \text{Compile}(C)$ où C est une contrainte de F et $B_i \equiv C$; ou
- $I_i : B_i = \text{Restructure}(B_j)$ où $j < i$ et $B_i \equiv B_j$; ou
- $I_i : B_i = \text{Apply}(B_j, B_k, \wedge)$ où $j, k < i$ et $B_i \equiv B_j \wedge B_k$ et B_i, B_j, B_k ont le même ordre sur leur variables.

Compilation Bottom-Up OBDD(\wedge, r)

Une compilation Bottom-Up OBDD(\wedge, r) d'un système F de contraintes est une séquence

$$(B_1, I_1), (B_2, I_2), \dots, (B_N, I_N)$$

où les B_i sont des OBDD, où $F \equiv B_N$ et où les I_i sont des instructions telles que, pour tout $i \in [N]$

- $I_i : B_i = \text{Compile}(C)$ où C est une contrainte de F et $B_i \equiv C$; ou
- $I_i : B_i = \text{Restructure}(B_j)$ où $j < i$ et $B_i \equiv B_j$; ou
- $I_i : B_i = \text{Apply}(B_j, B_k, \wedge)$ où $j, k < i$ et $B_i \equiv B_j \wedge B_k$ et B_i, B_j, B_k ont le même ordre sur leur variables.

Compilation Bottom-Up OBDD(\wedge, r)

Une compilation Bottom-Up OBDD(\wedge, r) d'un système F de contraintes est une séquence

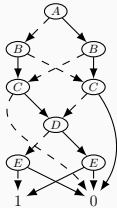
$$(B_1, I_1), (B_2, I_2), \dots, (B_N, I_N)$$

où les B_i sont des OBDD, où $F \equiv B_N$ et où les I_i sont des instructions telles que, pour tout $i \in [N]$

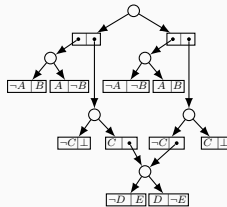
- $I_i : B_i = \text{Compile}(C)$ où C est une contrainte de F et $B_i \equiv C$; ou
- $I_i : B_i = \text{Restructure}(B_j)$ où $j < i$ et $B_i \equiv B_j$; ou
- $I_i : B_i = \text{Apply}(B_j, B_k, \wedge)$ où $j, k < i$ et $B_i \equiv B_j \wedge B_k$ et B_i, B_j, B_k ont le même ordre sur leur variables.

Compilation Bottom-Up $\text{SDD}(\wedge, r)$, $\text{SDNNF}(\wedge, r)$

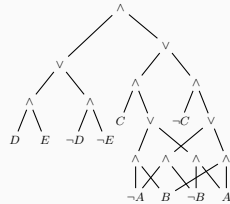
OBDD



SDD



SDNNF



Les circuits SDNNF [Pipatsrisawat and Darwiche, AAI 2008] sont plus compacts que les SDD [Darwiche, IJCAI 2011], qui sont plus compacts que les OBDD.

Les SDNNF, SDD et OBDD sont des circuits/diagrammes structurés pour lesquels on a des procédures $\text{Apply}(\cdot, \cdot, \wedge)$ quadratiques.

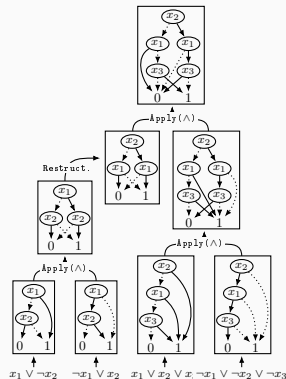
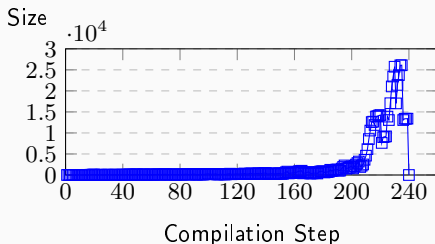
La compil. bottom-up dans ces langages est possible.

Bornes Inférieures pour les Réfutations OBDD(\wedge, r)

Un problème majeur de la compilation bottom-up est la taille des circuits intermédiaires. On peut avoir

$$\max_{1 \leq i \leq N} |B_i| \gg \max(|B_N|, |F|)$$

Dans le cas extrême F est non-satisfiable. Alors OBDD(\wedge, r) et OBDD(\wedge) sont des systèmes de preuve (au sens de Cook et Reckow).



Bornes Inférieures pour les Réfutations OBDD(\wedge, r)

- Explosion de la taille des résultats intermédiaires observée expérimentalement pour OBDD(\wedge).
[Narodytska et Walsh, IJCAI 2007]
- Bornes exponentielles pour des réfutations dans OBDD(\wedge).
[Krajíček, J. Symb. Log. 2008; Segerlind, CCC 2008 ; Järvisalo, CP 2011]
- Bornes exponentielles pour la réfutation dans OBDD $_{\prec}(\wedge)$ de CNF aléatoires pour un ordre \prec prédéfini et fixe des variables.
[Friedman et Xu, CSR 2013]
- Bornes exponentielles pour la réfutation dans OBDD(\wedge, r) du pigeon-hole-principle.
[Itsykson, Knop, Romashchenko et Sokolov, J. Symb. Log, 2020]
- Bornes exponentielles pour la réfutation dans OBDD(\wedge, r) de certaines formules de Tseitin.
[Itsykson, Knop, Romashchenko et Sokolov, J. Symb. Log, 2020]

Bornes Inférieures pour les Réfutations OBDD(\wedge, r)

- Explosion de la taille des résultats intermédiaires observée expérimentalement pour OBDD(\wedge).
[Narodytska et Walsh, IJCAI 2007]
- Bornes exponentielles pour des réfutations dans OBDD(\wedge).
[Krajíček, J. Symb. Log. 2008; Segerlind, CCC 2008 ; Jarvisalo, CP 2011]
- Bornes exponentielles pour la réfutation dans OBDD $_{\prec}(\wedge)$ de CNF aléatoires pour un ordre \prec prédéfini et fixe des variables.
[Friedman et Xu, CSR 2013]
- Bornes exponentielles pour la réfutation dans OBDD(\wedge, r) du pigeon-hole-principle.
[Itsykson, Knop, Romashchenko et Sokolov, J. Symb. Log, 2020]
- Bornes exponentielles pour la réfutation dans OBDD(\wedge, r) de certaines formules de Tseitin.
[Itsykson, Knop, Romashchenko et Sokolov, J. Symb. Log, 2020]

[Itsykson, Knop, Romashchenko et Sokolov, J. Symb. Log, 2020]

Toute compilation/réfutation OBDD(\wedge, r) de $T(G, c)$ (non-satisfiable) où G est de degré $\Delta(G) = O(1)$ et de “spectral expansion” constante génère des résultats intermédiaires de taille $2^{\Omega(n)}$.

Bornes Inférieures pour les Réfutations OBDD(\wedge, r)

[Itsykson, Knop, Romashchenko et Sokolov, J. Symb. Log, 2020]

Toute compilation/réfutation OBDD(\wedge, r) de $T(G, c)$ (non-satisfiable) où G est de degré $\Delta(G) = O(1)$ et de “spectral expansion” constante génère des résultats intermédiaires de taille $2^{\Omega(n)}$.



[de C et Mengel, AAAI 2022]

Toute compilation/réfutation SDNNF(\wedge, r) de $T(G, c)$ (non-satisfiable) où $\Delta(G) = O(1)$ génère des résultats intermédiaires de taille $2^{\Omega(tw(G))}$.

Bornes Inférieures pour les Réfutations OBDD(\wedge, r)

[Itsykson, Knop, Romashchenko et Sokolov, J. Symb. Log, 2020]

Toute compilation/réfutation OBDD(\wedge, r) de $T(G, c)$ (non-satisfiable) où G est de degré $\Delta(G) = O(1)$ et de “spectral expansion” constante génère des résultats intermédiaires de taille $2^{\Omega(n)}$.



[de C et Mengel, AAAI 2022]

Toute compilation/réfutation SDNNF(\wedge, r) de $T(G, c)$ (non-satisfiable) où $\Delta(G) = O(1)$ génère des résultats intermédiaires de taille $2^{\Omega(tw(G))}$.



[Itsykson, Riazanov et Smirnov, SAT 2022]

Toute compilation/réfutation SDNNF(\wedge, r) de $T(G, c)$ (non-satisfiable) où $\Delta(G) = O(1)$ génère des résultats intermédiaires de taille $2^{\Omega(tw(G))}$.

Bornes Inférieures pour les Réfutations OBDD(\wedge, r)

[de C et Mengel, AAAI 2022]

Toute compilation SDNNF(\wedge, r) de formules de Tseitin non-satisfiables $T(G, c)$ où $\Delta(G) = O(1)$ génère des résultats intermédiaires de taille $2^{\Omega(tw(G))}$.

$T(G, c)$ satisfiable

\Rightarrow si $\Delta(G) = O(1)$ alors tous les circuits SDNNF qui représentent $T(G, c)$ sont de taille $2^{\Omega(tw(G))}$

[de C et Mengel, SAT 2021]

$T(G, c)$ non-satisfiable

On regarde le dernier Apply de sa compilation SDNNF(\wedge, r)

$$\text{Apply}(D, D', \wedge) \equiv T(G, c) \equiv 0$$

On utilise D et D' pour construire un circuit D^* tel que

- D^* est un circuit SDNNF qui représente $T(G', c')$ satisfiable et,
- G' est un sous graphe de G avec $tw(G') = \Omega(tw(G))$ et,
- $|D^*| \leq O(|D| \cdot |D'|)$.

Plan de la présentation

- ~~Compilation de connaissances~~
- ~~Des fonctions dures à compiler : les formules de Tseitin~~
- ~~Bornes inférieures pour la résolution régulière~~
- ~~Bornes inférieures pour les réfutations OBDD(\wedge, r)~~