Logique

SAT : satisfaction de formules booléennes

Thomas Pietrzak
Licence Informatique

Université de Lille 1 Sciences et Technologies

Clauses de Horn

Prolog

Fait : a

But : $b_1, \ldots, b_n$

Clause : a :== $c_1, \ldots, c_n$

Logique

Fait : a

But : $b_1 \wedge \ldots \wedge b_n$

Clause : $c_1 \wedge \ldots \wedge c_n \Rightarrow a \equiv \neg c_1 \vee \ldots \vee \neg c_n \vee a$

Motivation

Trouver des variables qui satisfont une formule = résoudre un problème

Peu de variables : facile à faire la table de vérité.

Beaucoup de variables : exponentiel.

SAT

Soit φ une formule : existe-t-il une valuation v telle que $[\![\varphi]\!]_v = 1$?

FNC

Forme normale conjonctive (FNC) : conjonction de disjonctions

$$(\varphi_{1,1} \vee \ldots \varphi_{1,n_1}) \wedge \cdots \wedge (\varphi_{k,1} \vee \ldots \varphi_{k,n_1})$$

Littéral : variable propositionnelle ou négation de variable propositionnelle

Classes de complexité

Classe NP

Il existe un algorithme **non-déterministe** résolvant ce problème en temps polynomial.

Classe NP-complet

Ce problème appartient à NP.

Tous les problèmes de la classe NP se réduisent à ce problème en temps polynomial.

Théorème de Cook

SAT est NP-complet

Problèmes NP-complets

Tous les problèmes NP-complets sont équivalents à SAT.

En résolvant SAT on résout tous les autres problèmes.

3-SAT

SAT avec clauses de 3 variables

Équivalence entre SAT et 3-SAT

$(a_1 \vee a_2 \vee \ldots \vee a_n) \equiv (a_1 \vee a_2 \vee b_1) \wedge (a_3 \vee \neg b_1 \vee b_2) \wedge \ldots \wedge (a_{n-1} \vee a_n \vee \neg b_{n-3})$

n-SAT

SAT avec clauses de n variables

Équivalence entre SAT et n-SAT

$(a_1 \vee a_2 \vee \ldots \vee a_m) \equiv (a_1 \vee a_2 \vee \ldots \vee a_{n-1} \vee b_1) \wedge$

$(a_{n+1} \vee \ldots \vee a2_{n-3} \vee \neg b_1 \vee b_2) \wedge \ldots \wedge (a_{m-n+1} \vee \ldots \vee a_m \vee \neg b_k)$

2-SAT

SAT avec clauses de 2 variables

Pas NP-complet (P)

Résolution SAT

Table de vérité

2 possibilités par variable booléenne

Valuations à calculer : au plus 2^n

300 variables : 2^{300}, soit plus que le nombre d'atomes dans l'univers ($\simeq 10^{80} \simeq 2^{266}$)

Il va falloir simplifier…

Simplifications

$(a \vee a \vee \ldots) \equiv (a \vee \ldots)$: suppression des occurrences multiples

$(a \vee \neg a \vee \ldots) \equiv \top$: suppression des clauses contenant des opposés

C’est bien, mais avec ça on n’ira pas loin…

Propagation unitaire

Clause unaire : a ou $\neg a$

$a \wedge \neg a \wedge \ldots \equiv \bot$

$a \wedge (\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge R \equiv (b_1 \vee \ldots \vee b_n) \wedge R$ et $v(a) = \top$

$\neg a \wedge (\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge R \equiv (c_1 \vee \ldots \vee c_n) \wedge R$ et $v(a) = \bot$

Élimination des littéraux purs

Littéral pur : littéral qui est soit toujours positif, soir toujours négatif

$(a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge (d_1 \vee \ldots \vee d_n) \wedge R \equiv (d_1 \vee \ldots \vee d_n) \wedge R$ et $v(a) = \top$

$(\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (\neg a \vee c_1 \vee \ldots \vee c_n) \wedge (d_1 \vee \ldots \vee d_n) \wedge R \equiv (d_1 \vee \ldots \vee d_n) \wedge R$ et $v(a) = \bot$

Davis-Putnam (DP)

Résultante

$c_1 = (a \vee b_1 \vee b_2 \vee \ldots \vee b_n)$ $c_2 = (\neg a \vee d_1 \vee d_2 \vee \ldots \vee d_n)$

Résultante : $r = (b_1 \vee b_2 \vee \ldots \vee b_n \vee d_1 \vee d_2 \vee \ldots \vee d_n)$

$c_1 \wedge c_2$ satisfiable ssi r satisfiable

Démonstration $\Rightarrow$

Soit $a = \top$: c_2 satisfiable et $\neg a = \bot$ ➔ $d_1 \vee d_2 \vee \ldots \vee d_n$ satisfiable, donc r aussi

Soit $a = \bot$: c_1 satisfiable et $a = \bot$ ➔ $b_1 \vee b_2 \vee \ldots \vee b_n$ satisfiable, donc r aussi

Exemple

$(a \lor b) \land (a \lor \neg c) \land (\neg a \lor c)$

Il faut factoriser a

Exemple

$(a \vee b) \wedge (a \vee \neg c) \wedge (\neg a \vee c)$

$\equiv (a \vee (b \wedge \neg c)) \wedge (\neg a \vee c)$

Exemple

$(a \vee b) \wedge (a \vee \neg c) \wedge (\neg a \vee c)$

$\equiv (a \vee (b \wedge \neg c)) \wedge (\neg a \vee c)$

$\equiv (b \wedge \neg c) \vee c$

On calcule la résultante

Exemple

$(a \vee b) \wedge (a \vee \neg c) \wedge (\neg a \vee c)$

$\equiv (a \vee (b \wedge \neg c)) \wedge (\neg a \vee c)$

$\equiv (b \wedge \neg c) \vee c$

$\equiv (b \vee c) \wedge (\neg c \vee c)$

On remet en FNC

Exemple

$(a \vee b) \wedge (a \vee \neg c) \wedge (\neg a \vee c)$

$\equiv (a \vee (b \wedge \neg c)) \wedge (\neg a \vee c)$

$\equiv (b \wedge \neg c) \vee c$

$\equiv (b \vee c) \wedge (\neg c \vee c)$

$\equiv b \vee c$

Algorithme DP

1. **Éliminer les clauses unitaires tant qu'il y en a**

   $a \wedge \neg a \wedge R \equiv \bot$ ➔ Formule non satisfiable

   $a \wedge (\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge R$ ➔ $(b_1 \vee \ldots \vee b_n) \wedge R$

   $\neg a \wedge (\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge R$ ➔ $(c_1 \vee \ldots \vee c_n) \wedge R$

   Formule vide ➔ Formule satisfiable

2. **Éliminer les littéraux purs**

   $(a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge (d_1 \vee \ldots \vee d_n) \wedge R$ ➔ $(d_1 \vee \ldots \vee d_n) \wedge R$

   $(\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (\neg a \vee c_1 \vee \ldots \vee c_n) \wedge (d_1 \vee \ldots \vee d_n) \wedge R$ ➔ $(d_1 \vee \ldots \vee d_n) \wedge R$

3. **On simplifie les résultantes**

   $(a \vee b_1 \vee b_2 \vee \ldots \vee b_n) \wedge (\neg a \vee d_1 \vee d_2 \vee \ldots \vee d_n) \wedge R$

   ➔ $(b_1 \vee b_2 \vee \ldots \vee b_n \vee d_1 \vee d_2 \vee \ldots \vee d_n) \wedge R$

Exemple

$(a \lor \neg b \lor c \lor \neg d \lor f) \wedge (\neg b \lor \neg c \lor \neg d \lor e) \wedge (\neg b \lor \neg c \lor d \lor \neg f) \wedge (\neg a \lor \neg d) \wedge (b \lor c \lor d \lor \neg e)$

Exemple

$(a \lor \neg b \lor c \lor \neg d \lor f) \land (\neg b \lor \neg c \lor \neg d \lor e) \land (\neg b \lor \neg c \lor d \lor \neg f) \land (\neg a \lor e) \land (b \lor c \lor d \lor \neg e)$

➔ $(\neg b \lor c \lor \neg d \lor f \lor e) \land (\neg b \lor \neg c \lor \neg d \lor e) \land (\neg b \lor \neg c \lor d \lor \neg f) \land \neg d \land (b \lor c \lor d \lor \neg e)$

Exemple

$(a \vee \neg b \vee c \vee \neg d \vee f) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge (\neg a \vee e) \wedge (b \vee c \vee d \vee \neg e)$

➔ $(\neg b \vee c \vee \neg d \vee f \vee e) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge \neg d \wedge (b \vee c \vee d \vee \neg e)$

➔ $(\neg b \vee \neg c \vee \neg f) \wedge (b \vee c \vee \neg e)$

Exemple

$(a \vee \neg b \vee c \vee \neg d \vee f) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge (\neg a \vee e) \wedge (b \vee c \vee d \vee \neg e)$

➔ $(\neg b \vee c \vee \neg d \vee f \vee e) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge \neg d \wedge (b \vee c \vee d \vee \neg e)$

➔ $(\neg b \vee \neg c \vee \neg f) \wedge (b \vee c \vee \neg e)$

➔ $\neg b \vee \neg c \vee \neg e$

Exemple

$(a \vee \neg b \vee c \vee \neg d \vee f) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge (\neg a \vee e) \wedge (b \vee c \vee d \vee \neg e)$

➔ $(\neg b \vee c \vee \neg d \vee f \vee e) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge \neg d \wedge (b \vee c \vee d \vee \neg e)$

➔ $(\neg b \vee \neg c \vee \neg f) \wedge (b \vee c \vee \neg e)$

➔ $\neg b \vee \neg c \vee \neg e$

➔ $\top$

Algorithme DP

On n'obtient pas la valuation qui prouve la satisfiabilité.

Problème : simplification des résultantes

$(a \vee b_1 \vee b_2 \vee \ldots \vee b_n) \wedge (\neg a \vee d_1 \vee d_2 \vee \ldots \vee d_n) \equiv b_1 \vee b_2 \vee \ldots \vee b_n \vee d_1 \vee d_2 \vee \ldots \vee d_n$

Que vaut a ?

Besoin d'un algorithme constructif.

Davis–Putnam–Logemann–Loveland (DPLL)

Valuation partielle

F[x/$\perp$] est la formule F dans laquelle x est évaluée à $\perp$

F[x/$\top$] est la formule F dans laquelle x est évaluée à $\top$

F est satisfaisable ssi F[x/$\perp$] est satisfaisable ou F[x/$\top$] est satisfaisable.

Propriétés

$a[a/\top] = \top$

$a[a/\bot] = \bot$

$\neg a[a/\top] = \bot$

$\neg a[a/\bot] = \top$

$(a \vee b_1 \vee b_2 \vee \ldots \vee b_n)\ [a/\top] = \top$

$(a \vee b_1 \vee b_2 \vee \ldots \vee b_n)[a/\bot] = (b_1 \vee b_2 \vee \ldots \vee b_n)$

$(\neg a \vee b_1 \vee b_2 \vee \ldots \vee b_n)[a/\top] = (b_1 \vee b_2 \vee \ldots \vee b_n)$

$(\neg a \vee b_1 \vee b_2 \vee \ldots \vee b_n)[a/\bot] = \top$

Alternative à résultante

$F = (a \vee b_1 \vee b_2 \vee \ldots \vee b_n) \wedge (\neg a \vee d_1 \vee d_2 \vee \ldots \vee d_n) \wedge R$

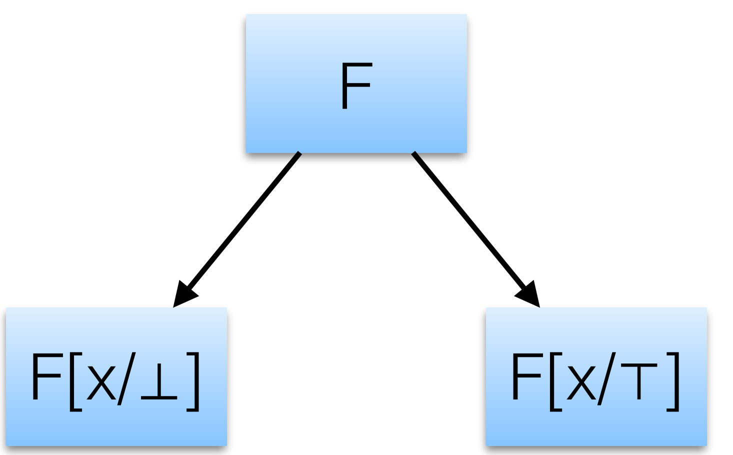
Recherche par cas

$a = \top$ ➔ F satisfaisable ssi $(d_1 \vee d_2 \vee \ldots \vee d_n) \wedge R$ est satisfaisable

$a = \bot$ ➔ F satisfaisable ssi $(b_1 \vee b_2 \vee \ldots \vee b_n) \wedge R$ est satisfaisable

Arbre de recherche

x est la **variable pivot**



Clauses unitaires

a clause unitaire dans $F \Rightarrow F$ satisfiable ssi $F[x/\top]$ satisfiable.

$\neg a$ clause unitaire dans $F \Rightarrow F$ satisfiable ssi $F[x/\bot]$ satisfiable.

$$(a \wedge (\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n))[a/\top] \equiv (b_1 \vee \ldots \vee b_n)[a/\top]$$

$$(\neg a \wedge (\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n))[a/\bot] \equiv (c_1 \vee \ldots \vee c_n)[a/\bot]$$

Élimination des littéraux purs

a présent et ¬a jamais présent dans F ⇒ F satisfiable ssi F[x/⊤] satisfiable.

¬a présent et a jamais présent dans F ⇒ F satisfiable ssi F[x/⊥] satisfiable.

$$((a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge (d_1 \vee \ldots \vee d_n))[x/\top] = (d_1 \vee \ldots \vee d_n)[x/\top]$$

$$((\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (\neg a \vee c_1 \vee \ldots \vee c_n) \wedge (d_1 \vee \ldots \vee d_n))[x/\bot] = (d_1 \vee \ldots \vee d_n)[x/\bot]$$

Algorithme DPLL

1. **Éliminer les clauses unitaires tant qu'il y en a**

   $a \wedge \neg a \wedge \ldots \equiv \bot$ ➔ Formule non satisfiable

   $a \wedge (\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge R$ ➔ $((b_1 \vee \ldots \vee b_n) \wedge R)[a/\top]$

   $\neg a \wedge (\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge R$ ➔ $((c_1 \vee \ldots \vee c_n) \wedge R)[a/\bot]$

   Formule vide ➔ Formule satisfiable

2. **Éliminer les littéraux purs**

   $(a \vee b_1 \vee \ldots \vee b_n) \wedge (a \vee c_1 \vee \ldots \vee c_n) \wedge (d_1 \vee \ldots \vee d_n) \wedge R$ ➔ $((d_1 \vee \ldots \vee d_n) \wedge R)[a/\top]$

   $(\neg a \vee b_1 \vee \ldots \vee b_n) \wedge (\neg a \vee c_1 \vee \ldots \vee c_n) \wedge (d_1 \vee \ldots \vee d_n) \wedge R$ ➔ $((d_1 \vee \ldots \vee d_n) \wedge R)[a/\bot]$

3. **On simplifie les résultantes**

   $(a \vee b_1 \vee b_2 \vee \ldots \vee b_n) \wedge (\neg a \vee d_1 \vee d_2 \vee \ldots \vee d_n) \wedge R$

   ➔ $((b_1 \vee b_2 \vee \ldots \vee b_n) \wedge R)[a/\bot]$

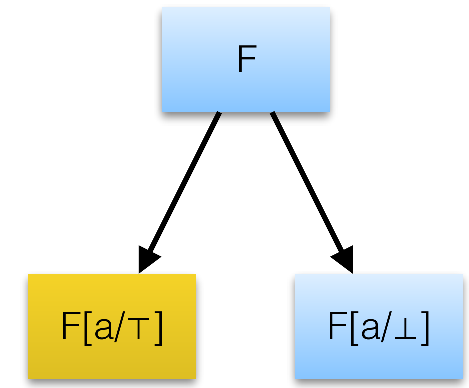
   ➔ $((d_1 \vee d_2 \vee \ldots \vee d_n) \wedge R)[a/\top]$

F

Exemple

$$(a \vee \neg b \vee c \vee \neg d \vee f) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge (\neg a \vee \neg d) \wedge (b \vee c \vee d \vee \neg e)$$

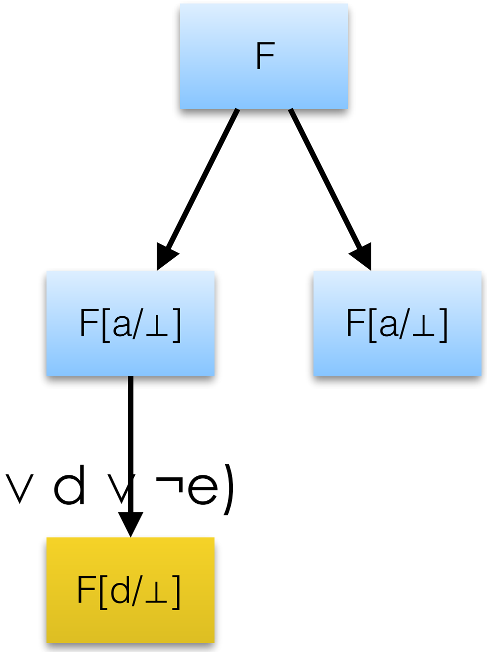
Exemple



$(a \vee \neg b \vee c \vee \neg d \vee f) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge (\neg a \vee \neg d) \wedge (b \vee c \vee d \vee \neg e)$

➔ $(\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge \neg d \wedge (b \vee c \vee d \vee \neg e)[a/\top]$

Exemple

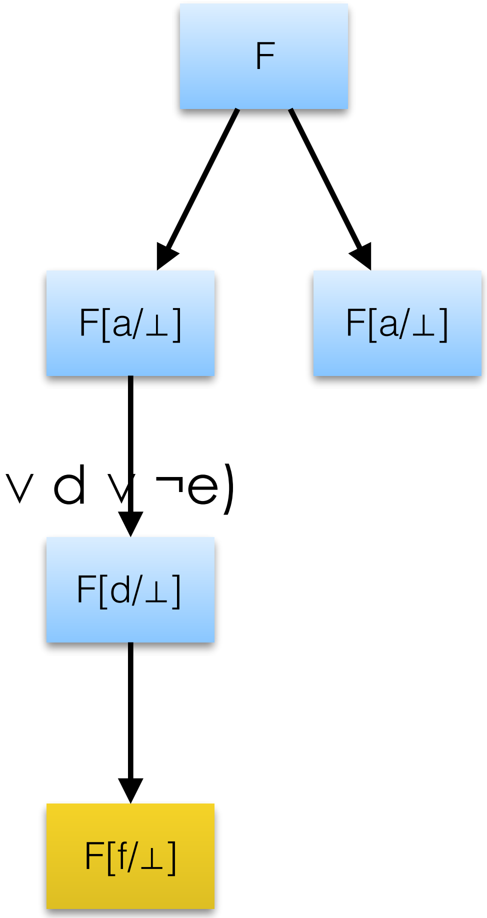


$(a \vee \neg b \vee c \vee \neg d \vee f) \wedge (\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge (\neg a \vee \neg d) \wedge (b \vee c \vee d \vee \neg e)$

➔ $(\neg b \vee \neg c \vee \neg d \vee e) \wedge (\neg b \vee \neg c \vee d \vee \neg f) \wedge \neg d \wedge (b \vee c \vee d \vee \neg e)[a/\top]$

➔ $(\neg b \vee \neg c \vee \neg f) \wedge (b \vee c \vee \neg e)[a/\top][d/\bot]$

Exemple



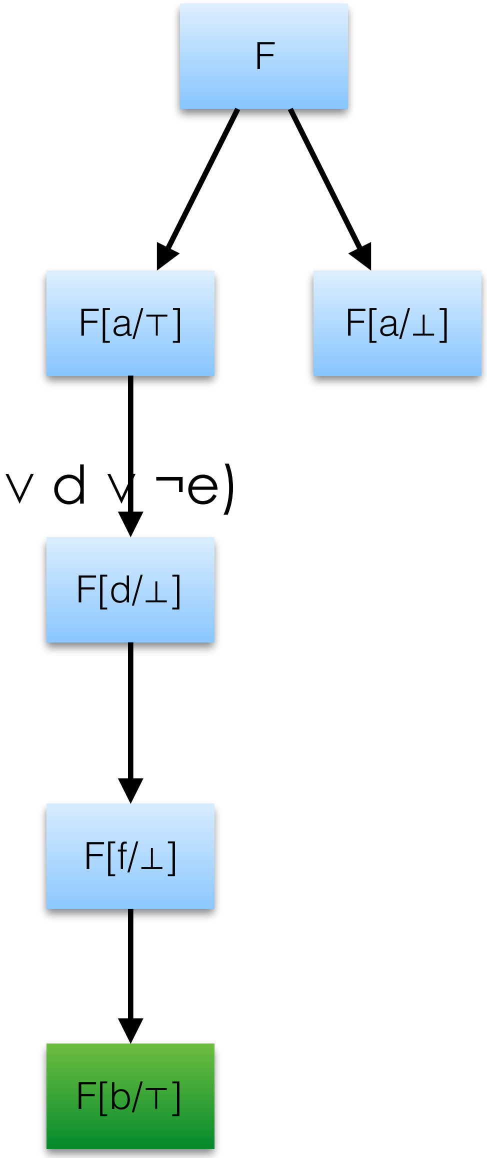
$(a \lor \neg b \lor c \lor \neg d \lor f) \land (\neg b \lor \neg c \lor \neg d \lor e) \land (\neg b \lor \neg c \lor d \lor \neg f) \land (\neg a \lor \neg d) \land (b \lor c \lor d \lor \neg e)$

➔ $(\neg b \lor \neg c \lor \neg d \lor e) \land (\neg b \lor \neg c \lor d \lor \neg f) \land \neg d \land (b \lor c \lor d \lor \neg e)[a/\top]$

➔ $(\neg b \lor \neg c \lor \neg f) \land (b \lor c \lor \neg e)[a/\top][d/\bot]$

➔ $(b \lor c \lor \neg e)[a/\top][d/\bot][f/\bot]$

Exemple



$(a \lor \neg b \lor c \lor \neg d \lor f) \land (\neg b \lor \neg c \lor \neg d \lor e) \land (\neg b \lor \neg c \lor d \lor \neg f) \land (\neg a \lor \neg d) \land (b \lor c \lor d \lor \neg e)$

➔ $(\neg b \lor \neg c \lor \neg d \lor e) \land (\neg b \lor \neg c \lor d \lor \neg f) \land \neg d \land (b \lor c \lor d \lor \neg e)[a/\top]$

➔ $(\neg b \lor \neg c \lor \neg f) \land (b \lor c \lor \neg e)[a/\top][d/\bot]$

➔ $(b \lor c \lor \neg e)[a/\top][d/\bot][f/\bot]$

➔ $\top[a/\top][d/\bot][f/\bot][b/\top]$