Algorithmes et structures de données pour la vérification formelle

SAT : la satisfaction Booléenne

Gérard Berry

Collège de France
Chaire Algorithmes, machines et langages
gerard.berry@college-de-france.fr

Cours 3, 16/03/2016

Suivi du séminaire de Laurent Simon (LABRI)

COLLÈGE DE FRANCE
1530

Agenda

1. Le calcul Booléen (rappels)
2. La forme normale conjonctive (CNF)
3. Exemples de problèmes SAT
4. De la résolution à Davis and Putnam
5. CDCL = Conflict Driven Clause Learning
6. Two Literal Watching
7. Conclusion

Agenda

1. Le calcul Booléen (rappels)
2. La forme normale conjonctive (CNF)
3. Exemples de problèmes SAT
4. De la résolution à Davis and Putnam
5. CDCL = Conflict Driven Clause Learning
6. Two Literal Watching
7. Conclusion

Lois de la conjonction et de la disjonction

vrai	*vrai*, 1, 1
faux	*faux*, 0, 0
variables	$x, y, x_1, y_1, \ldots$
formules	$A, B, C, \ldots$
conjonction (et)	$A \wedge B$
disjonction (ou)	$A \vee B$
associativité	$A \wedge (B \wedge C) = (A \wedge B) \wedge C$
	$A \vee (B \vee C) = (A \vee B) \vee C$
commutativité	$A \wedge B = B \wedge A$
	$A \vee B = B \vee A$
idempotence	$A \wedge A = A$
	$A \vee A = A$
éléments neutres	$A \wedge 1 = 1 \wedge A = A$
	$A \vee 0 = 0 \vee A = A$
éléments absorbants	$A \wedge 0 = 0 \wedge A = 0$
	$A \vee 1 = 1 \vee A = 1$
distributivité	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
	$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

Lois de la négation

négation (non)	$\neg A$
inversion	$\neg 0 = 1$
	$\neg 1 = 0$
double négation	$\neg(\neg A) = A$
	$A \wedge \neg A = 0$
tiers exclu	$A \vee \neg A = 1$
de Morgan	$\neg(A \wedge B) = \neg A \vee \neg B$
	$\neg(A \vee B) = \neg A \wedge \neg B$

Forme normale négative (NNF)

littéral positif	x, y, x_1, y_1
littéral négatif	$\bar{x} = \neg x, \overline{x_1} = \neg x_1, \ldots$
formule en NNF	les négations n'apparaissent que dans les littéraux
exemple	$(\bar{x} \vee y) \wedge (\bar{y} \vee (z \wedge \bar{t}))$
mise en NNF	simple application récursive de De Morgan

Le problème SAT

- SAT : étant donnée une formule $A(x_1,x_2, ..., x_n)$,
  existe-t-il des valeurs Booléennes x_i des x_i qui rendent A vraie ?

- VALID : étant donnée une formule $A(x_1,x_2, ..., x_n)$,
  A est-elle vraie pour toutes les valeurs Booléennes x_i des x_i ?

- Note : VALID(A) $\leftrightarrow$ ¬SAT(¬A)

SAT est un problème fondamental de l'informatique et des mathématiques, avec des applications partout

C'est le prototype du problème NP-complet auquel se réduisent bien d'autres problèmes

Agenda

1. Le calcul Booléen (rappels)
2. La forme normale conjonctive (CNF)
3. Exemples de problèmes SAT
4. De la résolution à Davis and Putnam
5. CDCL = Conflict Driven Clause Learning
6. Two Literal Watching
7. Conclusion

Forme normale conjonctive (CNF)

Clause : disjonction de littéraux

CNF : conjonction de clauses

$$(x \vee \bar{y} \vee z) \wedge (y \vee t \vee \bar{u}) \wedge (\bar{x} \vee u)$$

- Représentation naturelle en contraintes et vérification
  - représentation de connaissances (source historique)
  - model-checking et équivalence de circuits et programmes
  - encodage direct de très nombreux problèmes combinatoires
  - ...
- Toute formule peut être mise en CNF
  - algorithme évident mais exponentiel :
    1. mise en NNF par application répétée de De Morgan
    2. récurrence simple pour $A \wedge B$
    3. distributivité pour $(A_1 \wedge A_2 \wedge ... \wedge A_m) \vee (B_1 \wedge B_2 \wedge ... \wedge B_n)$

Mise en CNF efficace (Tseitin 1968)

Rappel : $A \Rightarrow B =_{def} \neg A \vee B$

$A \Leftrightarrow B =_{def} (\neg A \vee B) \wedge (A \vee \neg B)$

Note : $x \Leftrightarrow (y \wedge z) \rightarrow (\overline{x} \vee y) \wedge (\overline{x} \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$

$x \Leftrightarrow (y \vee z) \rightarrow (\overline{x} \vee y \vee z) \wedge (x \vee \overline{y}) \wedge (x \vee \overline{z})$

1. Mise en NNF

2. Introduction récursive de variables auxiliaires :

$(((a \wedge b) \vee (\overline{a} \wedge \overline{b})) \wedge \overline{c}) \vee d$

Mise en CNF efficace (Tseitin 1968)

Rappel : $A \Rightarrow B =_{def} \neg A \vee B$

$A \Leftrightarrow B =_{def} (\neg A \vee B) \wedge (A \vee \neg B)$

Note : $x \Leftrightarrow (y \wedge z) \rightarrow (\bar{x} \vee y) \wedge (\bar{x} \vee z) \wedge (x \vee \bar{y} \vee \bar{z})$

$x \Leftrightarrow (y \vee z) \rightarrow (\bar{x} \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (x \vee \bar{z})$

1. Mise en NNF

2. Introduction récursive de variables auxiliaires :

$(((a \wedge b) \vee (\bar{a} \wedge \bar{b})) \wedge \bar{c}) \vee d$

$(x \Leftrightarrow (a \wedge b)) \wedge (((x \vee (\bar{a} \wedge \bar{b})) \wedge \bar{c}) \vee d)$

Mise en CNF efficace (Tseitin 1968)

Rappel : $A \Rightarrow B =_{def} \neg A \vee B$

$A \Leftrightarrow B =_{def} (\neg A \vee B) \wedge (A \vee \neg B)$

Note : $x \Leftrightarrow (y \wedge z) \rightarrow (\bar{x} \vee y) \wedge (\bar{x} \vee z) \wedge (x \vee \bar{y} \vee \bar{z})$

$x \Leftrightarrow (y \vee z) \rightarrow (\bar{x} \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (x \vee \bar{z})$

1. Mise en NNF

2. Introduction récursive de variables auxiliaires :

$(((a \wedge b) \vee (\bar{a} \wedge \bar{b})) \wedge \bar{c}) \vee d$

$(x \Leftrightarrow (a \wedge b)) \wedge (((x \vee (\bar{a} \wedge \bar{b})) \wedge \bar{c}) \vee d)$

$(x \Leftrightarrow (a \wedge b)) \wedge (y \Leftrightarrow (\bar{a} \wedge \bar{b})) \wedge (((x \vee y) \wedge \bar{c}) \vee d)$

Mise en CNF efficace (Tseitin 1968)

Rappel : $A \Rightarrow B =_{def} \neg A \vee B$

$A \Leftrightarrow B =_{def} (\neg A \vee B) \wedge (A \vee \neg B)$

Note : $x \Leftrightarrow (y \wedge z) \rightarrow (\bar{x} \vee y) \wedge (\bar{x} \vee z) \wedge (x \vee \bar{y} \vee \bar{z})$

$x \Leftrightarrow (y \vee z) \rightarrow (\bar{x} \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (x \vee \bar{z})$

1. Mise en NNF

2. Introduction récursive de variables auxiliaires :

$(((a \wedge b) \vee (\bar{a} \wedge \bar{b})) \wedge \bar{c}) \vee d$

$(x \Leftrightarrow (a \wedge b)) \wedge (((x \vee (\bar{a} \wedge \bar{b})) \wedge \bar{c}) \vee d)$

$(x \Leftrightarrow (a \wedge b)) \wedge (y \Leftrightarrow (\bar{a} \wedge \bar{b})) \wedge (((x \vee y) \wedge \bar{c}) \vee d)$

$(x \Leftrightarrow (a \wedge b)) \wedge (y \Leftrightarrow (\bar{a} \wedge \bar{b})) \wedge (z \Leftrightarrow (x \vee y)) \wedge ((z \wedge \bar{c}) \vee d)$

Mise en CNF efficace (Tseitin 1968)

Rappel : $A \Rightarrow B =_{def} \neg A \vee B$

$A \Leftrightarrow B =_{def} (\neg A \vee B) \wedge (A \vee \neg B)$

Note : $x \Leftrightarrow (y \wedge z) \rightarrow (\bar{x} \vee y) \wedge (\bar{x} \vee z) \wedge (x \vee \bar{y} \vee \bar{z})$

$x \Leftrightarrow (y \vee z) \rightarrow (\bar{x} \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (x \vee \bar{z})$

1. Mise en NNF

2. Introduction récursive de variables auxiliaires :

$(((a \wedge b) \vee (\bar{a} \wedge \bar{b})) \wedge \bar{c}) \vee d$

$(x \Leftrightarrow (a \wedge b)) \wedge (((x \vee (\bar{a} \wedge \bar{b})) \wedge \bar{c}) \vee d)$

$(x \Leftrightarrow (a \wedge b)) \wedge (y \Leftrightarrow (\bar{a} \wedge \bar{b})) \wedge (((x \vee y) \wedge \bar{c}) \vee d)$

$(x \Leftrightarrow (a \wedge b)) \wedge (y \Leftrightarrow (\bar{a} \wedge \bar{b})) \wedge (z \Leftrightarrow (x \vee y)) \wedge ((z \wedge \bar{c}) \vee d)$

$(x \Leftrightarrow (a \wedge b)) \wedge (y \Leftrightarrow (\bar{a} \wedge \bar{b})) \wedge (z \Leftrightarrow (x \vee y)) \wedge (t \Leftrightarrow (z \vee \bar{c})) \wedge (t \vee d)$

Mise en CNF efficace (Tseitin 1968)

Note : $x \Leftrightarrow (y \wedge z) \rightarrow \ (\bar{x} \vee y) \wedge (\bar{x} \vee z) \wedge (x \vee \bar{y} \vee \bar{z})$

$x \Leftrightarrow (y \vee z) \rightarrow \ (\bar{x} \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (x \vee \bar{z})$

$$(x \Leftrightarrow (a \wedge b)) \wedge (y \Leftrightarrow (\bar{a} \wedge \bar{b})) \wedge (z \Leftrightarrow (x \vee y)) \wedge (t \Leftrightarrow (z \vee \bar{c})) \wedge (t \vee d)$$

$$\begin{aligned} & (\bar{x} \vee a) \wedge (\bar{x} \vee b) \wedge (x \vee \bar{a} \vee \bar{b}) \\ \wedge\ & (\bar{y} \vee \bar{a}) \wedge (\bar{y} \vee \bar{b}) \wedge (y \vee a \vee b) \\ \wedge\ & (\bar{z} \vee x \vee y) \wedge (z \vee \bar{x}) \wedge (z \vee \bar{y}) \\ \wedge\ & (\bar{t} \vee z \vee \bar{c}) \wedge (t \vee \bar{x}) \wedge (t \vee c) \\ \wedge\ & (t \vee d) \end{aligned}$$

Linéaire, et encore optimisable (pas expliqué ici)
Equivalent pour la satisfiabilité et la validité
mais pas pour d'autres propriétés : taille des modèles, etc.

Agenda

1. Le calcul Booléen (rappels)
2. La forme normale conjonctive (CNF)
3. Exemples de problèmes SAT
4. De la résolution à Davis and Putnam
5. CDCL = Conflict Driven Clause Learning
6. Two Literal Watching
7. Conclusion

Résolution d'un Sudoku par codage en SAT

http://www.cs.qub.ac.uk/~I.Spence/SuDoku/SuDoku.html

4				7	9	6		
		9						
			6				8	3
								7
5			8			1		
7	8			1	2		4	
6	5							
				8		7		4
		4		2		3		

4	3	8	2	7	9	6	5	1
1	6	9	3	5	8	4	7	2
2	7	5	6	4	1	9	8	3
9	2	1	5	6	4	8	3	7
5	4	6	8	3	7	1	2	9
7	8	3	9	1	2	5	4	6
6	5	7	4	9	3	2	1	8
3	9	2	1	8	5	7	6	4
8	1	4	7	2	6	3	9	5

Sudoku en CNF : facile et ultra-rapide !

Variable $x_{i,j,v}$: ligne $1 \le i \le 9$, colonne $1 \le j \le 9 \rightarrow$ valeur $1 \le v \le 9$

Ordre lexicographique et incompatibilité :

$(i,j) <_\ell (i',j')$ ssi $i < i'$ ou $i = i'$ et $j < j'$

$(i,j) \not\prec (i',j')$ ssi $(i,j) <_\ell (i',j')$ et (i,j), (i',j') Sudoku-incompatibles

Exemple : $(1,1) \not\prec (1,2..9), (2..9,1), (2,2), (2,3), (3,2), (3,3)$

Contrainte 1 : Une et une seule valeur par case :

$$\bigwedge_{i,j} ((\bigvee_v x_{i,j,v}) \wedge (\bigvee_{1 \le v < v' \le 9} (\overline{x}_{i,j,v} \vee \overline{x}_{i,j,v'})))$$

Contrainte 2 : incompatibilité entre cases :

$$\bigwedge_{i,j,i',j'.(i,j) \not\prec (i',j')} (\bigvee_{1 \le v \le 9} (\overline{x}_{i,j,v} \vee \overline{x}_{i',j',v}))$$

Contraintes initiales :

$x_{i,j,v}$ si la case (i,j) contient la valeur v dans le problème

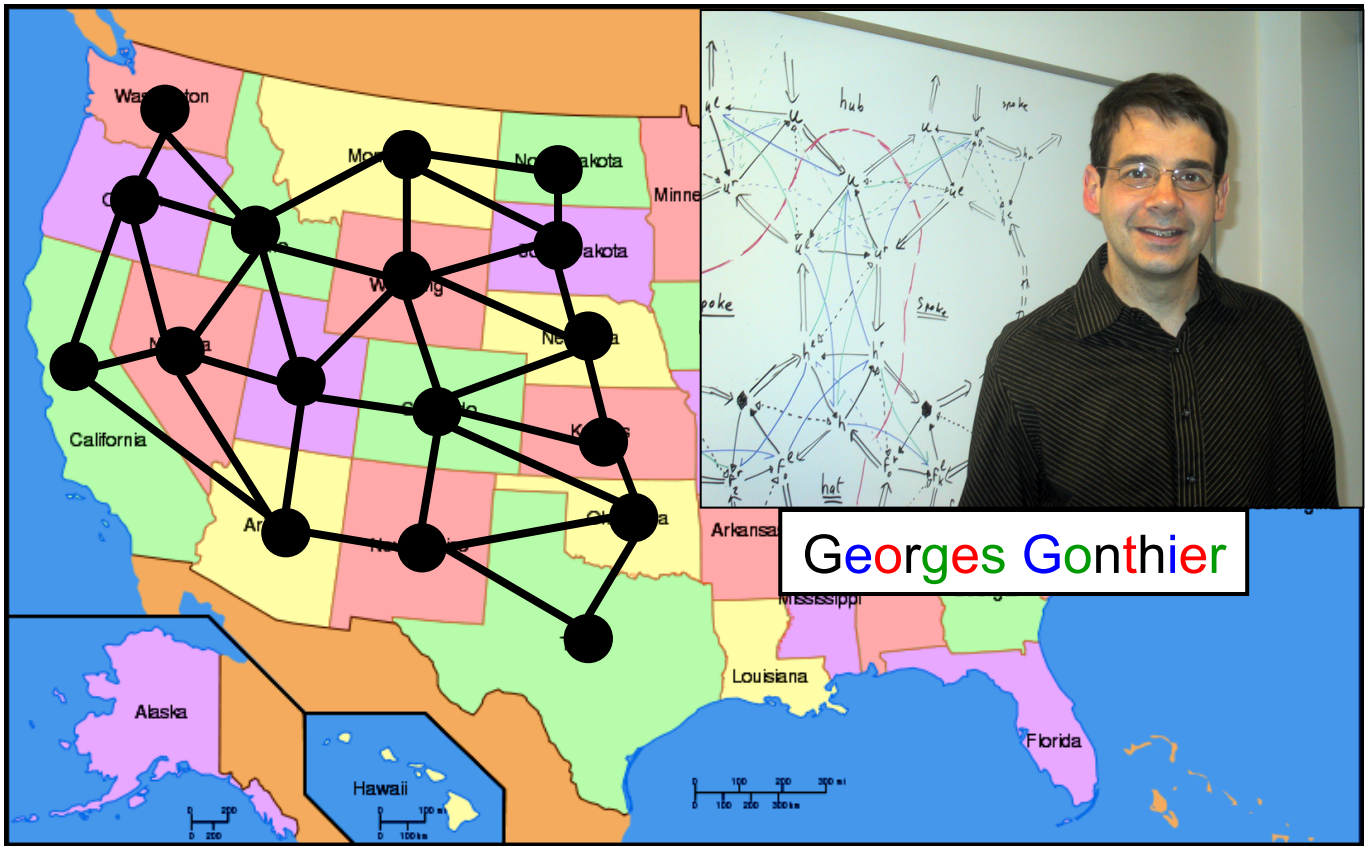
Problème général : coloration de graphe

Soit $G = (S, A)$ un graphe non orienté de sommets S et d'arêtes $A \subset S \times S$. Une coloration de G par un ensemble C de couleurs est une application c de S dans C telle que pour toute arête $a : s_1 - s_2$ on ait $c(s_1) \neq c(s_2)$.

Nombre chromatique de G = $|C|$ minimal pour colorier G

- Complexité
  - Polynomial pour 2 couleurs
  - NP-complet pour plus de 2 couleurs
- Nombreuses applications
  - Sudoku (graphe d'incompatibilité des cases), autres jeux
  - optimisation : scheduling, allocation de registres
  - cryptologie : 3-coloration utilisée pour l'authentification
  - ...

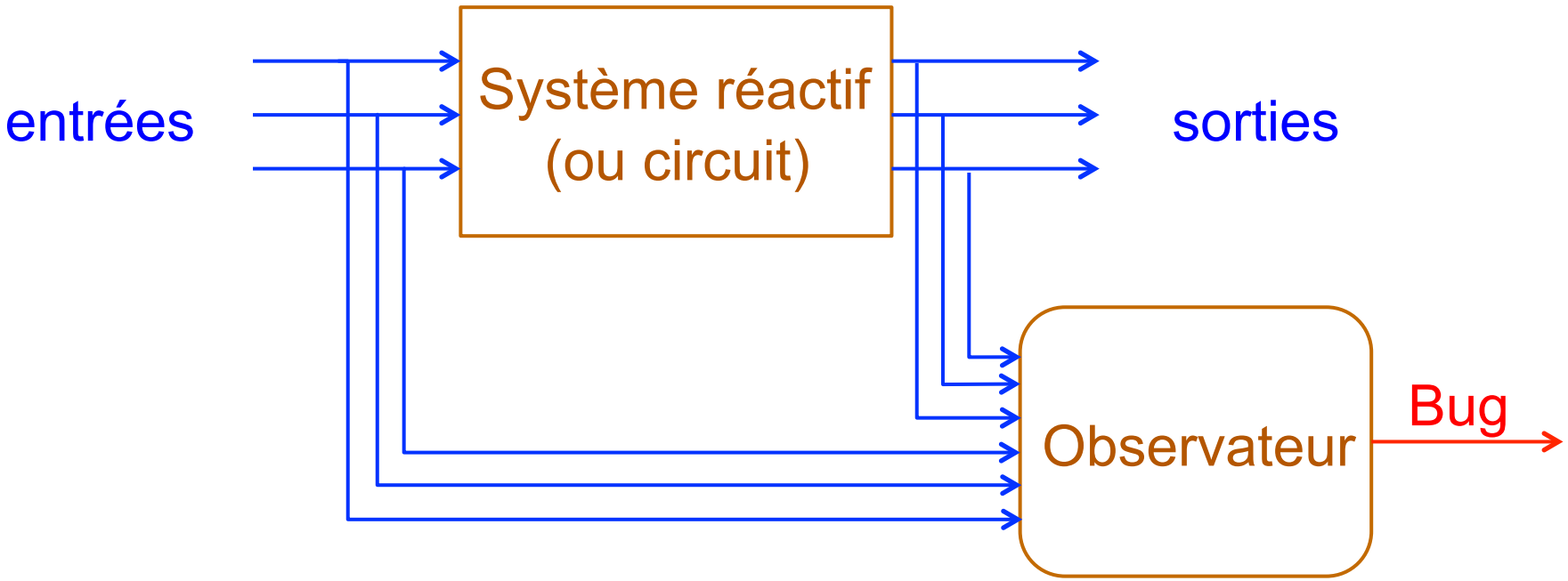
Théorème des 4 couleurs = coloration de graphe



- 1852 Guthrie
- 1976 Appel – Haken
- 2005 Gonthier (en Coq)

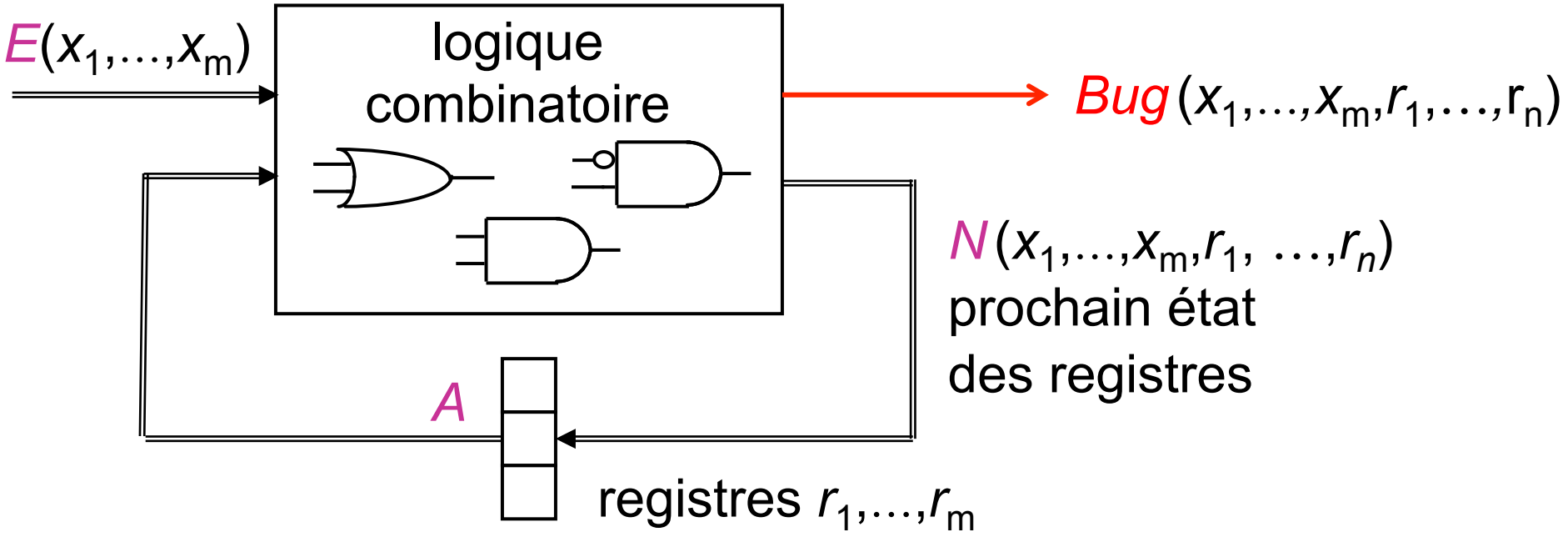
Théorème : toute carte planaire peut être coloriée avec 4 couleurs de façon à ce que tous les pays ayant une frontière commune non réduite à un point soient de couleurs différentes

Vérification booléenne de sûreté



Formule booléenne exprimant que Bug n'est jamais émis :
pour tout état accessible R, O et toute entrée valide E
du couple système / observateur, le booléen Bug est faux.

Model-Checking borné en SAT

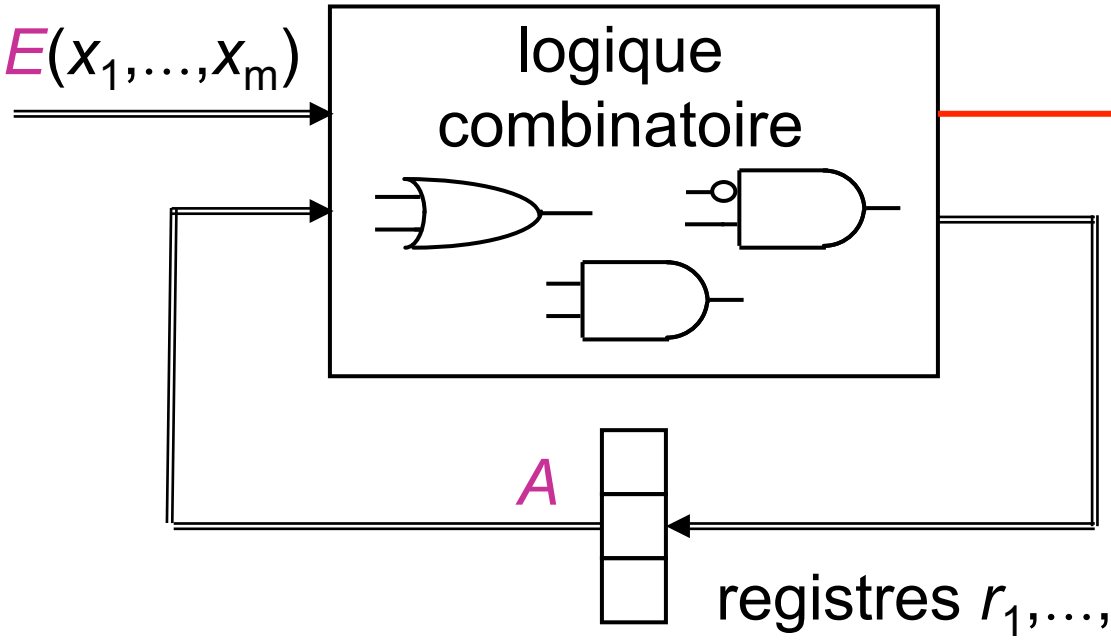


Prédicat de validité des entrées : $E(x_0,\ldots,x_m)$

Valeur initiale des registres : $I(r_0,\ldots,r_n)$

SAT ≠ BDD : il n'est ni simple ni forcément utile de calculer l'ensemble des états accessibles $A(r_0,\ldots,r_n)$ et l'ensemble des contre-exemples possibles

Model-Checking borné en SAT



Prédicat de validité des entrées : $E(x_0,\ldots,x_m)$

Valeur initiale des registres : $I(r_0,\ldots,r_n)$

Solution (partielle) : déplier itérativement le circuit

Model-Checking borné en SAT

Dépliage d'ordre k :

$$I(\vec{r}^{\,0})$$
$$\wedge\ E(\vec{x}^{\,0}) \wedge (\vec{r}^{\,1} \Leftrightarrow N(\vec{x}^{\,0}, \vec{r}^{\,0}))$$
$$\wedge\ E(\vec{x}^{\,1}) \wedge (\vec{r}^{\,2} \Leftrightarrow N(\vec{x}^{\,1}, \vec{r}^{\,1}))$$
$$\wedge \ldots$$
$$\wedge\ E(\vec{x}^{\,k\text{-}1}) \wedge (\vec{r}^{\,k} \Leftrightarrow N(\vec{x}^{\,k\text{-}1}, \vec{r}^{\,k\text{-}1}))$$
$$\wedge\ Bug(\vec{x}^{\,k}, \vec{r}^{\,k})$$

Si la formule est satisfiable pour k ,un bug est trouvé et le SAT-solver fournit une suite d'entrées $\vec{x}^{\,0}, \vec{x}^{\,1}, \ldots, \vec{x}^{\,k}$
Mais introduction de variables à chaque dépliement !
Et la méthode ne peut décider que *Bug* ne sera jamais vrai

Problème dur : n+1 pigeons pour n trous

- On se donne *m* pigeons et *n* trous (chaussettes / tiroirs)
- $x_{i,j}$: variable vraie si le pigeon *i* est dans le trou *j*
- Chaque pigeon est dans un trou

$$\bigwedge_{i<m} \left(\bigvee_{j<n} x_{i,j} \right)$$

... et chaque pigeon est dans au plus un trou

$$\bigwedge_{i<m,\ j<n,\ j'<n} \left(\overline{x_{i,j}} \vee \overline{x_{i,j'}} \right)$$

... et chaque trou contient au plus un pigeon

$$\bigwedge_{i<m,\ i'<n,\ j<n} \left(\overline{x_{i,j}} \vee \overline{x_{i',j}} \right)$$

SAT est facile si $m \leqslant n$, UNSAT est très dur si $m \geqq n+1$
SAT ne peut pas prouver cela pour tout m (pas de récurrence) !

Agenda

1. Le calcul Booléen (rappels)
2. La forme normale conjonctive (CNF)
3. Exemples de problèmes SAT
4. De la résolution à Davis and Putnam
5. CDCL = Conflict Driven Clause Learning
6. Two Literal Watching
7. Conclusion

Comment vérifier une formule (SAT / UNSAT) ?

1. Méthodes syntaxiques :
   - déduction logique, utilisation des règles algébriques
   - méthode standard : la résolution de Robinson (1965)

   itération de $(x \vee A) \wedge (\overline{x} \vee B) \rightarrow (A \vee B)$

   ex. 1 : $(x \vee \overline{y} \vee t \vee \overline{u}) \wedge (y \vee \overline{u} \vee v)$
   $\rightarrow x \vee t \vee \overline{u} \vee \overline{u} \vee v \rightarrow x \vee t \vee \overline{u} \vee v$

   ex. 2 : $(x \vee \overline{y} \vee t \vee \overline{u}) \wedge (y \vee u \vee v)$
   $\rightarrow x \vee t \vee \overline{u} \vee u \vee v \rightarrow 1$

   Méthode complète mais facilement exponentielle !

   ex. 3 : Modus Ponens : $(A \wedge (A \Rightarrow B)) \Rightarrow B$
   Preuve par réfutation : $x \wedge (x \Rightarrow y) \wedge \overline{y} \rightarrow 0$
   $x \wedge (\overline{x} \vee y) \wedge \overline{y}$
   $\rightarrow y \wedge \overline{y}$
   $\rightarrow$ clause vide $\rightarrow 0 \rightarrow$ UNSAT

Principe de DPLL

$$\varphi_1 = x_1 \vee x_4$$
$$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$$
$$\varphi_3 = x_1 \vee x_8 \vee x_{12}$$
$$\varphi_4 = x_2 \vee x_{11}$$
$$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$$
$$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$$
$$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$$

1. Choix d'une variable et d'une valeur

$$\varphi_1 = x_1 \vee x_4$$
$$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$$
$$\varphi_3 = x_1 \vee x_8 \vee x_{12}$$
$$\varphi_4 = x_2 \vee x_{11}$$
$$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$$
$$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$$
$$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$$

1. Choix d'une variable et d'une valeur

$$\varphi_1 = x_1 \vee x_4$$
$$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$$
$$\varphi_3 = x_1 \vee x_8 \vee x_{12}$$
$$\varphi_4 = x_2 \vee x_{11}$$
$$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$$
$$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$$
$$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$$

x_1

1a. Clause unitaire

$$\varphi_1 = x_1 \vee x_4$$
$$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$$
$$\varphi_3 = x_1 \vee x_8 \vee x_{12}$$
$$\varphi_4 = x_2 \vee x_{11}$$
$$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$$
$$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$$
$$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$$

x_1

1a. Clause unitaire

$$\varphi_1 = x_1 \vee x_4$$
$$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$$
$$\varphi_3 = x_1 \vee x_8 \vee x_{12}$$
$$\varphi_4 = x_2 \vee x_{11}$$
$$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$$
$$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$$
$$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$$

$$x_1 \Rightarrow x_4[\varphi_1]$$

2. Choix de variable et de valeur

$$\varphi_1 = x_1 \vee x_4$$
$$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$$
$$\varphi_3 = x_1 \vee x_8 \vee x_{12}$$
$$\varphi_4 = x_2 \vee x_{11}$$
$$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$$
$$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$$
$$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$$

$$x_1 \Rightarrow x_4 [\varphi_1]$$

2. Choix de variable et de valeur

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x_3} \vee \overline{x_8}$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x_3} \vee \overline{x_7} \vee x_{13}$

$\varphi_6 = \overline{x_3} \vee \overline{x_7} \vee \overline{x_{13}} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x_7} \vee \overline{x_9}$

$x_1 \Rightarrow x_4 [\varphi_1]$

$x_1 \rightarrow x_3$

2a. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

x_1

$\downarrow$

$\overline{x}_3$

2a. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2]$

2b. Propagation

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2]$

2c. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2]$

2c. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

3. Choix de variable et de valeur

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

3. Choix de variable et de valeur

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

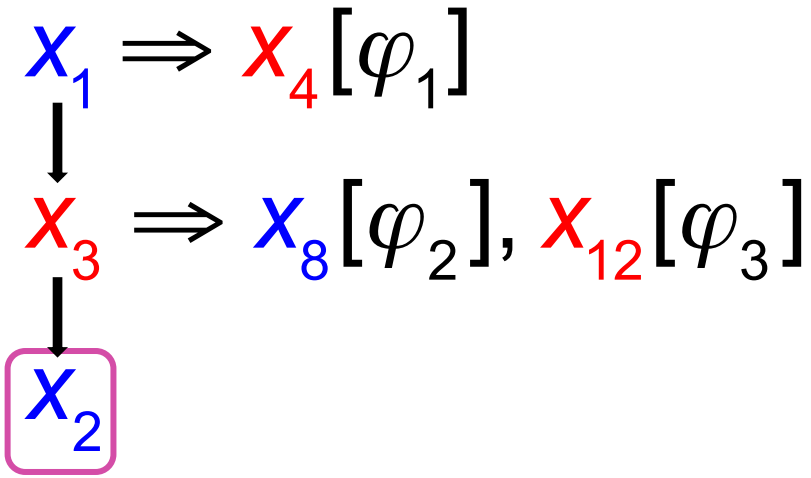
$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$



3a. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

x_2

3a. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

4. Choix de variable et de valeur

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

4. Choix de variable et de valeur

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

x_7

4a. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

x_7

4a. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

$x_7 \Rightarrow x_{13}[\varphi_5]$

4b. Propagation

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

$x_7 \Rightarrow x_{13}[\varphi_5]$

4c. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

$x_7 \Rightarrow x_{13}[\varphi_5]$

4c. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

$x_7 \Rightarrow x_{13}[\varphi_5], x_9[\varphi_6]$

4d. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

$x_7 \Rightarrow x_{13}[\varphi_5], x_9[\varphi_6]$

4d. Clause unitaire

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

$x_7 \Rightarrow x_{13}[\varphi_5], x_9[\varphi_6]$

4e. Contradiction !

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9 = 0$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

$x_7 \Rightarrow x_{13}[\varphi_5], x_9[\varphi_6]$

4e. Ou clause unitaire → contradiction !

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

$x_7 \Rightarrow x_{13}[\varphi_5], x_9[\varphi_6], x_9[\varphi_7]$

C'est là que les choses intéressantes commencent !

DPLL : Davis–Putnam–Logemann–Loveland (1962)

$$\varphi_1 = x_1 \vee x_4$$
$$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$$
$$\varphi_3 = x_1 \vee x_8 \vee x_{12}$$
$$\varphi_4 = x_2 \vee x_{11}$$
$$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$$
$$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$$
$$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9 = 0$$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

x_7

Inversion du dernier choix (backtrack simple)
marche ici, mais inefficace en général !

Agenda

1. Le calcul Booléen (rappels)
2. La forme normale conjonctive (CNF)
3. Exemples de problèmes SAT
4. De la résolution à Davis and Putnam
5. CDCL = Conflict Driven Clause Learning
6. Two Literal Watching
7. Conclusion

4e. Contradiction !

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$x_1 \Rightarrow x_4[\varphi_1]$

$\downarrow$

$x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

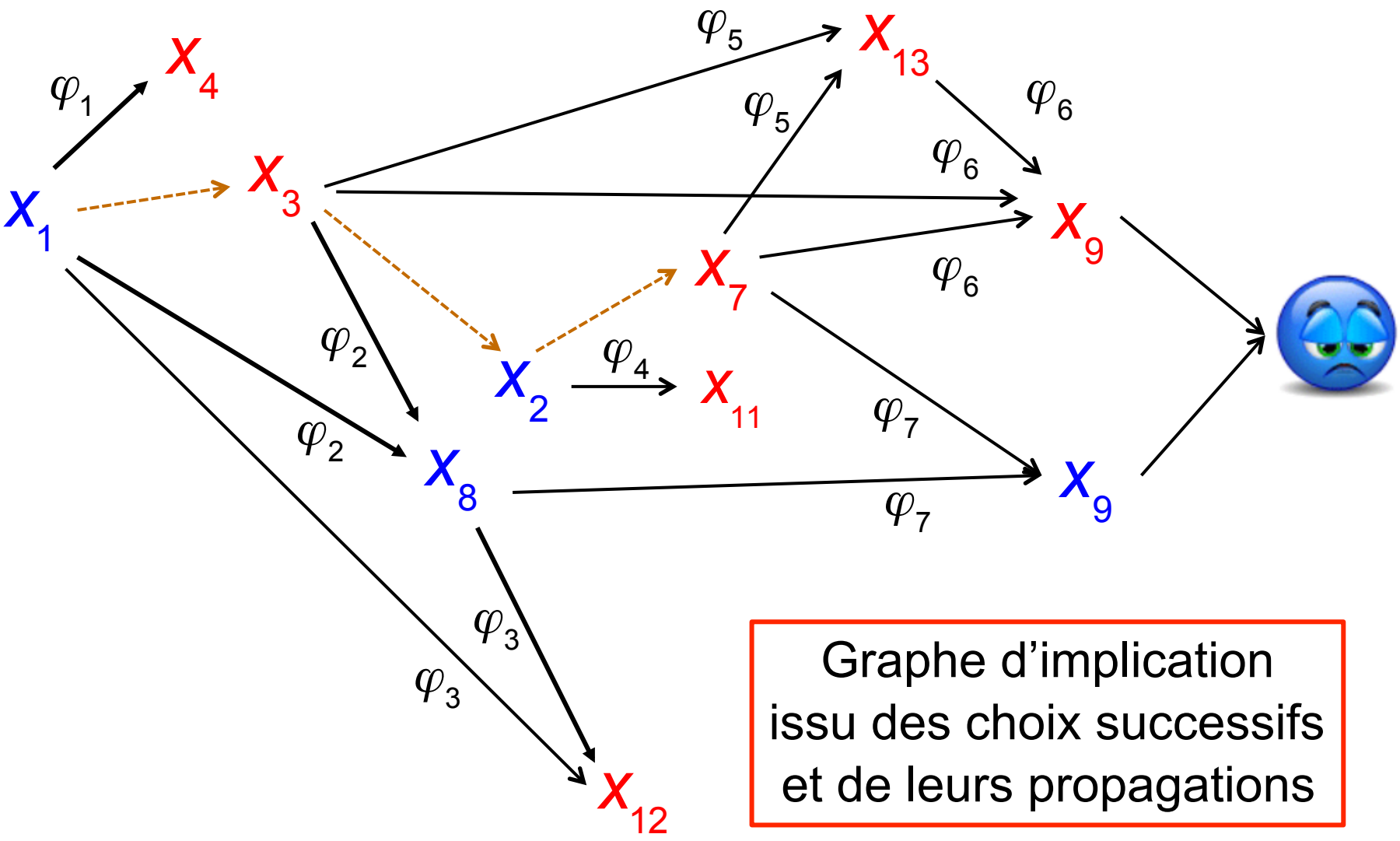
$\downarrow$

$x_2 \Rightarrow x_{11}[\varphi_4]$

$\downarrow$

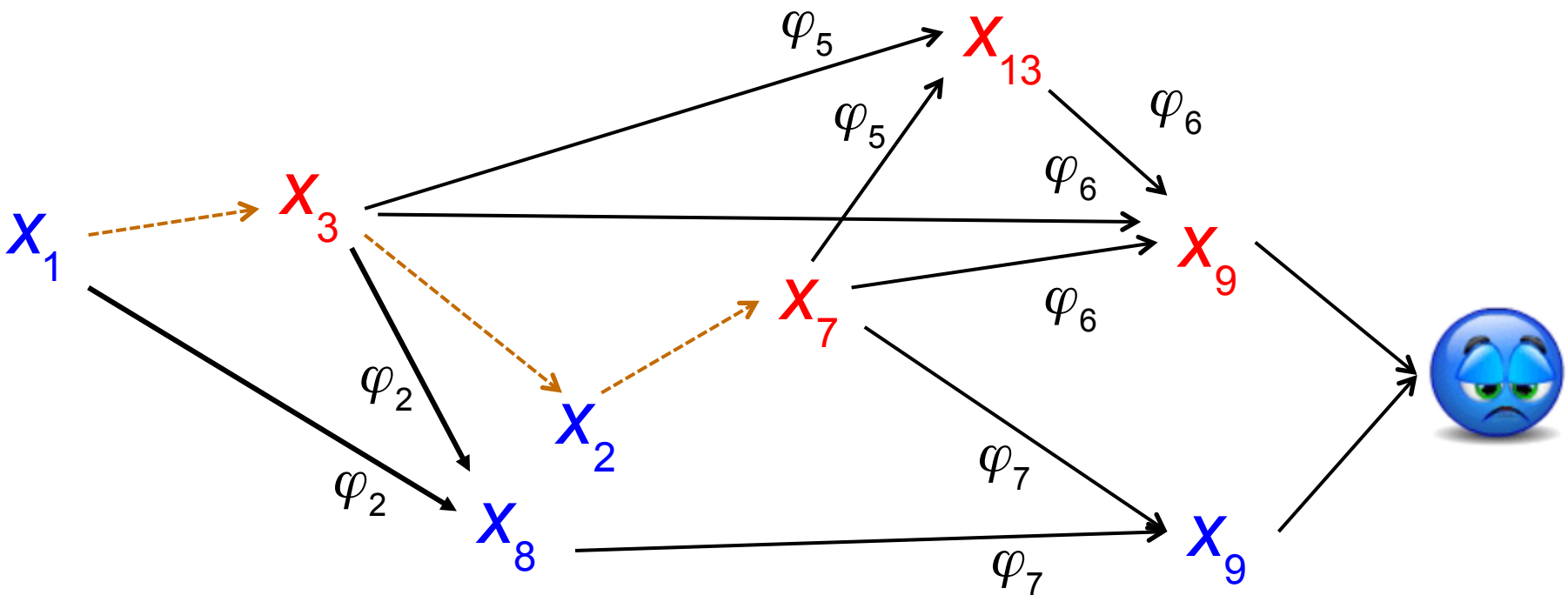
$x_7 \Rightarrow x_{13}[\varphi_5], x_9[\varphi_6], x_9[\varphi_7]$

CDCL : Conflict Driven Clause Learning

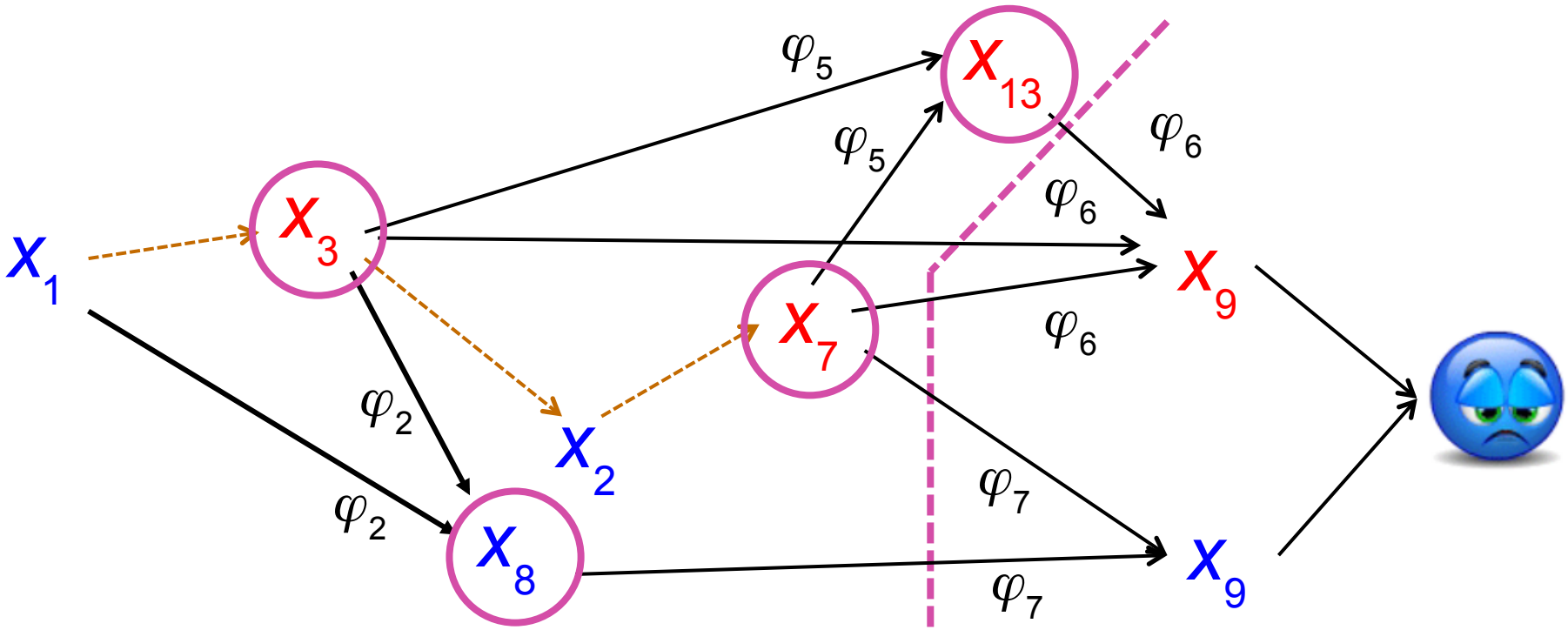


Graphe d'implication issu des choix successifs et de leurs propagations

Enlever ce qui ne participe pas au conflit



Ajouter une clause qui évite le conflit – v1



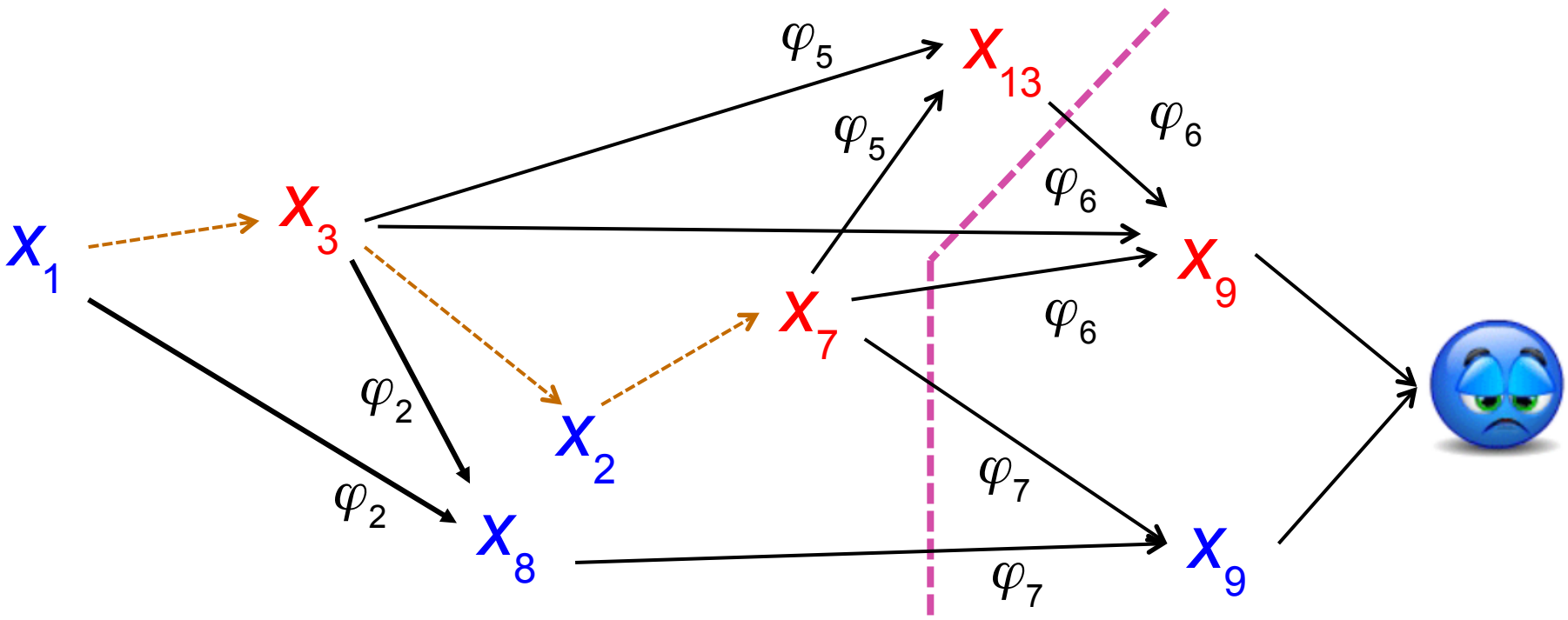
Coupure → résolution de φ_6 et φ_7 par x_9

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

$\beta_1 = res(x_9, \varphi_6, \varphi_7) = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_8$

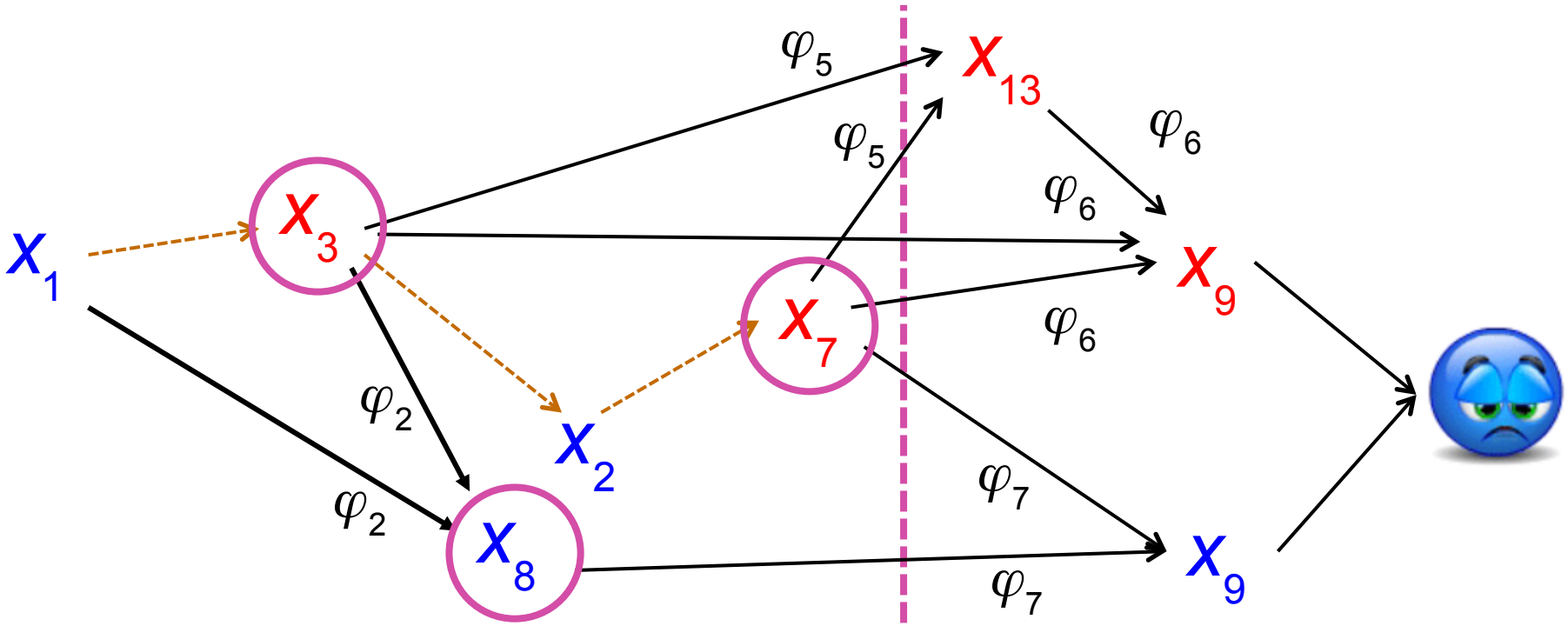
Mais peut mieux faire !



Deux variables du dernier niveau de décision

$$\beta_1 = res(x_9, \varphi_6, \varphi_7) = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_8$$

FUIP : une seule variable du niveau de décision



Résolution de x_{13} pour β_1 et φ_5

$$\beta_1 = res(x_9, \varphi_6, \varphi_7) = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_8$$

$$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$$

$$\beta = res(x_{13}, \beta_1, \varphi_5) = \overline{x}_3 \vee \overline{x}_7 \vee x_8$$

Mémoire habile du conflit

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

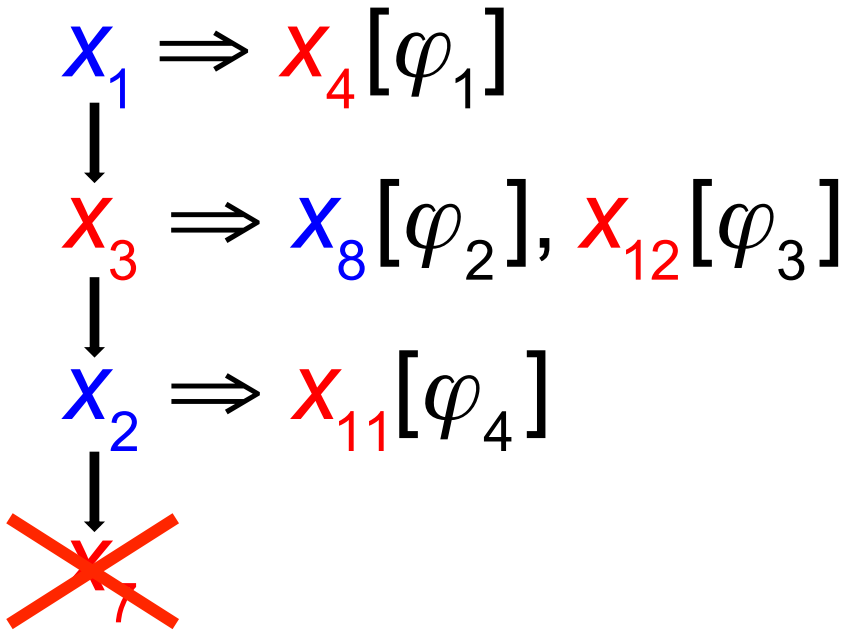
$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

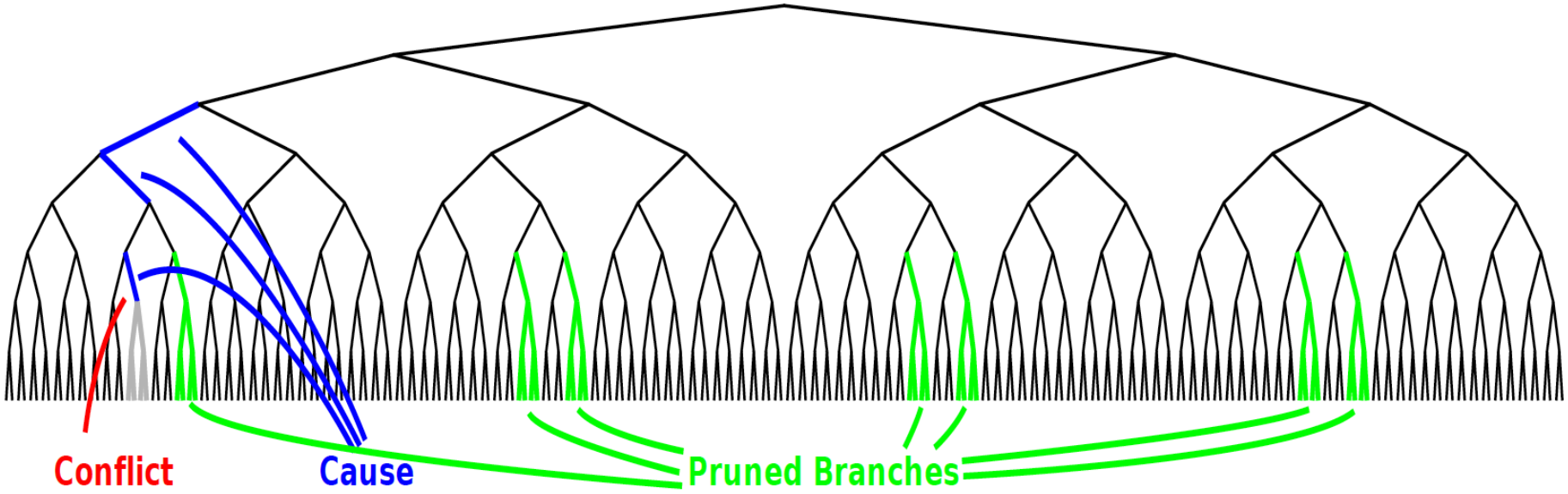
$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$



$\beta = \overline{x}_3 \vee \overline{x}_7 \vee x_8$ ← mémoire du conflit

β provoque immédiatement x_7 si x_1 et x_3

Effet sur la recherche



Source Sol Swords, 2008

La clause ajoutée permet d'éviter ce conflit
dans toutes les occasions où il peut se reproduire

Retour arrière ou redémarrage

$\varphi_1 = x_1 \vee x_4$

$\varphi_2 = x_1 \vee \overline{x}_3 \vee \overline{x}_8$

$\varphi_3 = x_1 \vee x_8 \vee x_{12}$

$\varphi_4 = x_2 \vee x_{11}$

$\varphi_5 = \overline{x}_3 \vee \overline{x}_7 \vee x_{13}$

$\varphi_6 = \overline{x}_3 \vee \overline{x}_7 \vee \overline{x}_{13} \vee x_9$

$\varphi_7 = x_8 \vee \overline{x}_7 \vee \overline{x}_9$

➡ redémarrage complet

➡ $x_1 \Rightarrow x_4[\varphi_1]$

➡ $x_3 \Rightarrow x_8[\varphi_2], x_{12}[\varphi_3]$

➡ $x_2 \Rightarrow x_{11}[\varphi_4]$

➡ x_7

$\beta = \overline{x}_3 \vee \overline{x}_7 \vee x_8$ ← mémoire du conflit

On repart de là, ou de plus haut, ou même de tout en haut
cf. le séminaire de Laurent Simon ce jour !

Agenda

1. Le calcul Booléen (rappels)
2. La forme normale conjonctive (CNF)
3. Exemples de problèmes SAT
4. De la résolution à Davis and Putnam
5. CDCL = Conflict Driven Clause Learning
6. Two Literal Watching
7. Conclusion

Attention à l'administration !

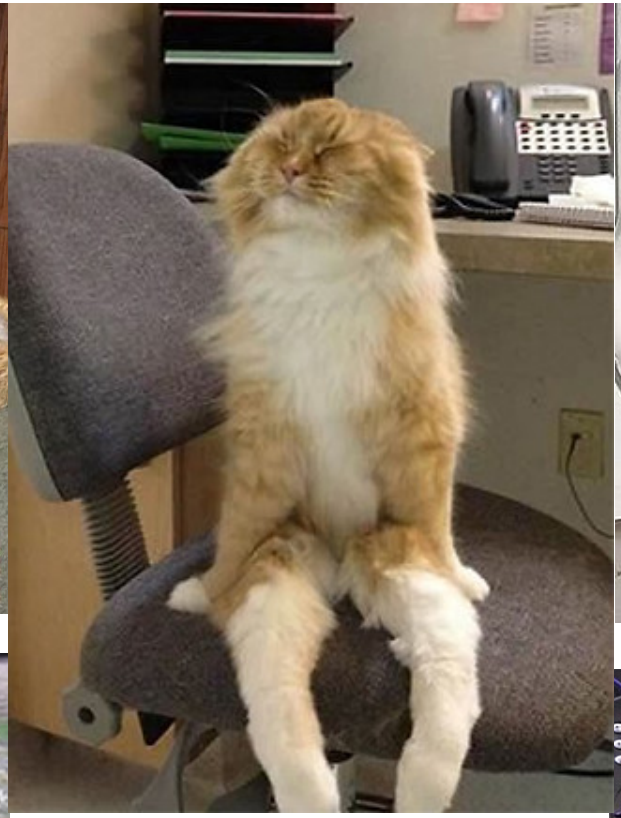
- Implémentation naïve :
  - tout littéral pointe sur toutes ses instance dans les clauses
  - on propage les choix dans les clauses, on cherche les clauses unitaires, on propage leurs effets, etc.
  - à chaque retour arrière, on défait explicitement partout les affectations de valeurs liées aux choix annulés

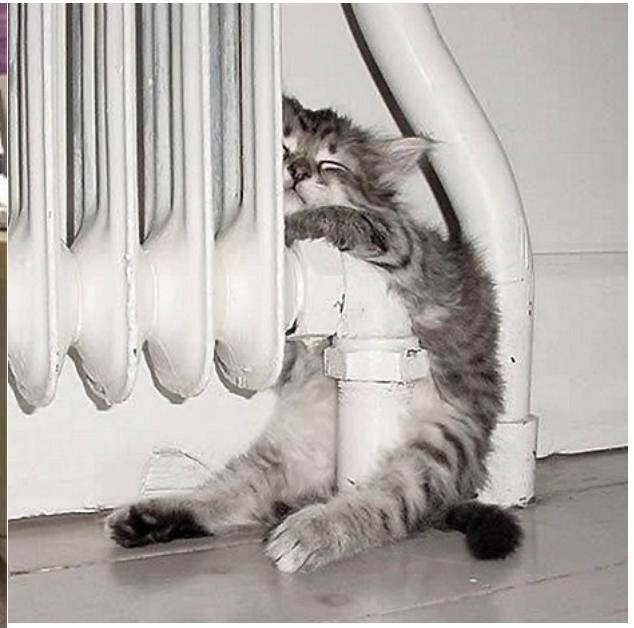
Coût administratif très élevé

Two Literals Watching (zChaff)

Si une clause a 100 littéraux,
il suffit de n'en gérer que 2 à la fois,
et de ne rien faire la plupart du temps

Principe : avoir la paresse du chat !

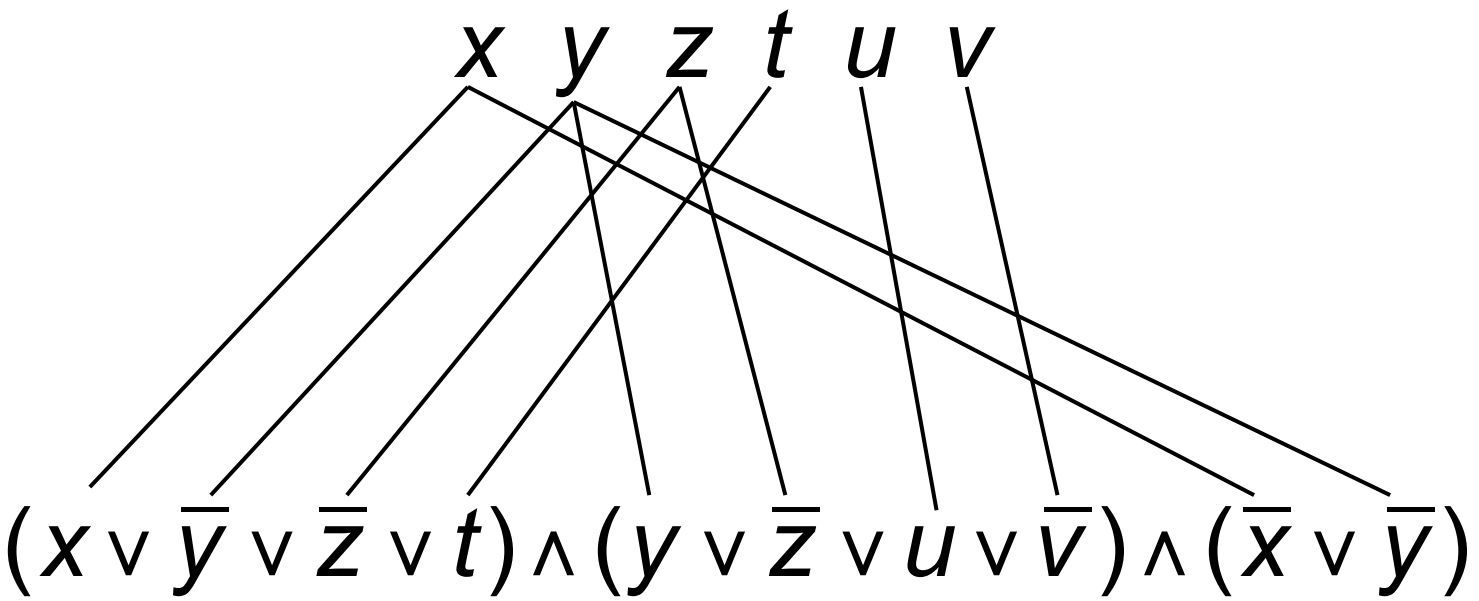




...mais bouger
une patte
de temps en temps

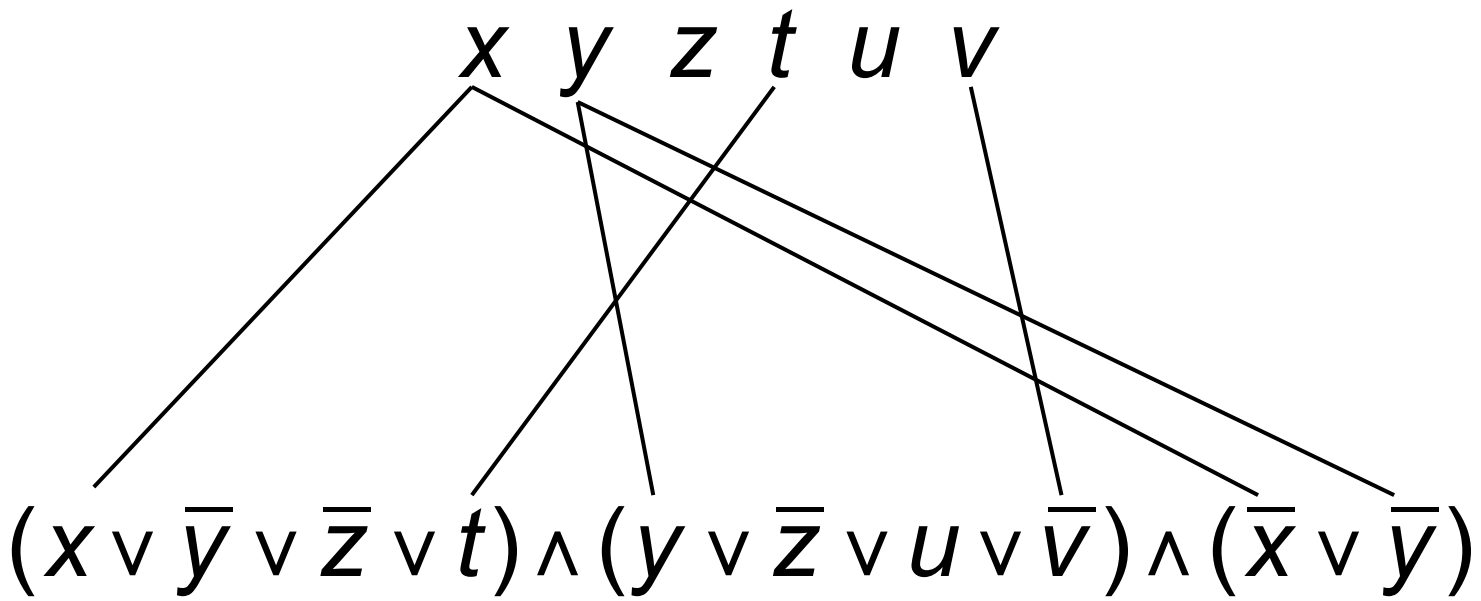
source fr.yummypets.com

Attention à l'administration !



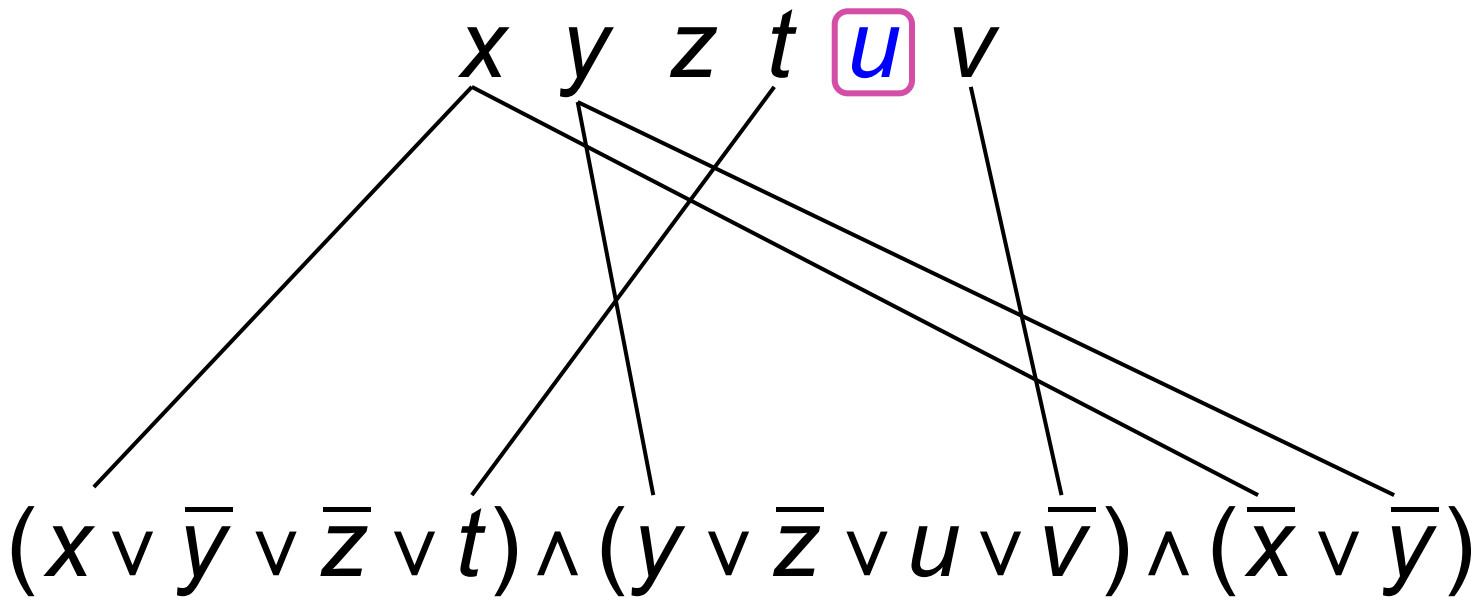
A chaque affectation, il faut propager,
rechercher les clauses unitaires
et défaire les affectations à chaque retour arrière
TROP CHER !

2-Literal Watching (zChaff)



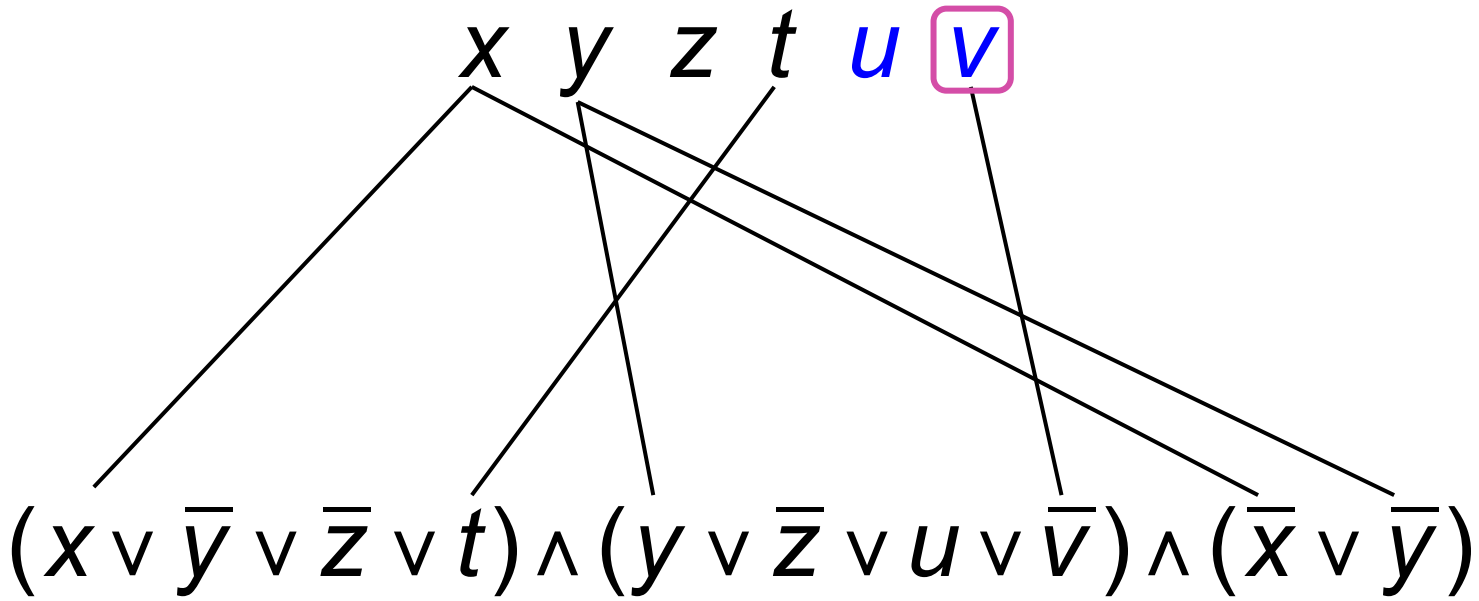
Idée : une clause n'étant productive que quand elle devient unitaire, il suffit de regarder seulement deux littéraux par clause en les gardant toujours distincts

2-Literal Watching (zChaff)



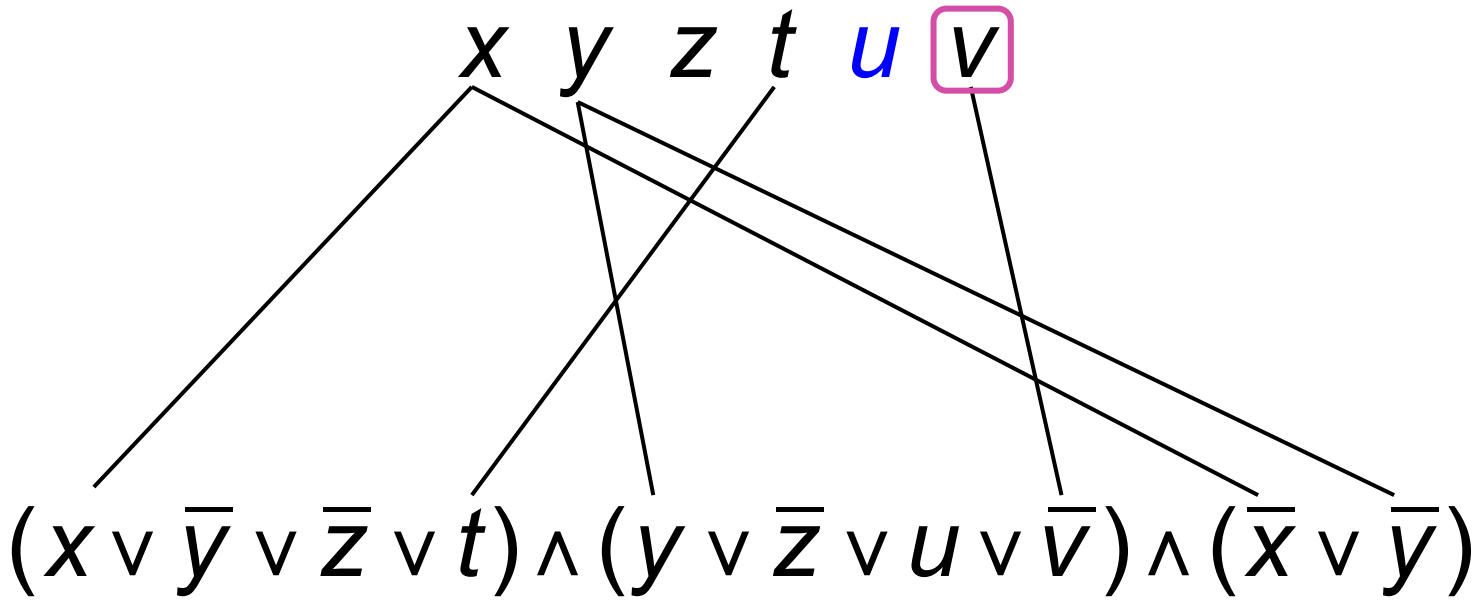
si une variable passe à 1 ou 0,
pour les littéraux non pointés
on ne fait rien du tout
(ici *u* est mis à 0 mais non pointé)

2-Literal Watching (zChaff)



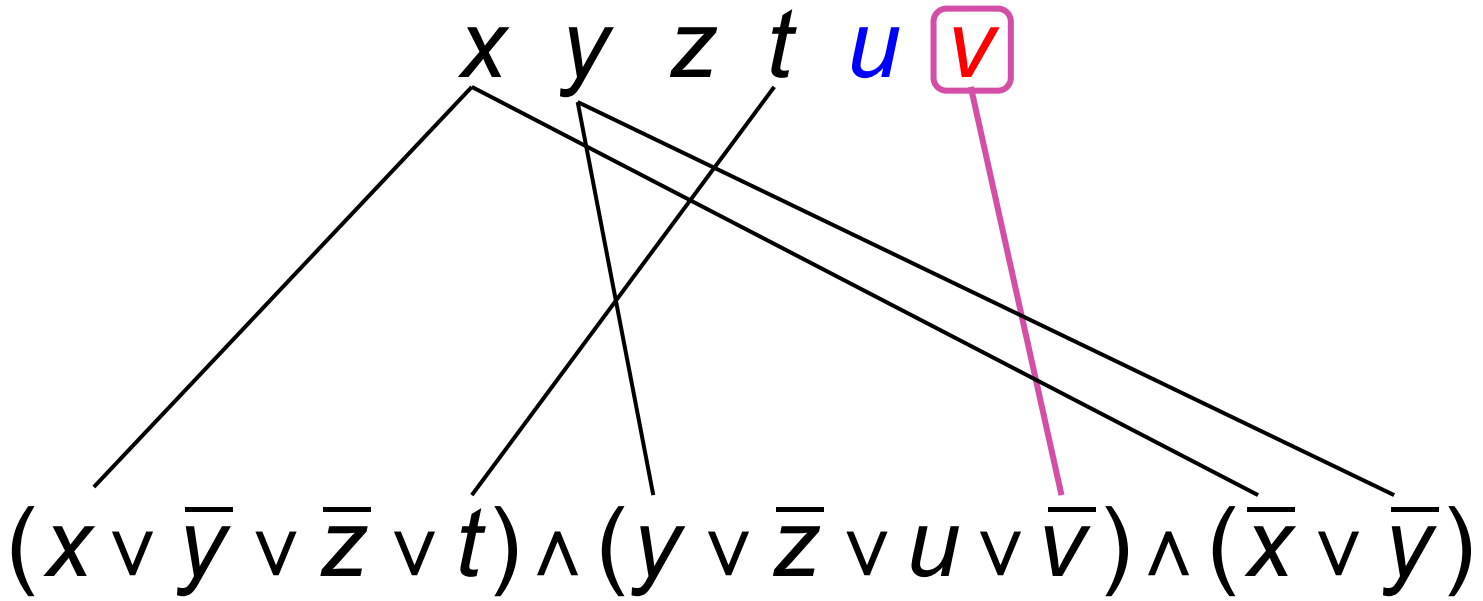
pour un littéral pointé devenant 1
on ne fait rien du tout
(ici $\bar{v}$ pointé de la 2e clause est mis à 1)

2-Literal Watching (zChaff)



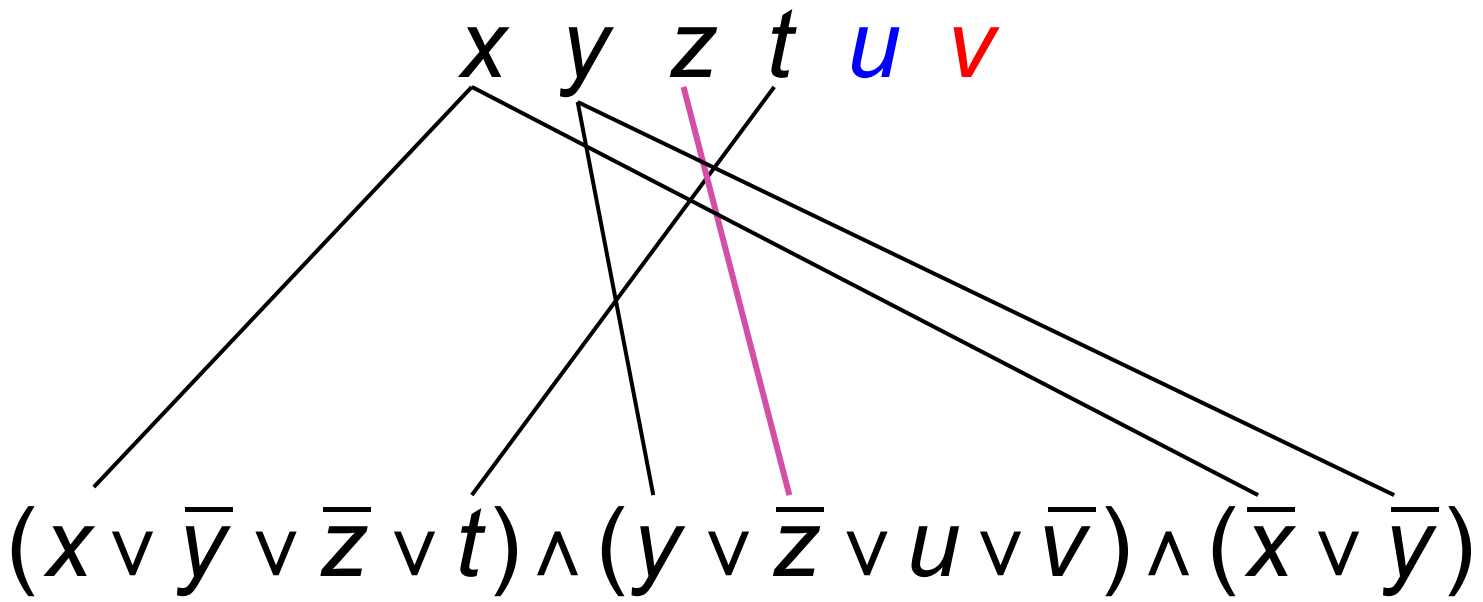
en cas de retour arrière (*bactrack*),
on ne fait rien du tout
(ici $\bar{v}$ pointé de la 2e clause redevient inconnu)

2-Literal Watching (zChaff)



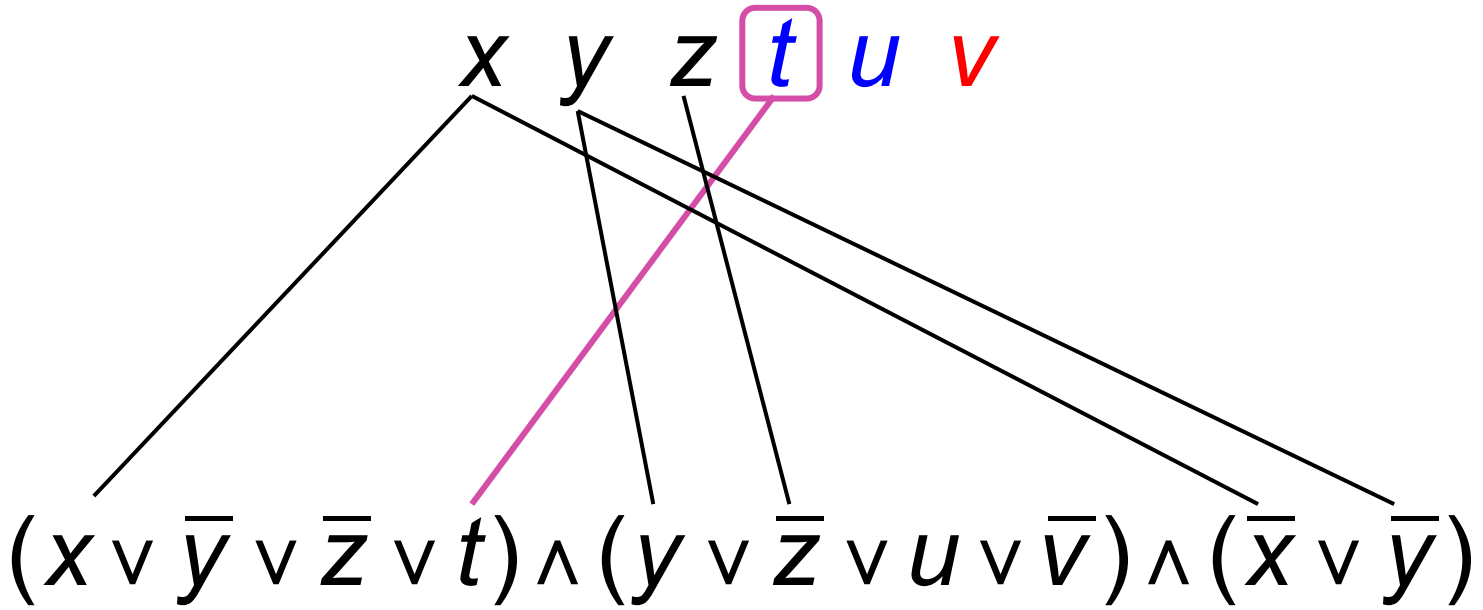
pour un littéral pointé devenant 0 dans une clause,
on essaie de bouger le pointeur sur un littéral non pointé
de la clause valant 1 ou encore indéfini
si c'est possible, on bouge le pointeur et c'est tout
(ici, v à 1, le pointeur de la 2e clause bouge de $\bar{v}$ sur $\bar{z}$)

2-Literal Watching (zChaff)



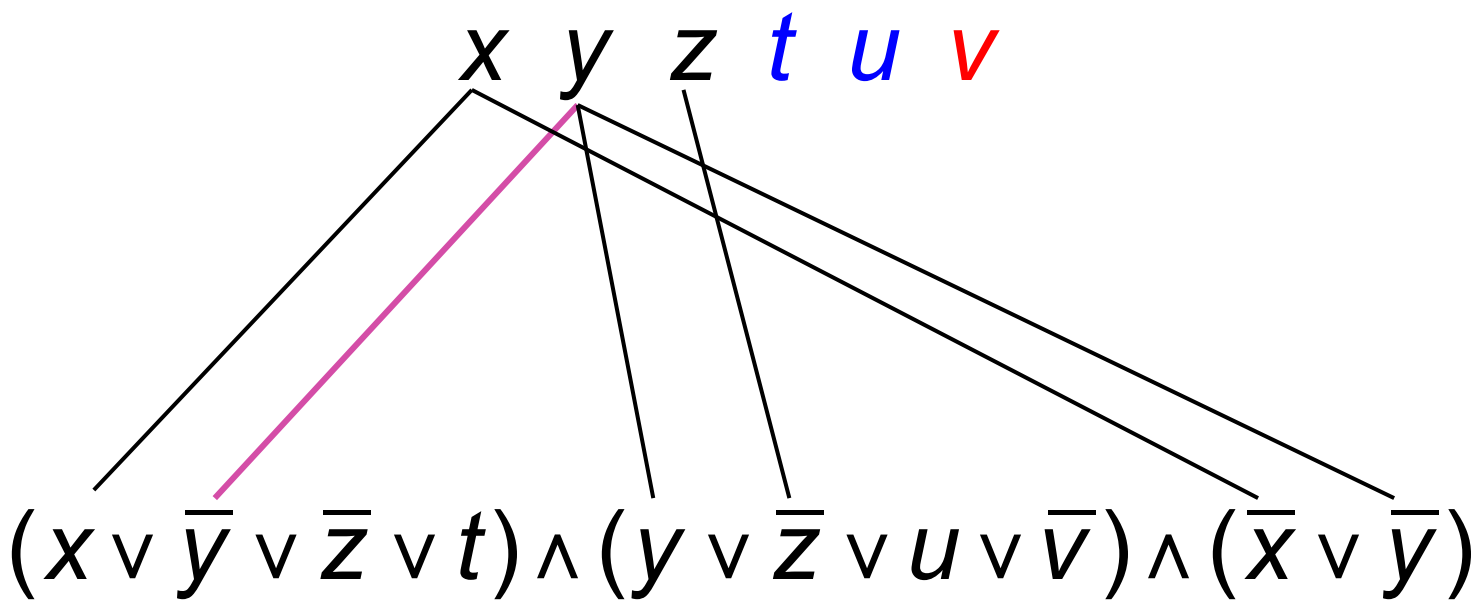
pour un littéral pointé devenant 0 dans une clause,
on essaie de bouger le pointeur sur un littéral non pointé
de la clause valant 1 ou encore indéfini
si c'est possible, on bouge le pointeur et c'est tout
(ici, *v* à 1, le pointeur de la 2e clause bouge de $\bar{v}$ sur $\bar{z}$)

2-Literal Watching (zChaff)



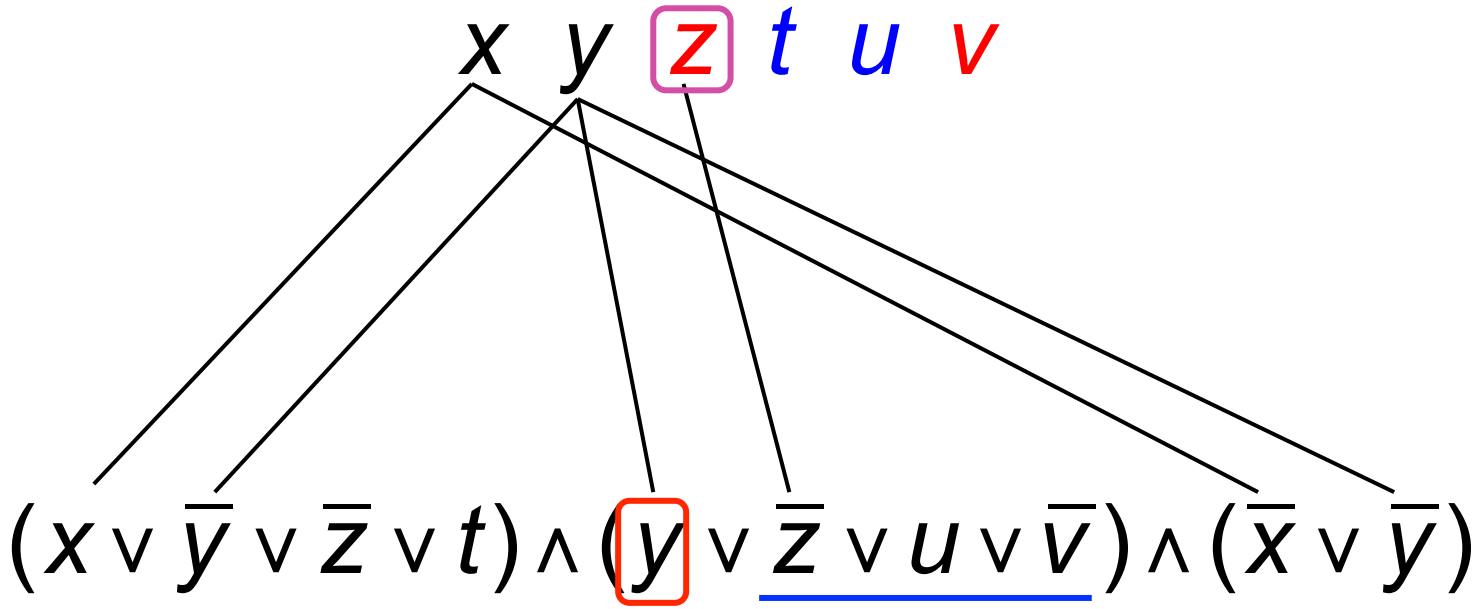
pour un littéral pointé devenant 0 dans une clause,
on essaie de bouger le pointeur sur un littéral non pointé
de la clause valant 1 ou encore indéfini
si c'est possible, on bouge le pointeur et c'est tout
(ici, t à 0, le pointeur de la 1e clause bouge de t sur $\bar{y}$)

2-Literal Watching (zChaff)



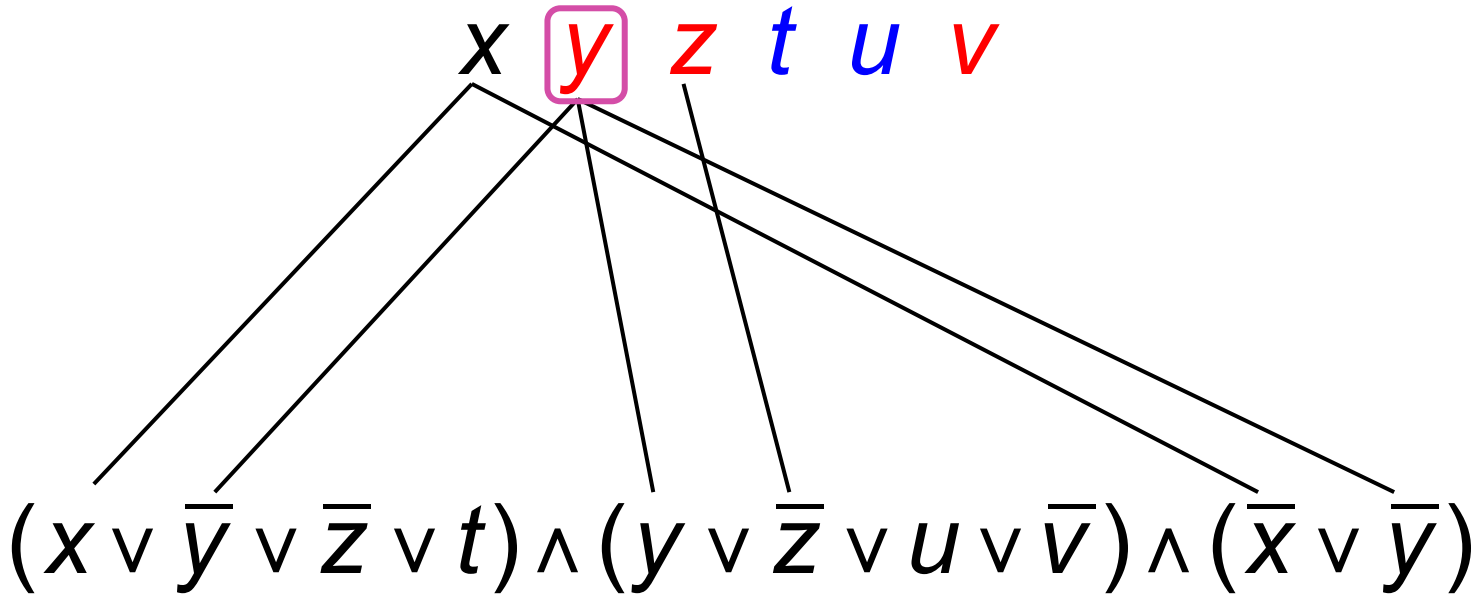
pour un littéral pointé devenant 0 dans une clause,
on essaie de bouger le pointeur sur un littéral non pointé
de la clause valant 1 ou encore indéfini
si c'est possible, on bouge le pointeur et c'est tout
(ici, t à 0, le pointeur de la 1e clause bouge de t sur $\bar{y}$)

2-Literal Watching (zChaff)



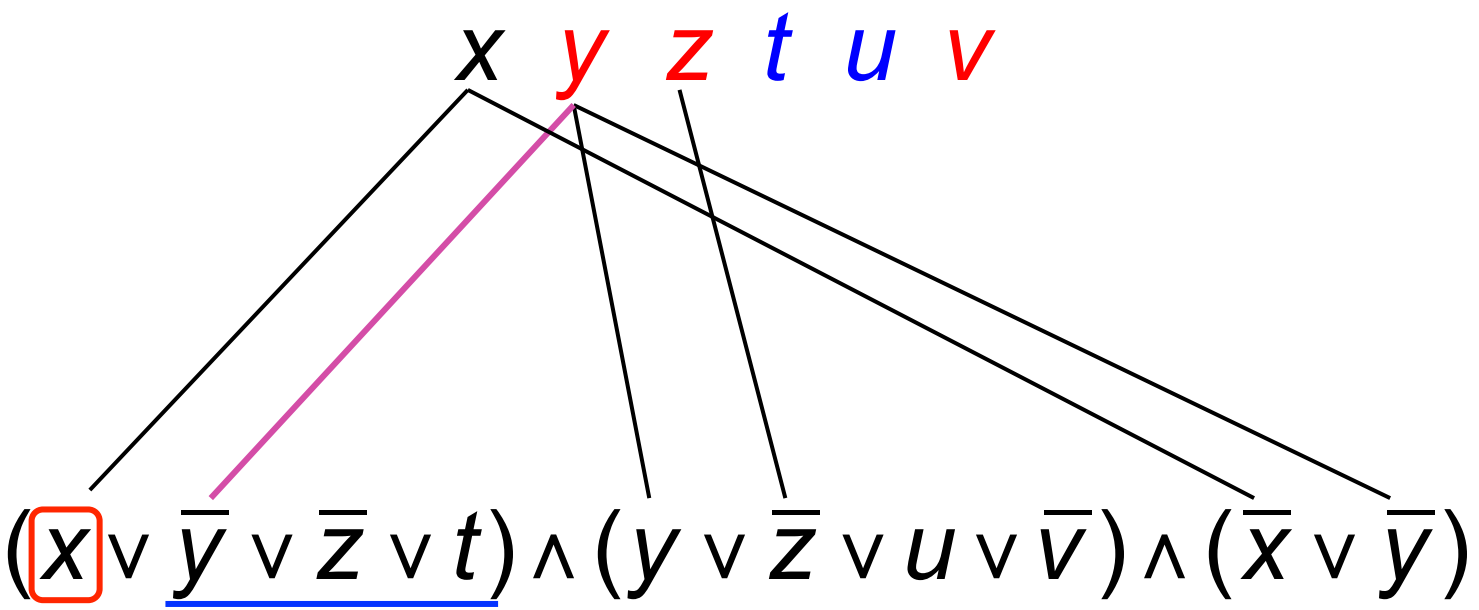
pour un littéral pointé devenant 0 dans une clause
on essaie de bouger le pointeur sur un littéral non pointé
Si pas de littéral non pointé indéfini ou 1, et si l'autre littéral pointé est indéfini, alors la clause est unitaire,
on met cet autre littéral à 1 et c'est tout
(*z* mis à 1 ⇒ la 2e clause est unitaire, et *y* passe à 1)

2-Literal Watching (zChaff)



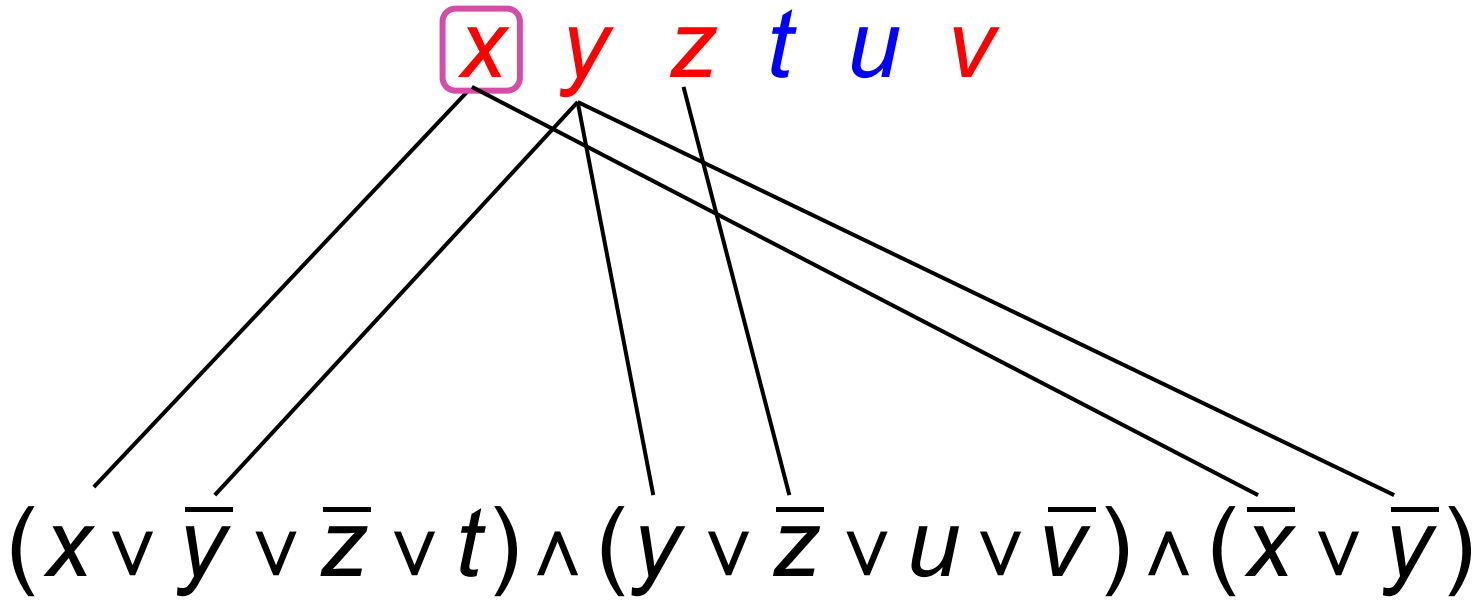
pour un littéral pointé devenant 0 dans une clause
on essaie de bouger le pointeur sur un littéral non pointé
Si pas de littéral non pointé indéfini ou 1, et si l'autre littéral pointé est indéfini, alors la clause est unitaire,
on met cet autre littéral à 1 et c'est tout
(*z* mis à 1 ⇒ la 2e clause est unitaire, et *y* passe à 1)

2-Literal Watching (zChaff)



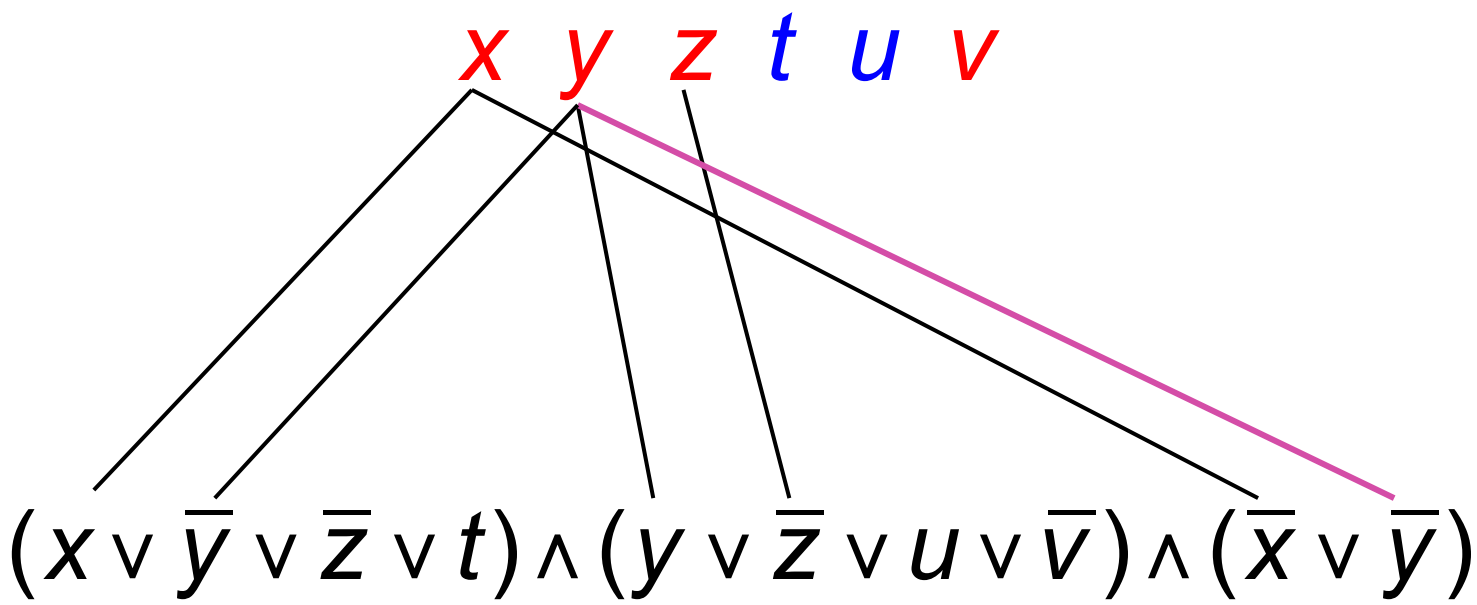
pour un littéral pointé devenant 0 dans une clause
on essaie de bouger le pointeur sur un littéral non pointé
Si pas de littéral non pointé indéfini ou 1, et si l'autre littéral pointé est indéfini, alors la clause est unitaire,
on met cet autre littéral à 1 et c'est tout
(*y* à 1 ⇒ la 1e clause est unitaire, et *x* passe à 1)

2-Literal Watching (zChaff)



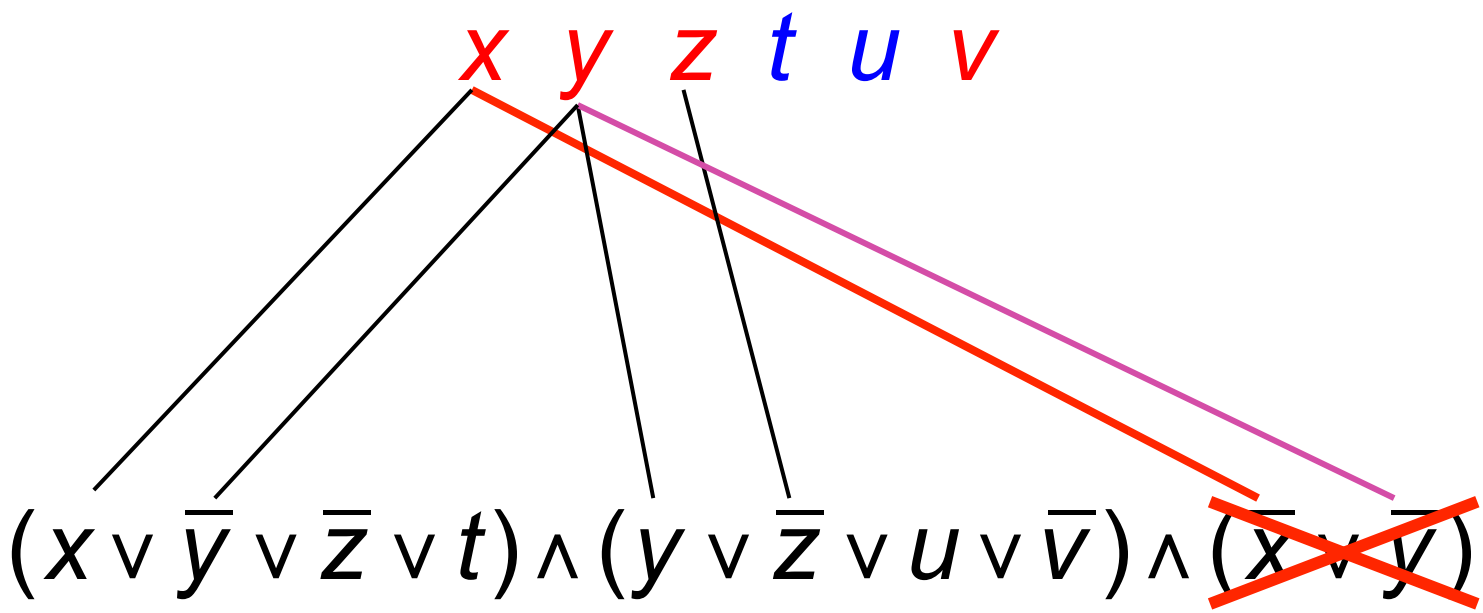
pour un littéral pointé devenant 0 dans une clause
on essaie de bouger le pointeur sur un littéral non pointé
Si pas de littéral indéfini ou 1, et si l'autre littéral pointé est indéfini, alors la clause est unitaire,
on met cet autre littéral à 1 et c'est tout
(*y* mis à 1 ⇒ la 1e clause est unitaire, et *x* passe à 1)

2-Literal Watching (zChaff)



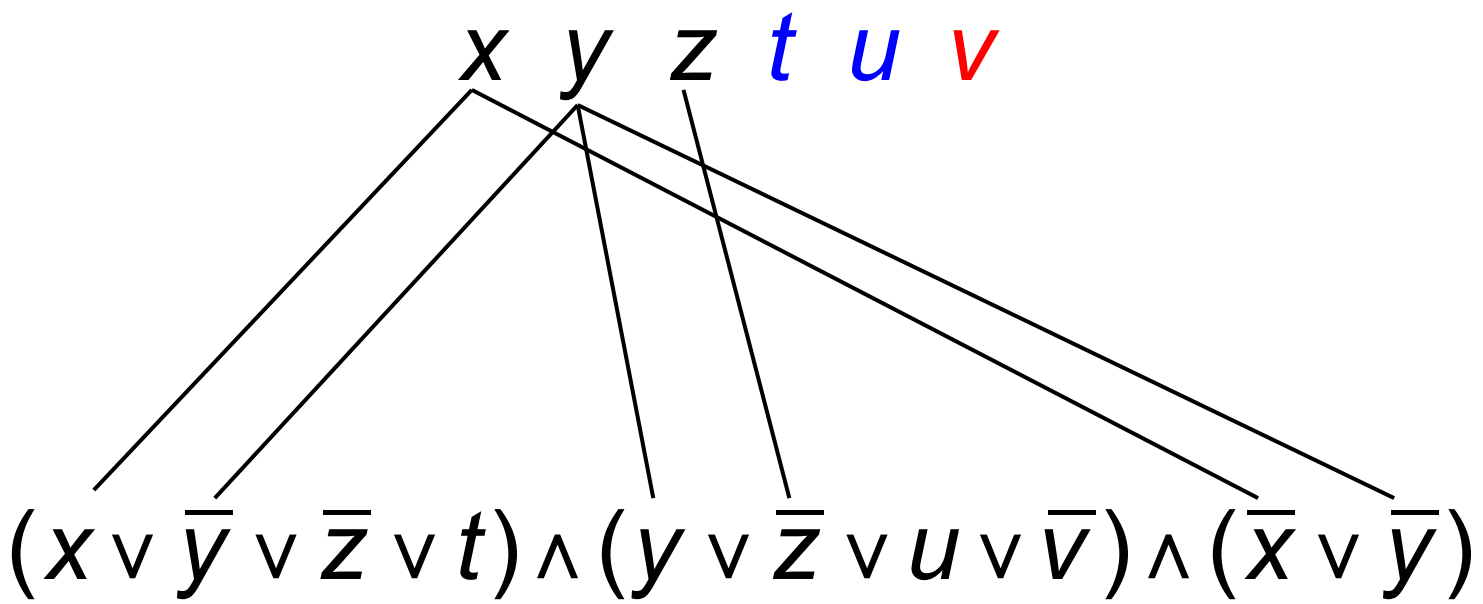
pour un littéral pointé devenant 0 dans une clause
on essaie de bouger le pointeur sur un littéral non pointé
Si pas de littéral indéfini ou 1, et si l'autre littéral pointé vaut 0, alors on a atteint une contradiction
(*y* mis à 1 ⇒ la 3e clause devient unitaire, contradiction)

2-Literal Watching (zChaff)



pour un littéral pointé devenant 0 dans une clause
on essaie de bouger le pointeur sur un littéral non pointé
Si pas de littéral indéfini ou 1, et si l'autre littéral pointé vaut 0, alors on a atteint une contradiction
(*y* mis à 1 ⇒ la 3e clause devient unitaire, contradiction)

2-Literal Watching (zChaff)



Retour arrière : rien à faire sur les pointeurs !
(ici retour arrière sur z, en défaisant *z*, *y* et *x*)

Résumé (addition par rapport à l'exposé)

- On fonce sur les conflits, en dépensant le moins d'énergie possible
- Les algorithmes sont centrés sur la structure de données des doubles pointeurs
- On ne regarde même pas si les clauses deviennent vraies tant qu'on n'a pas affecté toutes les variables, car ça coûterait de l'énergie
- Donc les valeurs fournies en cas de SAT ne sont pas minimales (mais il est facile de les minimiser)
- Certains solveurs n'essaient même pas d'être complets, i.e. de détecter UNSAT, pour mieux viser SAT.

Agenda

1. Le calcul Booléen (rappels)
2. La forme normale conjonctive (CNF)
3. Exemples de problèmes SAT
4. De la résolution à Davis and Putnam
5. CDCL = Conflict Driven Clause Learning
6. Two Literal Watching
7. Conclusion

Quelle est donc la complexité réelle ?

- Pire cas : exponentielle dans tous les algos connus
  - mais les pires cas ne se produisent pas fréquemment
    se raréfieraient-ils avec la taille?

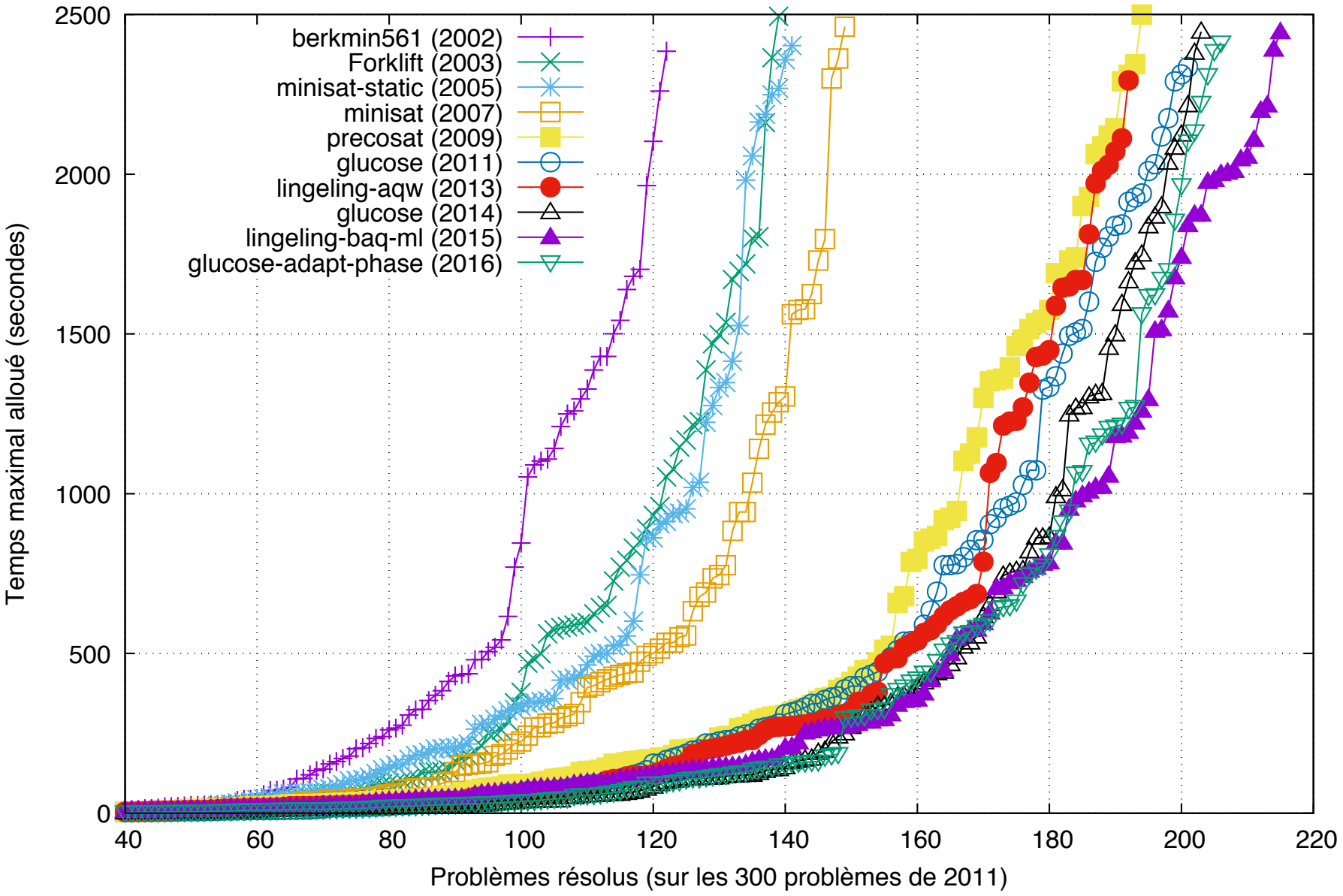
- Expression aléatoire
  - des propriétés de physique statistique !
    (très différent des BDDs, exponentielle pour eux)
  - accélérer l'aléatoire ne semble pas utile en pratique

- Cas pratiques « industriels »
  - souvent traitables à grande échelle
    mais pourquoi donc ? y a-t-il une géométrie cachée ?

Choix stratégiques (séminaire de Laurent Simon)

- Sur quels littéraux faire les choix ?
- Sur quel choix revenir en arrière en cas d'échec ?
- Combien de clauses garder ?
- Faut-il de temps en temps redémarrer au début en gardant les bonnes clauses apprises ? Ou de façon aléatoire?
- Quel est le bon équilibre entre exploration brutale et calcul savant ?

Des progrès considérables en 15 ans !



Conclusion

- SAT est un des problèmes combinatoires les plus fondamentaux de l'informatique
- Des solutions radicalement nouvelles sont apparues récemment (Grasp, zChaff, Glucose, etc.), avec des idées créatives à la pelle
- Le sujet étant très expérimental, la compétition annuelle joue un rôle fondamental (cf. séminaire de Laurent Simon)
- Des problèmes à 100 variables restent impossibles, mais d'autres à des millions de variables se résolvent tous les jours dans des domaines variés, y compris industriels

Et on ne sait toujours pas pourquoi
les solveurs sont aussi efficaces en pratique !

Bibliographie

J. A. Robinson : *A Machine-Oriented Logic Based on the Resolution Principle*.
J. Assoc. Comput. Mach. 12, 23-41, 1965.

G.S. Tseytin : *On the Complexity of Derivation in Propositional Calculus.*
In: Slisenko, A.O. (ed.) Studies in Constructive Mathematics and Mathematical Logic, Part II, Seminars in Mathematics, pp. 115–125. Steklov Mathematical Institute (1970).

J. P. Marques-Silva et K.A. Sakallah: *A search algorithm for propositional satisfiability*
IEEE Transactions on Computers, vol. 48, n. 5, mai 1999.

M.W.Moskewicz, C.F. Madigan, Yig Zhao, Linata Zhang et S. Malik :
Chaff : Engineering an Efficient SAT Solver.
Proc. DAC'01, 38th annual Design Automation Conference

S. Malik : *Boolean Satisfiability : From Theoretical Hardness to Practical Success*
Communications of the ACM, vol. 52, n. 8l, pp. 76-82.

D. Knuth : *Satisfiability*
The Art of Computer Programming, vol. 4, Fascicle 6, Addison-Wesley 2016