Deux fragments polynomiaux complets pour le problème de la validité des formules booléennes quantifiées

Article Jeune Chercheur JFPC'06

Florian Letombe
CRIL, CNRS FRE 2499
Université d'Artois, Lens

9 juin 2006

QBF : définition formelle

Définition (QBF)

Une QBF Π est une expression de la forme

$$Q_1 X_1 \ldots Q_n X_n . \Phi, \qquad (n \geq 0)$$

- $Q_i (0 \leq i \leq n)$ un quantificateur existentiel $\exists$ ou universel $\forall$
- $X_1 \ldots X_n$ ensembles de variables propositionnelles
- Φ une formule propositionnelle sur ces variables

Validité d'une QBF

Existence d'une stratégie gagnante dans un jeu contre la nature ($\forall$)

Exemple

$$\forall\ x\ \exists\ y_1,\ y_2.$$
$$[(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge$$
$$(\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$

Validité d'une QBF

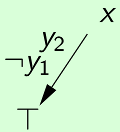
Existence d'une stratégie gagnante dans un jeu contre la nature ($\forall$)

Exemple

$$\forall\, x\; \exists\, y_1,\; y_2 .$$

$$[(y_1 \vee y_2) \cancel{\wedge (\neg y_2 \vee x)} \wedge$$

$$(\neg y_1 \vee \neg y_2) \wedge (y_2 \cancel{\vee \neg x})]$$



Validité d'une QBF

Existence d'une stratégie gagnante dans un jeu contre la nature ($\forall$)

Exemple

$$\forall\ x\ \exists\ y_1,\ y_2.$$

$$[(y_1 \vee y_2) \wedge (\neg y_2 \cancel{\vee x}) \wedge$$

$$(\neg y_1 \vee \neg y_2) \cancel{\wedge (y_2 \vee \neg x)}]$$

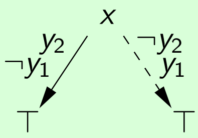
$$\begin{tikzcd} & x \arrow[dl, "y_2\ \neg y_1"'] \arrow[dr, dashed, "\neg y_2\ y_1"] & \\ \top & & \top \end{tikzcd}$$

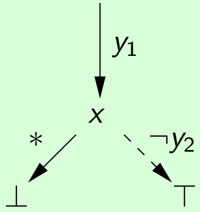
Validité d'une QBF

Existence d'une stratégie gagnante dans un jeu contre la nature ($\forall$)

Exemple

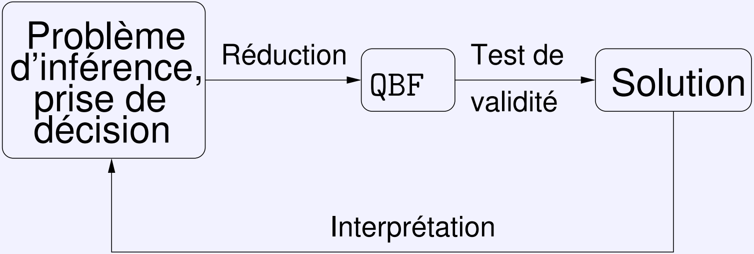
$$\forall\, x\ \exists\, y_1,\ y_2.\ [(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge (\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)] \not\equiv \exists\, y_1\ \forall\, x\ \exists\, y_2.\ [(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge (\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$





Le problème VAL($\mathtt{QPROP}_{PS}$)

- Généralisation de SAT
- Problème **PSPACE**-complet canonique [Stockmeyer & Meyer 1973]
- Nombreuses applications en IA : planification, raisonnement non monotone, inférence paraconsistante, etc

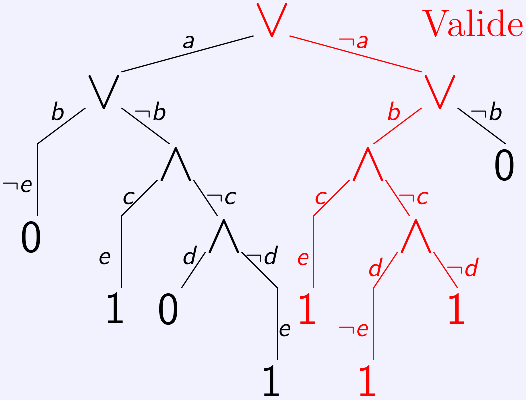


Classes polynomiales : présentation

- Beaucoup étudié depuis longtemps dans la communauté SAT
- Étudié récemment pour QBF [Gent & Rowley 2002]
- Deux étapes :
  - Reconnaissance polynomiale
  - Résolution polynomiale
- Exemples de classes polynomiales : Krom [Gent & Rowley 2002], Horn [Kleine-Büning *et al.* 1995], Horn renommable, ...

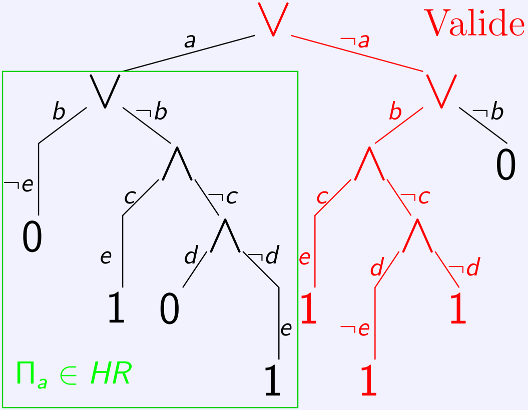
Classes polynomiales : motivation

$$\Pi = \exists\ a, b\ \forall\ c, d\ \exists\ e\ .$$
$$\left[\begin{array}{ll} (b \vee c \vee \neg d \vee e) & \wedge \\ (\neg a \vee \neg b \vee \neg e) & \wedge \\ (\neg a \vee d \vee e) & \wedge \\ (\neg c \vee e) & \wedge \\ (a \vee b \vee \neg c) & \wedge \\ (c \vee \neg d \vee \neg e) & \end{array}\right]$$



Classes polynomiales : motivation

$$\Pi = \exists\ a, b\ \forall\ c, d\ \exists\ e\ .\ \left[\begin{array}{ll} (b \vee c \vee \neg d \vee e) & \wedge \\ (\neg a \vee \neg b \vee \neg e) & \wedge \\ (\neg a \vee d \vee e) & \wedge \\ (\neg c \vee e) & \wedge \\ (a \vee b \vee \neg c) & \wedge \\ (c \vee \neg d \vee \neg e) & \end{array}\right]$$



Formules Horn CNF quantifiées

Définition *(Formule Horn CNF quantifiée)*

- Clause de Horn : clause comprenant au plus un littéral positif
- Formule Horn CNF quantifiée : QBF dont la matrice ne contient que des clauses de Horn

Exemple

$$\Pi_1 = \forall\ a, b\ \exists\ c, d\ \forall\ e\ .\ \left[\begin{array}{ll} (a \vee \neg b \vee \neg d \vee \neg e) & \wedge \\ (\neg a \vee c \vee \neg e) & \wedge \\ (\neg b \vee \neg c \vee d) & \wedge \\ (\neg a \vee \neg c \vee \neg d) & \end{array}\right]$$

Formules Horn renommables CNF quantifiées

Définition *(Formules Horn renommables CNF quantifiées)*

- Renommer = changer la polarité d'un littéral
- Une formule est Horn renommable ssi il existe un renommage uniforme de ses variables tq sa matrice devienne Horn après renommage

Exemple

$$\Pi_2 = \forall\ a, b\ \exists\ c, d\ \forall\ e\ .\ \left[\begin{array}{ll} (a \vee b \vee \neg d \vee \neg e) & \wedge \\ (\neg a \vee \neg c \vee \neg e) & \wedge \\ (b \vee c \vee d) & \wedge \\ (\neg a \vee c \vee \neg d) & \end{array}\right] \qquad \Pi_1 = \forall\ a, b\ \exists\ c, d\ \forall\ e\ .\ \left[\begin{array}{ll} (a \vee \neg b \vee \neg d \vee \neg e) & \wedge \\ (\neg a \vee c \vee \neg e) & \wedge \\ (\neg b \vee \neg c \vee d) & \wedge \\ (\neg a \vee \neg c \vee \neg d) & \end{array}\right]$$

Π_2 est Π_1 dans laquelle on a renommé b et c

Les fragments MODS et CI

Définition *(MODS et CI)*

- MODS : DNF contenant des formules $\Sigma = \gamma_1 \vee \ldots \vee \gamma_k$ tq, pour chaque terme γ_i $(i \in 1 \ldots k)$ et pour chaque variable $x \in Var(\Sigma)$, γ_i contient x ou $\neg x$
- CI : CNF contenant des formules $\Sigma = \gamma_1 \wedge \ldots \wedge \gamma_k$ tq, pour chaque clause γ_i $(i \in 1 \ldots k)$ et pour chaque variable $x \in Var(\Sigma)$, γ_i contient x ou $\neg x$

Exemple

Formule MODS

$$\begin{array}{ll} (a \wedge b \wedge c) & \vee \\ (\neg a \wedge \neg b \wedge c) & \vee \\ (\neg a \wedge b \wedge c) & \vee \\ (\neg a \wedge \neg b \wedge \neg c) & \end{array}$$

Formule CI

$$\begin{array}{ll} (a \vee b \vee c) & \wedge \\ (\neg a \vee \neg b \vee c) & \wedge \\ (\neg a \vee b \vee c) & \wedge \\ (\neg a \vee \neg b \vee \neg c) & \end{array}$$

Les fragments $\mathtt{ODNF}_<$ et $\mathtt{OCNF}_<$

Définition ($\mathtt{ODNF}_<$ et $\mathtt{OCNF}_<$)

- $\mathtt{ODNF}_<$: $\mathtt{DNF}$ $\Sigma = \gamma_1 \vee \ldots \vee \gamma_k$ où chaque terme satisfiable γ_i ($i \in 1 \ldots k$) de Σ est tq pour tout $j \in 1 \ldots n$, si x_j ou $\neg x_j$ apparaît dans γ_i, alors pour tout l tel que $1 \leq l < j$, x_l ou $\neg x_l$ apparaît dans γ_i
- $\mathtt{OCNF}_<$: $\mathtt{CNF}$ $\Sigma = \gamma_1 \wedge \ldots \wedge \gamma_k$ où chaque clause non tautologique γ_i ($i \in 1 \ldots k$) de Σ est tq pour tout $j \in 1 \ldots n$, si x_j ou $\neg x_j$ apparaît dans γ_i, alors pour tout l tel que $1 \leq l < j$, x_l ou $\neg x_l$ apparaît dans γ_i

Exemple

$PS = \{a, b, c\}$, $<$ tq $a < b < c$

Formule $\mathtt{ODNF}_<$

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

Formule $\mathtt{OCNF}_<$

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

Résultats de complexité

Définition *(Préfixe compatible)*

$<$ est une extension de l'ordre des variables induit par le préfixe de la $\mathtt{QBF}$

Proposition *(Polynomialité)*

- Les restrictions de $\textsc{val}(\mathtt{QODNF}_<)$ et $\textsc{val}(\mathtt{QOCNF}_<)$ aux instances ayant un préfixe compatible avec $<$ sont dans **P**
- $\textsc{val}(\mathtt{QMODS})$ et $\textsc{val}(\mathtt{QCI})$ sont dans **P**

Un cas polynomial : $\textsc{val}(\texttt{QODNF}_<)$

Ordre lexicographique choisi sur les variables : $a < b < c < d < e$

$$\Pi_3 = \exists\ a, b\ \forall\ c, d\ \exists\ e\ .$$
$$\begin{bmatrix} (a \wedge \neg b \wedge \neg c \wedge \neg d \wedge \neg e) & \vee \\ (\neg a \wedge b \wedge c \wedge \neg d \wedge e) & \vee \\ (\neg a \wedge b \wedge c \wedge d) & \vee \\ (\neg a \wedge b \wedge \neg c) & \vee \\ (a \wedge \neg b \wedge c) & \vee \\ (a \wedge b) & \vee \end{bmatrix}$$

Un cas polynomial : $\textsc{VAL}(\texttt{QODNF}_<)$

Ordre lexicographique choisi sur les variables : $a < b < c < d < e$

$$\begin{array}{l} \Pi_3 = \exists\ a, b\ \forall\ c, d\ . \\ \left[\begin{array}{ll} (a \wedge \neg b \wedge \neg c \wedge \neg d) & \vee \\ (\neg a \wedge b \wedge c \wedge \neg d) & \vee \\ (\neg a \wedge b \wedge c \wedge d) & \vee \\ (\neg a \wedge b \wedge \neg c) & \vee \\ (a \wedge \neg b \wedge c) & \vee \\ (a \wedge b) & \vee \end{array} \right] \end{array}$$

$W = \{\gamma \mid x \notin Var(\gamma)\}$ en noir
$S = \{\gamma \mid switch(\gamma, x) \in \Phi\}$ en vert
S' : tous les autres termes en rouge

Un cas polynomial : $\mathrm{VAL}(\mathtt{QODNF}_<)$

Ordre lexicographique choisi sur les variables : $a < b < c < d < e$

$$\Pi_3 = \exists\ a, b\ \forall\ c\ .\ \left[\begin{array}{ll} (\neg a \wedge b \wedge c) & \vee \\ (\neg a \wedge b \wedge \neg c) & \vee \\ (a \wedge \neg b \wedge c) & \vee \\ (a \wedge b) & \vee \end{array}\right]$$

$W = \{\gamma \mid x \notin Var(\gamma)\}$ en noir
$S = \{\gamma \mid switch(\gamma, x) \in \Phi\}$ en vert
S' : tous les autres termes en rouge

Un cas polynomial : $\mathrm{VAL}(\mathtt{QODNF}_{<})$

Ordre lexicographique choisi sur les variables : $a < b < c < d < e$

$$\Pi_3 = \exists\ a, b\ .\ \left[\begin{array}{ll} (\neg a \wedge b) & \vee \\ (a \wedge b) & \vee \end{array}\right]$$

Un cas polynomial : $\mathrm{VAL}(\mathtt{QODNF}_{<})$

Ordre lexicographique choisi sur les variables : $a < b < c < d < e$

$$\Pi_3 = \exists\, a\ .\ \begin{bmatrix} \neg a & \vee \\ a & \end{bmatrix}$$

Donc Π_3 est valide

Les prouveurs

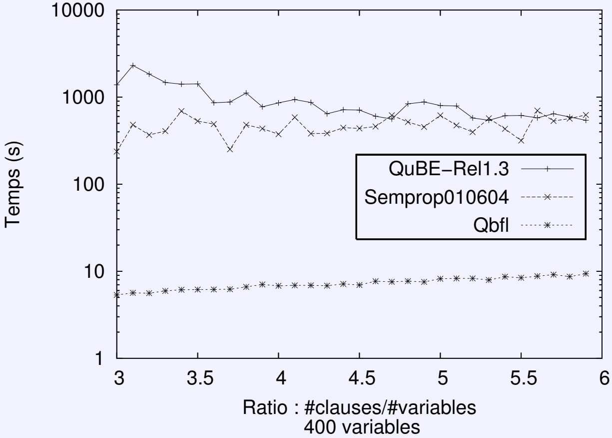
- PIV 3GHz, 1.5Go de RAM, Linux Fedora
- Qbfl : mon prouveur, version 1.7 (`http://www.cril.univ-artois.fr/~letombe/qbfl`)
- OpenQBF : G. Audemard, D. Le Berre et O. Roussel Dernière Sun Java VM
- QuBE-Rel : Giunchiglia *et al.* 2001, version 1.3 (`http://www.star.dist.unige.it/~qube`)
- Semprop : Letz 2002, version 010604 (`http://www4.in.tum.de/~letz/semprop`)

Les instances

- Fragments incomplets
  - Formules Horn CNF soumises à l'évaluation comparative des prouveurs QBF 2005
  - Formules Horn renommables CNF soumises à l'évaluation comparative des prouveurs QBF 2005
- Fragments complets CI et $\texttt{OCNF}_<$
  - Générés aléatoirement (distribution uniforme)
  - Taille des clauses de $\texttt{OCNF}_<$ aléatoire
  - $n = 100, 200, 300, 400, 500, 1000$ variables
  - $m = n \times 4$ clauses
  - 12 alternances de quantificateurs (dont 11 de taille n *div* 11)

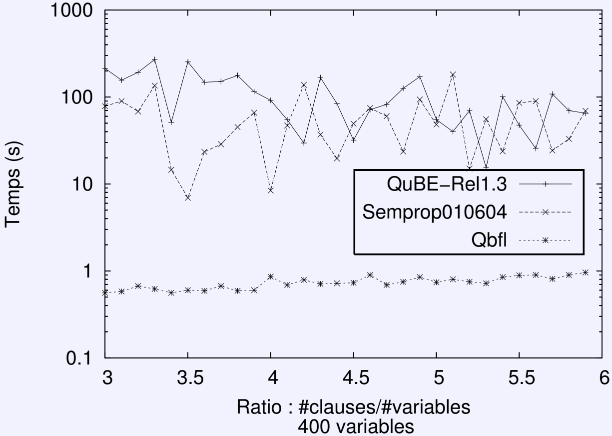
Formules Horn CNF

- 1000 instances
- Temps cumulé sur les instances résolues, TimeOut : 60 s
- 400 variables, 1200 à 2360 clauses



Formules Horn renommables CNF

- 100 instances
- Temps cumulé sur les instances résolues, TimeOut : 60 s
- 400 variables, 1200 à 2360 clauses

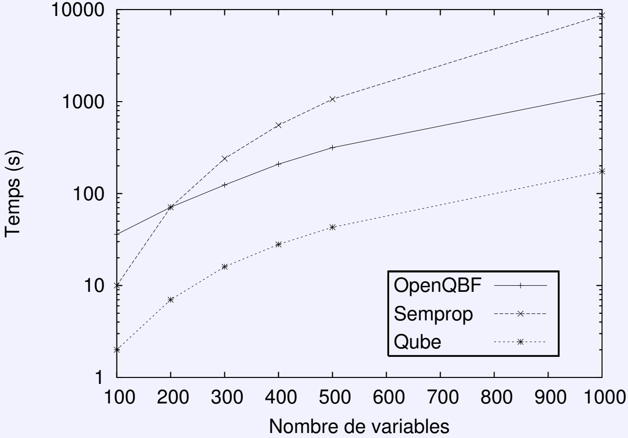


Prouveurs soumis à l'évaluation QBF 2005 : résultats sur des formules Horn CNF

Prouveur	# vrai /81	# faux /82	Total /163
ssolve	78	77	155
semprop	54	55	**109**
yquaffle	40	35	75
GRL	7	59	66
qbfbdd	42	15	57
QchaffLearn	2	53	55
walkqsat	0	39	39
sKizzo 0.4	0	33	33
sKizzo 0.5	0	32	32
openqbf	0	20	**20**
quantor	0	10	10
QMRes	0	0	0

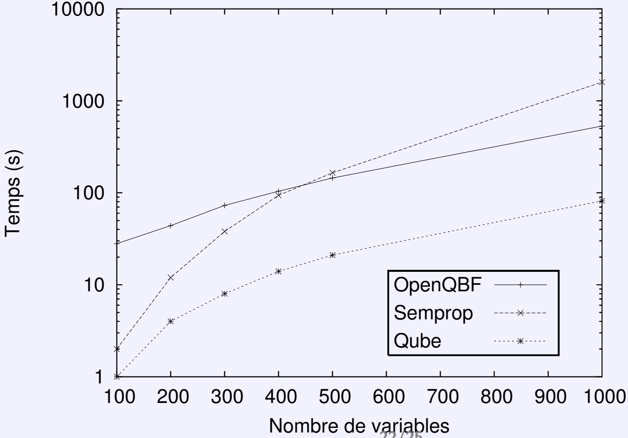
Formules CI

- 100 instances par point
- Temps cumulé sur les instances résolues
- $n = 100, 200, 300, 400, 500, 1000$ variables, $m = 4 \times n$ clauses



Formules $\mathtt{OCNF}_<$

- 100 instances par point
- Temps cumulé sur les instances résolues
- $n = 100, 200, 300, 400, 500, 1000$ variables, $m = 4 \times n$ clauses



Discussion

- Conclusion
  - Deux nouveaux fragments polynomiaux pour le problème de la validité
  - Mauvais comportement des prouveurs QBF actuels sur les fragments polynomiaux incomplets
  - Très bon comportement de ces mêmes prouveurs sur les fragments polynomiaux complets
  - Utilité de reconnaître les fragments polynomiaux complets ?
- Perspectives
  - Répondre à cette dernière question
  - Découverte d'autres fragments (polynomiaux) complets ou incomplets
  - Analyse plus fine du préfixe

Merci de votre attention