Fragments Propositionnels pour la Compilation de Connaissances et Formules Booléennes Quantifiées

Sylvie Coste-Marquis Daniel Le Berre
Florian Letombe Pierre Marquis
CRIL, CNRS FRE 2499
Lens, Université d'Artois, France

25 janvier 2006

Le problème QBF

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

- Problème important mais de complexité élevée
- Une solution possible : les classes polynomiales
- Instances de ces classes polynomiales difficiles pour les prouveurs QBF actuels

$\mathcal{QBF}$: définition formelle

Définition *($\mathcal{QBF}$)*

Une $\mathcal{QBF}$ Π est une expression de la forme

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

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

Validité d'une $\mathcal{QBF}$

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 $\mathcal{QBF}$

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

Exemple

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

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

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

$$x$$

$$y_2$$

$$\neg y_1$$

$$\top$$

Validité d'une $\mathcal{QBF}$

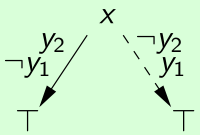
Existence d'une stratégie gagnante dans un jeu contre la nature (∀)

Exemple

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

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

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

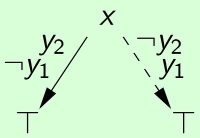


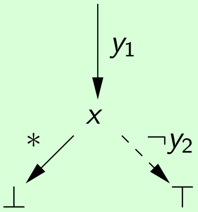
Validité d'une $\mathcal{QBF}$

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

Exemple

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



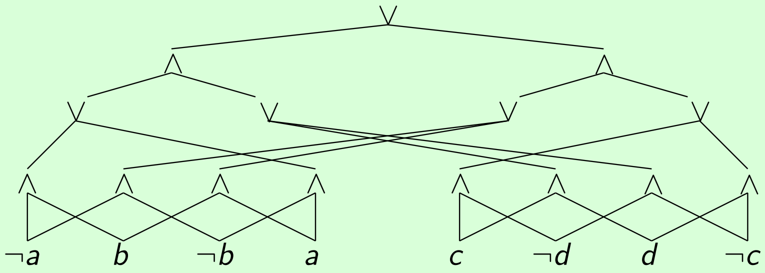


Définition ($\mathcal{NNF}$ [Darwiche 1999])

Une formule dans $\mathcal{NNF}_{PS}$ est un DAG où :

- chaque feuille est étiquetée avec Vrai, Faux, x or $\neg x$, $x \in PS$
- chaque nœud interne est étiqueté avec $\wedge$ ou $\vee$ et peut avoir un nombre arbitraire de fils

Exemple

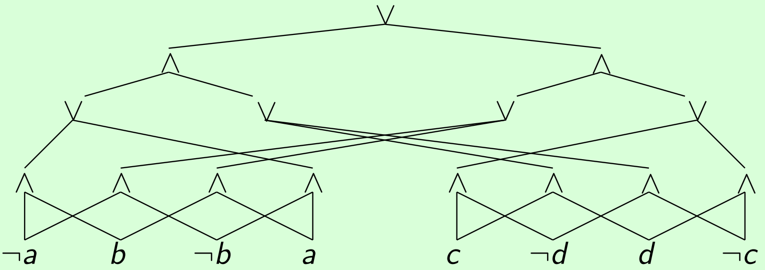


Propriétés [Darwiche 1999]

- Décomposabilité
- Déterminisme
- Uniformité
- Décision
- Ordonnancement

Fragments de $\mathcal{NNF}_{PS}$: exemples

Exemple

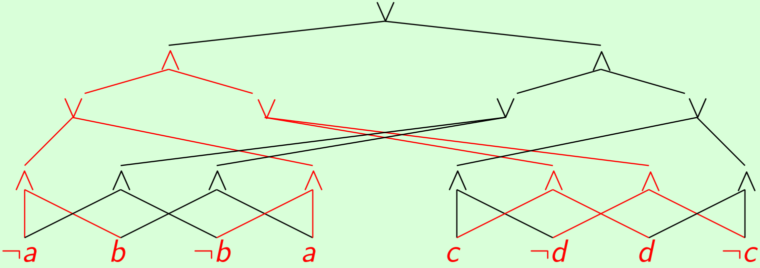


Fragments de $\mathcal{NNF}_{PS}$: exemples

Décomposabilité : si $C_1, \ldots, C_n$ sont les fils du nœud « et » C, alors $Var(C_i) \cap Var(C_j) = \emptyset$ pour $i \neq j$

Exemple

Décomposabilité

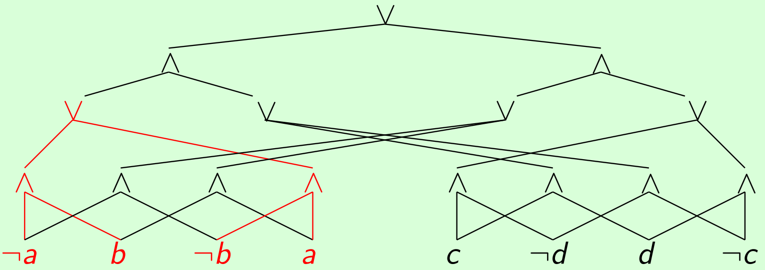


Fragments de $\mathcal{NNF}_{PS}$: exemples

Déterminisme : si $C_1, \ldots, C_n$ sont les fils du nœud « ou » C, alors $C_i \wedge C_j \models$ Faux pour $i \neq j$

Exemple

Déterminisme

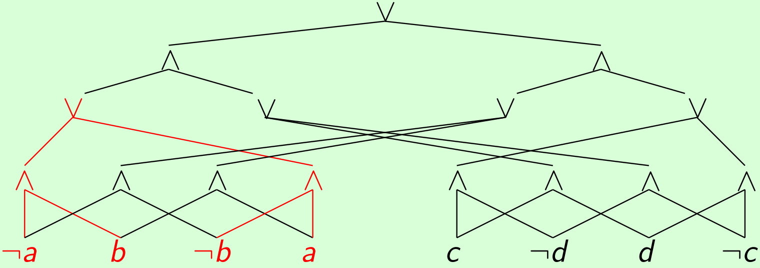


Fragments de $\mathcal{NNF}_{PS}$: exemples

Uniformité : si $C_1, \ldots, C_n$ sont les fils du nœud « ou » C, alors $Var(C_i) = Var(C_j)$

Exemple

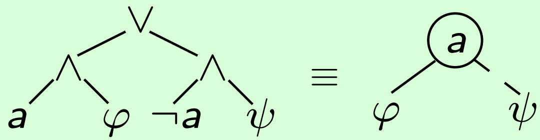
Uniformité



Fragments de $\mathcal{NNF}_{PS}$: exemples

Décision : nœud étiqueté par Vrai, Faux, ou un nœud « ou » de la forme $(a \wedge \varphi) \vee (\neg a \wedge \psi)$

Exemple



Fragments de $\mathcal{NNF}_{PS}$: définitions

Définition *(Fragments propositionnels [Darwiche & Marquis 2001])*

- $\mathcal{DNNF}$: $\mathcal{NNF}_{PS}$ + décomposabilité
- d-$\mathcal{DNNF}$: $\mathcal{NNF}_{PS}$ + décomposabilité et déterminisme
- sd-$\mathcal{DNNF}$: $\mathcal{NNF}_{PS}$ + décomposabilité, déterminisme et uniformité
- $\mathcal{FBDD}$: $\mathcal{NNF}_{PS}$ + décomposabilité et décision
- $\mathcal{OBDD}_{<}$: $\mathcal{NNF}_{PS}$ + décomposabilité, décision et ordonnancement
- $\mathcal{MODS}$: $\mathcal{DNF} \cap$ d-$\mathcal{DNNF}$ + uniformité

Impliqués premiers ($\mathcal{PI}$)

Définition *(Impliqué, Impliqué premier)*

- Une clause π est un impliqué de Σ ssi $\Sigma \models \pi$
- Une clause π est un impliqué premier de Σ ssi :
  - π est un impliqué de Σ, et
  - pour chaque impliqué π' de Σ, si $\pi' \models \pi$, alors $\pi \models \pi'$

Exemple

$$\Sigma = \begin{bmatrix} (\neg a \vee b) & \wedge \\ (\neg b \vee c) & \end{bmatrix}$$

Impliqués premiers de Σ :

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

Impliquants premiers ($\mathcal{IP}$)

Définition *(Impliquant, Impliquant premier)*

- Un terme π est un impliquant de Σ ssi $\pi \models \Sigma$
- Un terme π est un impliquant premier de Σ ssi :
  - π est un impliqué de Σ, et
  - pour chaque impliqué π' de Σ, si $\pi \models \pi'$, alors $\pi' \models \pi$

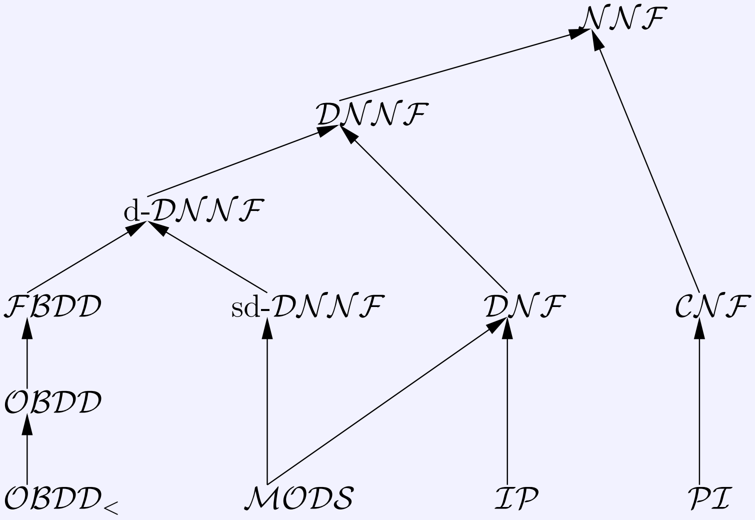
Exemple

$$\Sigma = \left[\begin{array}{ll} (\neg a \vee b) & \wedge \\ (\neg b \vee c) & \end{array} \right]$$

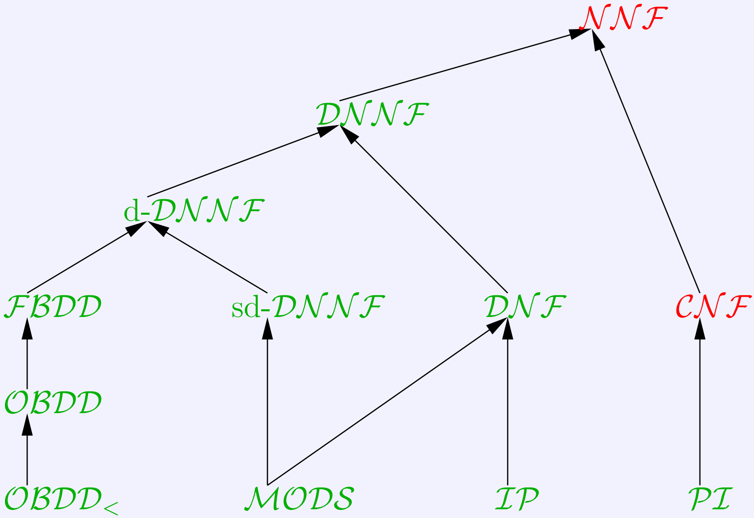
Impliquants premiers de Σ :

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

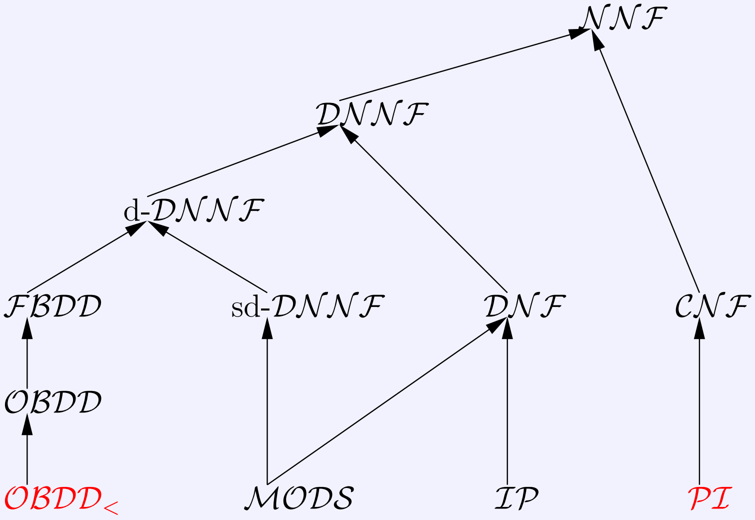
Inclusion des fragments [Darwiche & Marquis 2001]



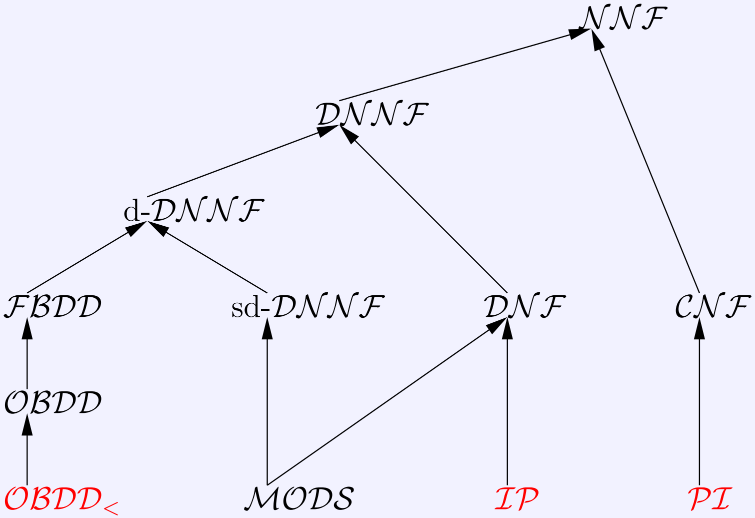
Complexité pour SAT [Darwiche & Marquis 2001]



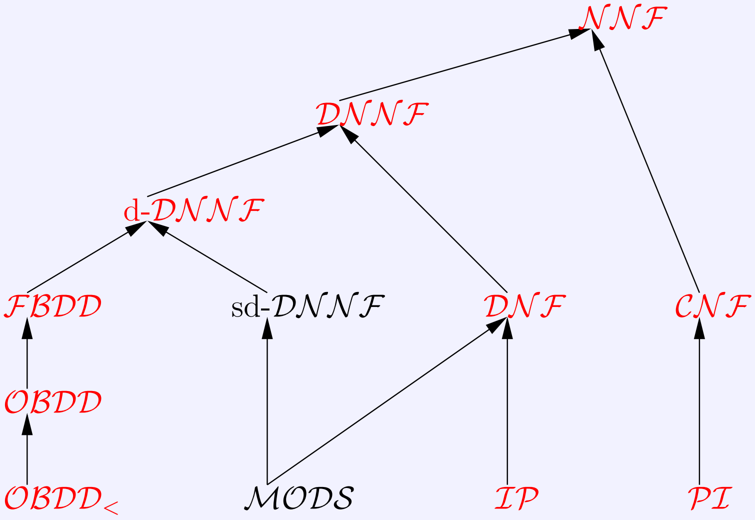
Résultats de complexité pour QBF



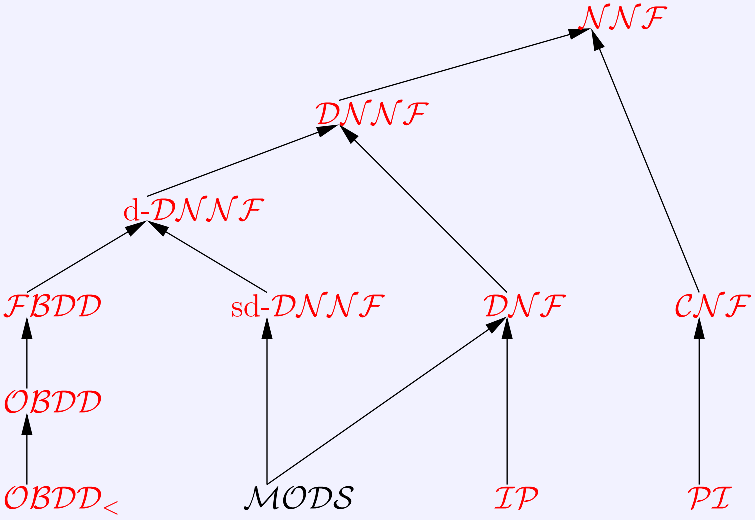
Résultats de complexité pour QBF



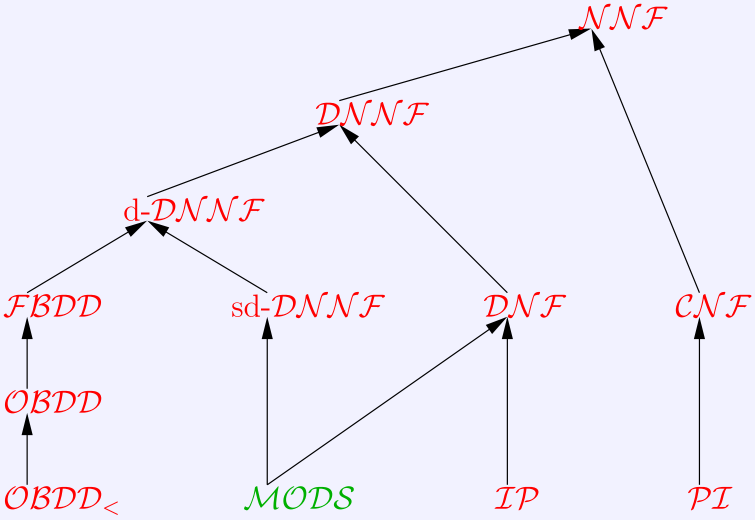
Résultats de complexité pour QBF



Résultats de complexité pour QBF



Résultats de complexité pour QBF



Résultats de complexité pour QBF

Fragment	**Complexité**
$PROP_{PS}$ (cas général)	PSPACE-c
$\mathcal{CNF}$	PSPACE-c
$\mathcal{DNF}$	PSPACE-c
d-$\mathcal{DNNF}$	PSPACE-c
sd-$\mathcal{DNNF}$	PSPACE-c
$\mathcal{DNNF}$	PSPACE-c
$\mathcal{FBDD}$	PSPACE-c
$\mathcal{OBDD}_<$	PSPACE-c
$\mathcal{OBDD}_<$ (préfixe compatible)	$\in$ P
$\mathcal{PI}$	PSPACE-c
$\mathcal{IP}$	PSPACE-c
$\mathcal{MODS}$	$\in$ P

$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

- Préfixe compatible : $<$ est une extension de l'ordre des variables induit par le préfixe de la $\mathcal{QBF}$

- Éliminer les quantificateurs du plus interne au plus externe
  - Éliminer les quantificateurs existentiels
  - Éliminer les quantificateurs universels ($\forall x \equiv \neg\exists x \neg$)
  - Réduire (élimination et fusion [Bryant 1986]) l'$\mathcal{OBDD}_<$ à chaque étape d'élimination

- Remarque : la négation se fait en temps constant dans un $\mathcal{OBDD}_<$

$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

$$\Sigma = \exists x \, \forall y \, \exists z \, . \, \phi$$

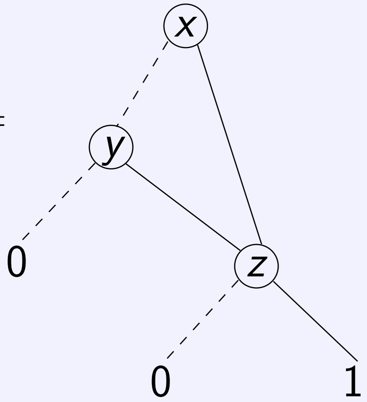
$$\phi \equiv (x \vee y) \wedge z$$

$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

$$\Sigma = \exists x \; \forall y \; \exists z \; . \; \phi$$

$$\phi \equiv (x \vee y) \wedge z$$

$\phi =$

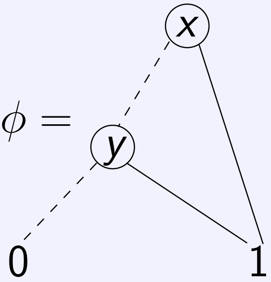


$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

$$\Sigma = \exists x \; \forall y \; \exists z \; . \; \phi$$

$$\phi \equiv (x \vee y) \wedge z$$

$\exists z \; . \; \phi =$

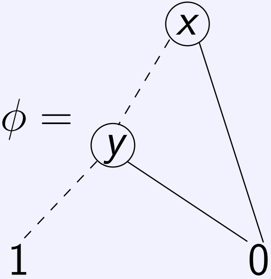


$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

$$\Sigma = \exists x \; \forall y \; \exists z \; . \; \phi$$

$$\phi \equiv (x \vee y) \wedge z$$

$$\neg \exists z \; . \; \phi =$$

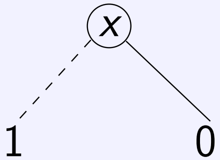


$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

$$\Sigma = \exists x \; \forall y \; \exists z \; . \; \phi$$

$$\phi \equiv (x \vee y) \wedge z$$

$$\exists y \neg \exists z \; . \; \phi =$$

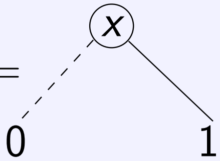


$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

$$\Sigma = \exists x \; \forall y \; \exists z \; . \; \phi$$

$$\phi \equiv (x \vee y) \wedge z$$

$$\underbrace{\neg\exists y \neg}_{\forall y} \exists z \; . \; \phi =$$



$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

$$\Sigma = \exists x \ \forall y \ \exists z \ . \ \phi$$

$$\phi \equiv (x \vee y) \wedge z$$

$$\exists x \forall y \exists z \ . \ \phi = 1$$

$\Rightarrow \Sigma$ est valide

Conclusion

- Présentation de fragments propositionnels
- Mise en évidence de nombreux fragments intraitables
- Quelques résultats de traitabilité (sous des conditions très restrictives)
- Mise en évidence des problèmes des prouveurs QBF actuels sur des instances polynomiales

Perspectives

- Découverte d'autres fragments (polynomiaux) complets ou incomplets
  - Algorithme de détection de ces fragments
  - Algorithme de résolution de ces fragments
- Intégrer de nouvelles classes polynomiales à notre prouveur
- Analyse plus fine du préfixe

Questions ?