De la validité des formules booléennes quantifiées

étude de complexité et exploitation de classes traitables au sein d'un prouveur QBF

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

Encadrants :
Sylvie Coste-Marquis
Daniel Le Berre
Pierre Marquis (Directeur de thèse)

13 décembre 2005

Introduction

Introduction

Complexité de QBF pour certains fragments propositionnels

- Présentation des fragments
- Aperçu des preuves
- Résultats de complexité
- Un cas polynomial

Exploitation de classes traitables au sein d'un prouveur QBF

- Algorithmique
- Classes polynomiales
- Qbfl
- L'heuristique Δ
- Résultats expérimentaux

Conclusion et perspectives

$\mathcal{QBF}$: 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 $\mathcal{QBF}$

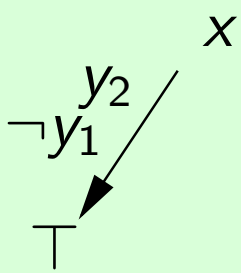
Existence d'une stratégie gagnante dans un jeu contre la nature ($\forall$)

Exemple

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

$$[(y_1 \vee y_2) \wedge (\neg y_1 \vee x) \wedge$$

$$(\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$



Validité d'une $\mathcal{QBF}$

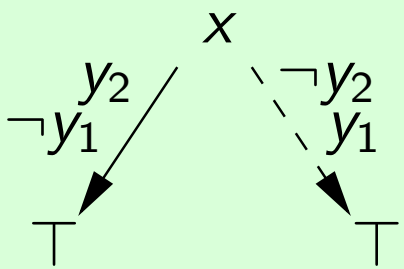
Existence d'une stratégie gagnante dans un jeu contre la nature ($\forall$)

Exemple

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

$$[(y_1 \vee y_2) \wedge (\neg y_2 \not\vee x) \wedge$$

$$(\neg y_1 \vee \neg y_2) \wedge (y_2 \not\vee \top x)]$$



Validité d'une $\mathcal{QBF}$

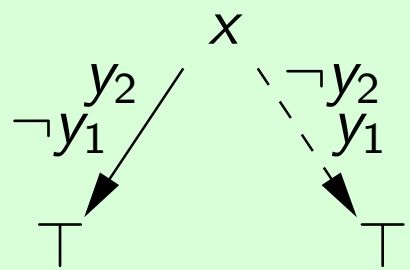
Existence d'une stratégie gagnante dans un jeu contre la nature ($\forall$)

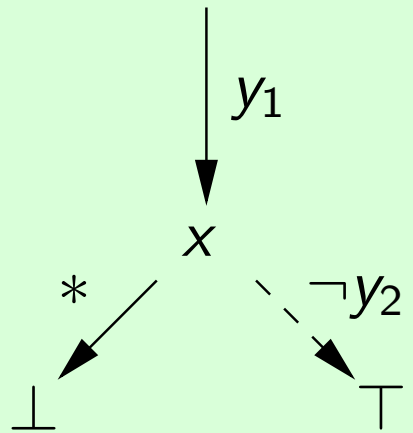
Exemple

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

$$\not\equiv$$

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

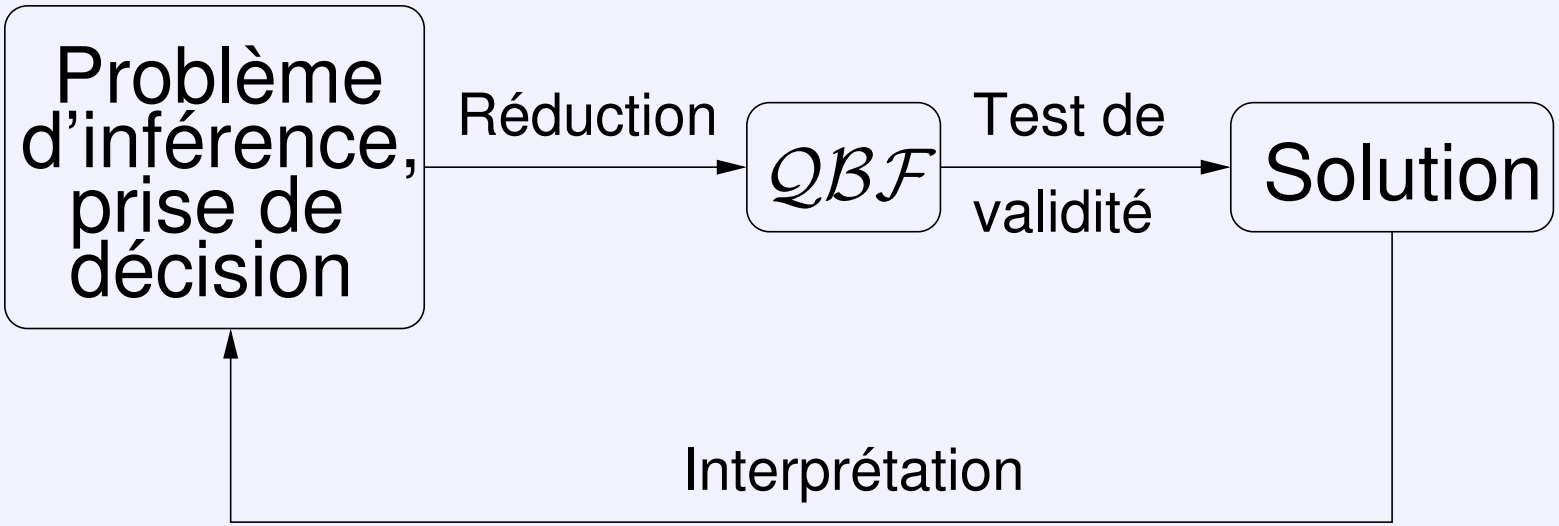




Le problème QBF

- Généralisation de SAT
- Problème **PSPACE**-complet canonique

  [Stockmeyer & Meyer 1973]
- Nombreuses applications en IA : planification, raisonnement non monotone, inférence paraconsistante, abduction, etc



Énoncé des contributions

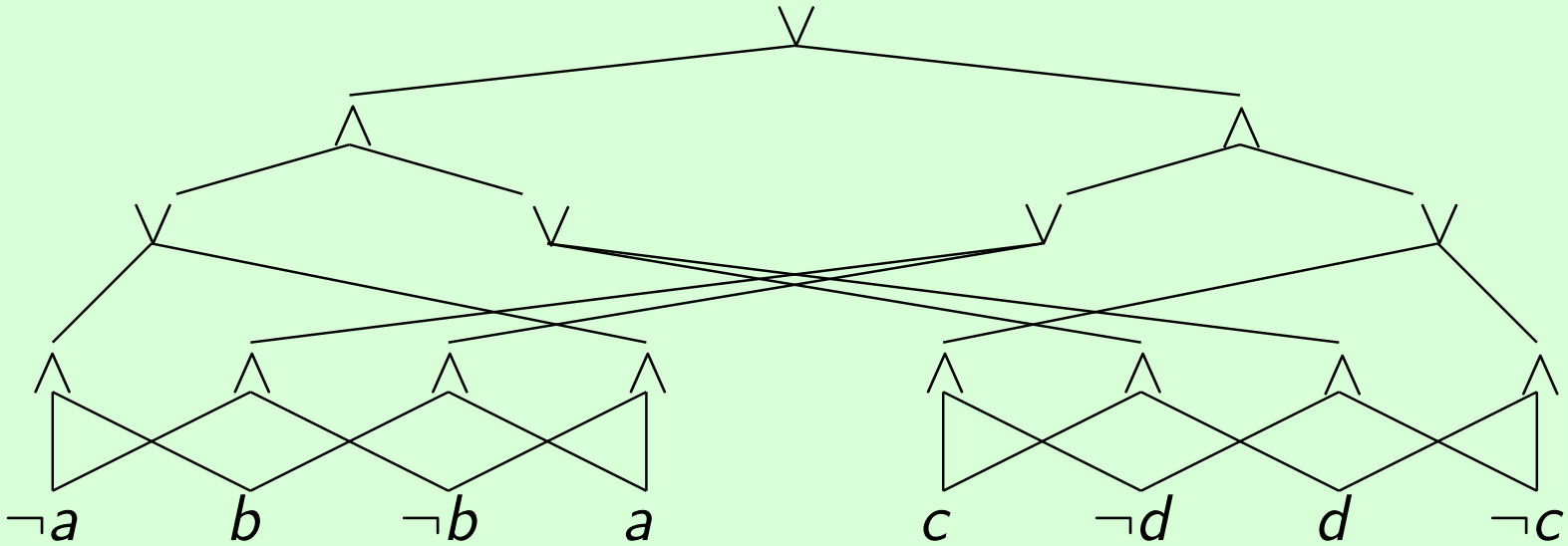
- Problème important mais de complexité élevée
- Quelles restrictions du problème le rendent décidable en temps polynomial ?
- Contributions :
  - étude de restrictions de $\mathcal{QBF}$ dont la matrice appartient à des fragments complets pour la logique propositionnelle
  - exploitation de fragments incomplets mais traitables au sein d'un prouveur QBF général

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

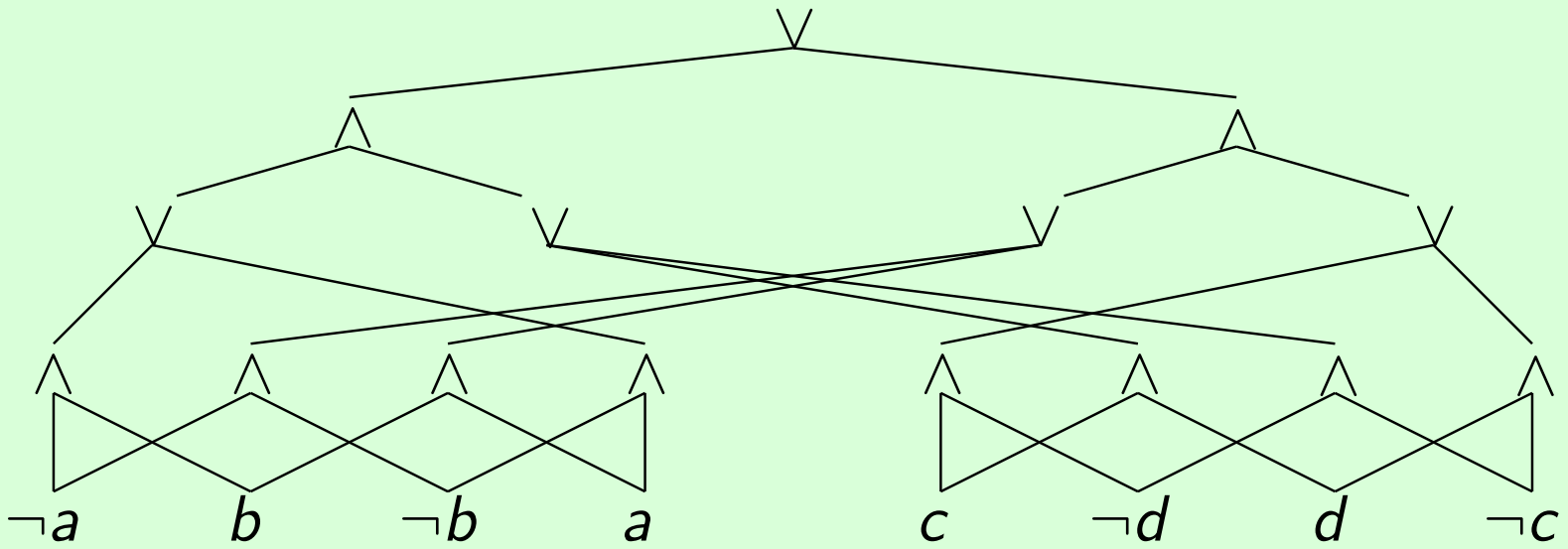


Propriétés [Darwiche 1999]

- Décomposabilité
- Déterminisme
- Uniformité
- Décision
- Ordonnancement

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

Exemple

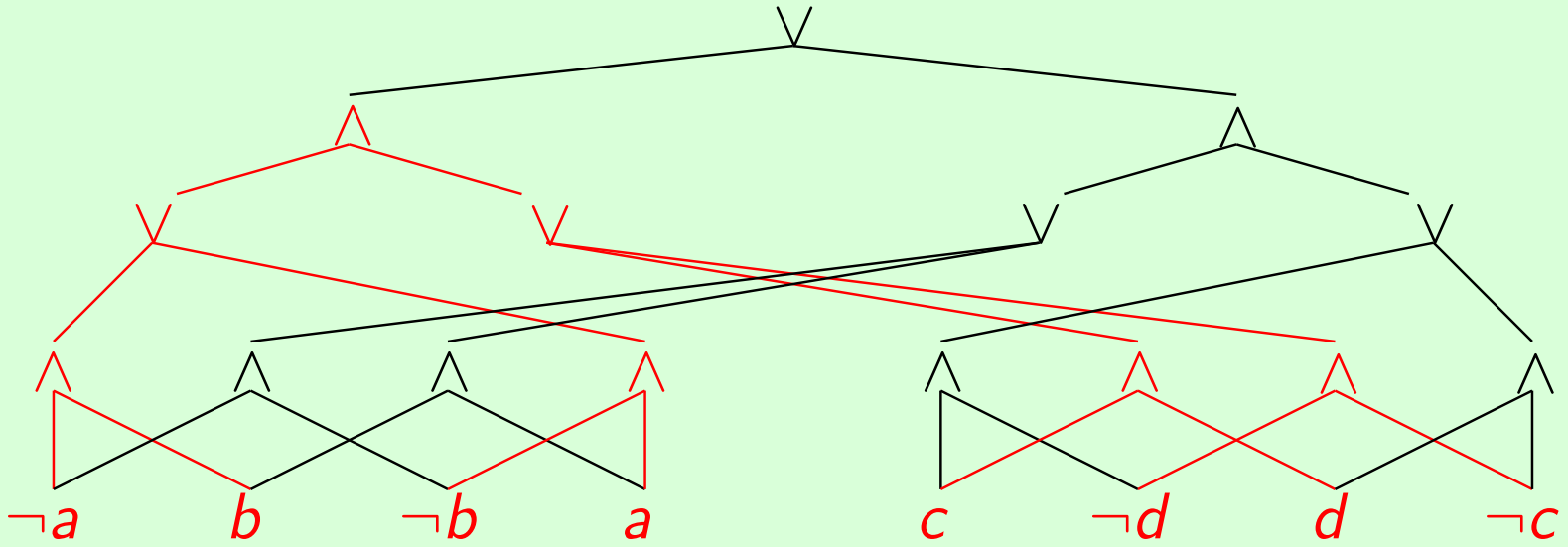


Fragments de $\mathcal{NNF}_{PS}$: exemples

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

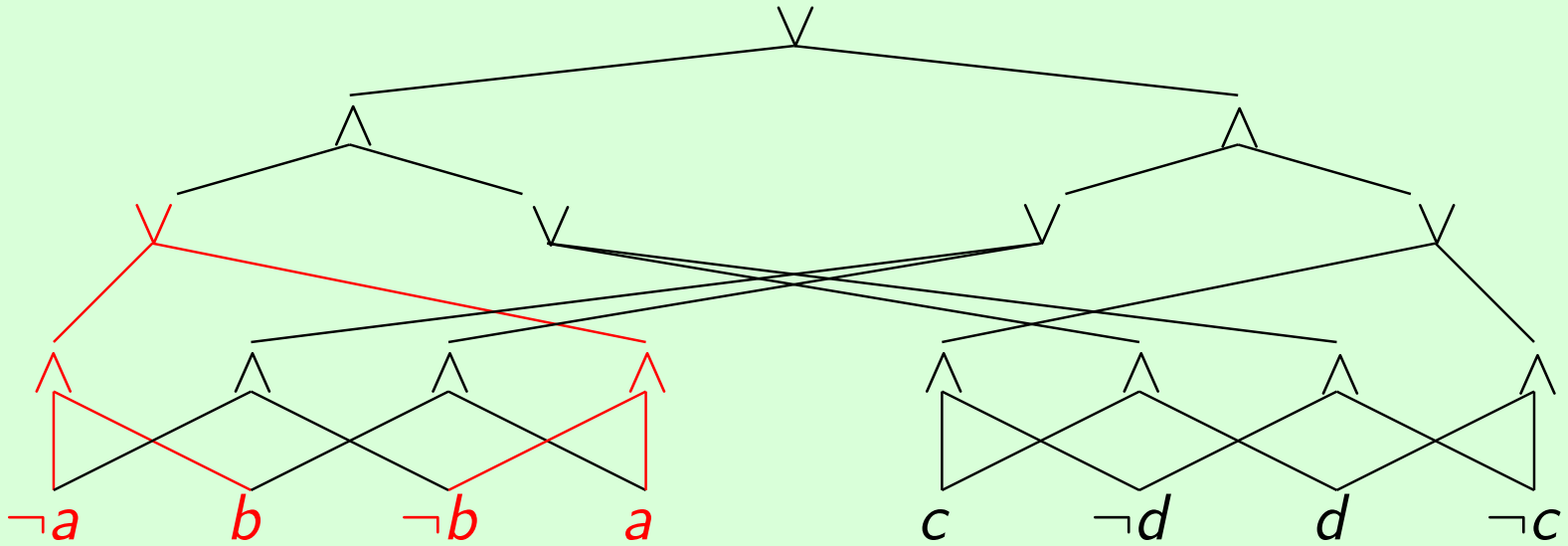


Fragments de $\mathcal{NNF}_{PS}$: exemples

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

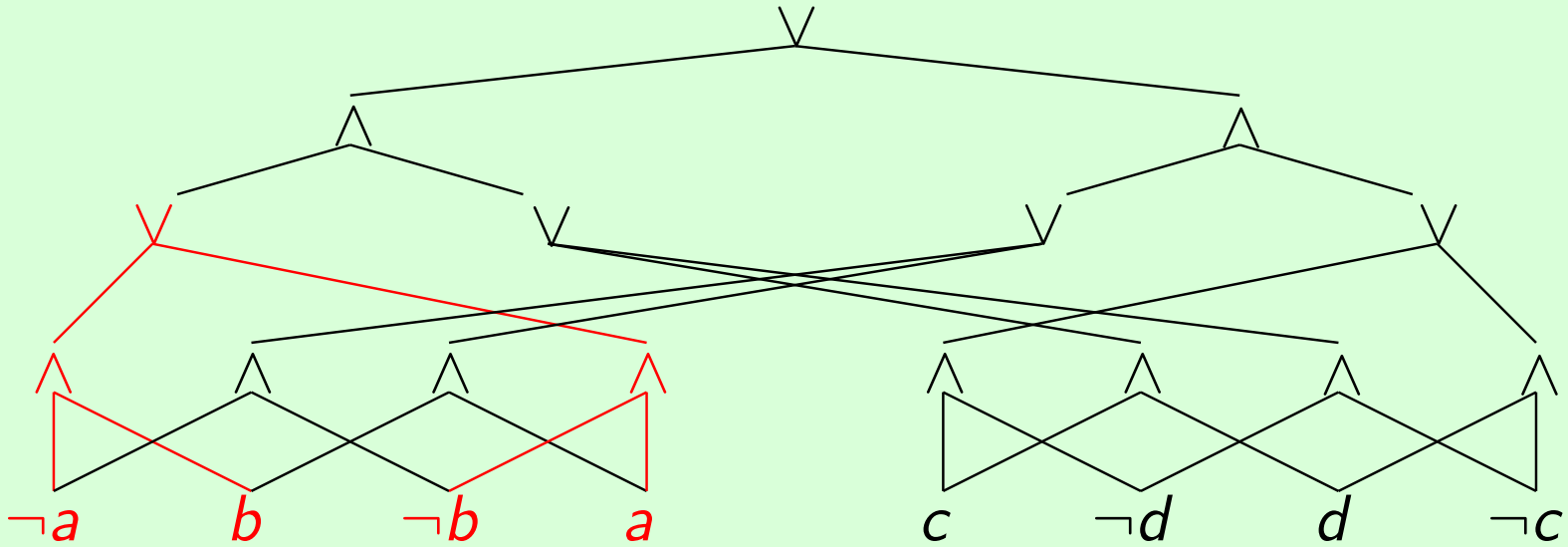


Fragments de $\mathcal{NNF}_{PS}$: exemples

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

Exemple

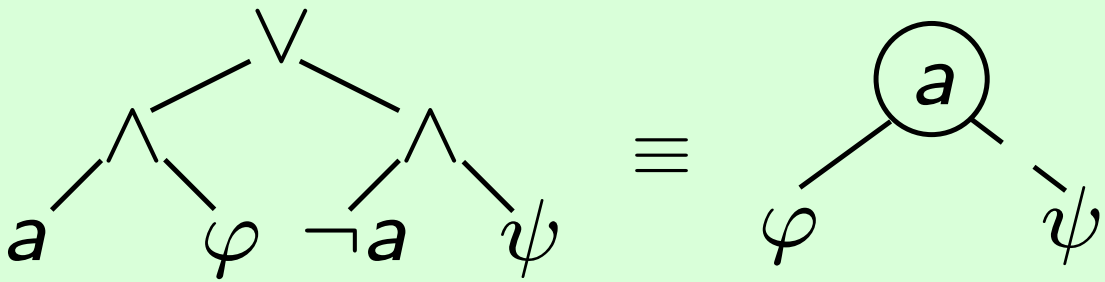
Uniformité



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

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

Impliqués premiers de Σ :

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

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 $\pi' \models \pi$

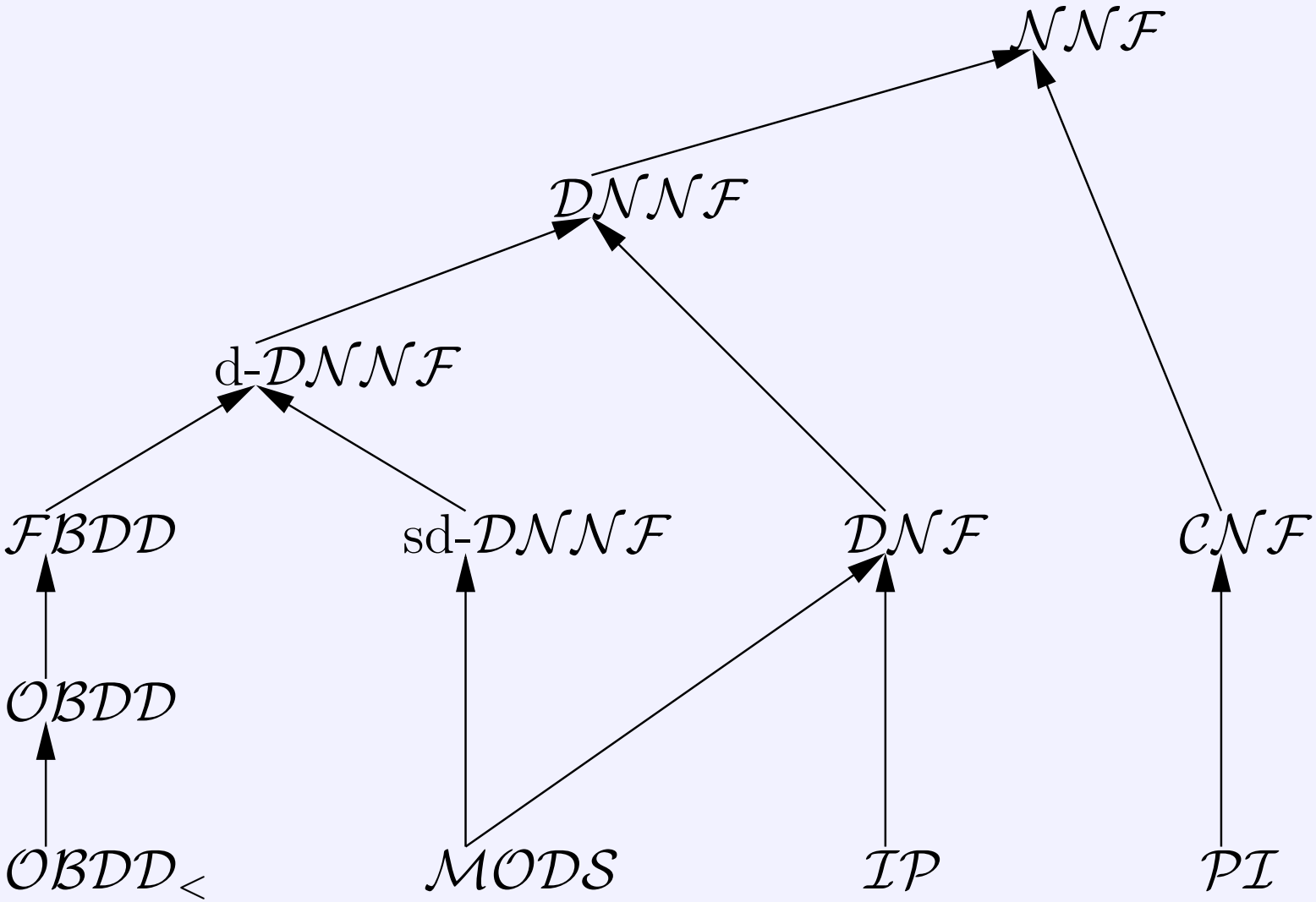
Exemple

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

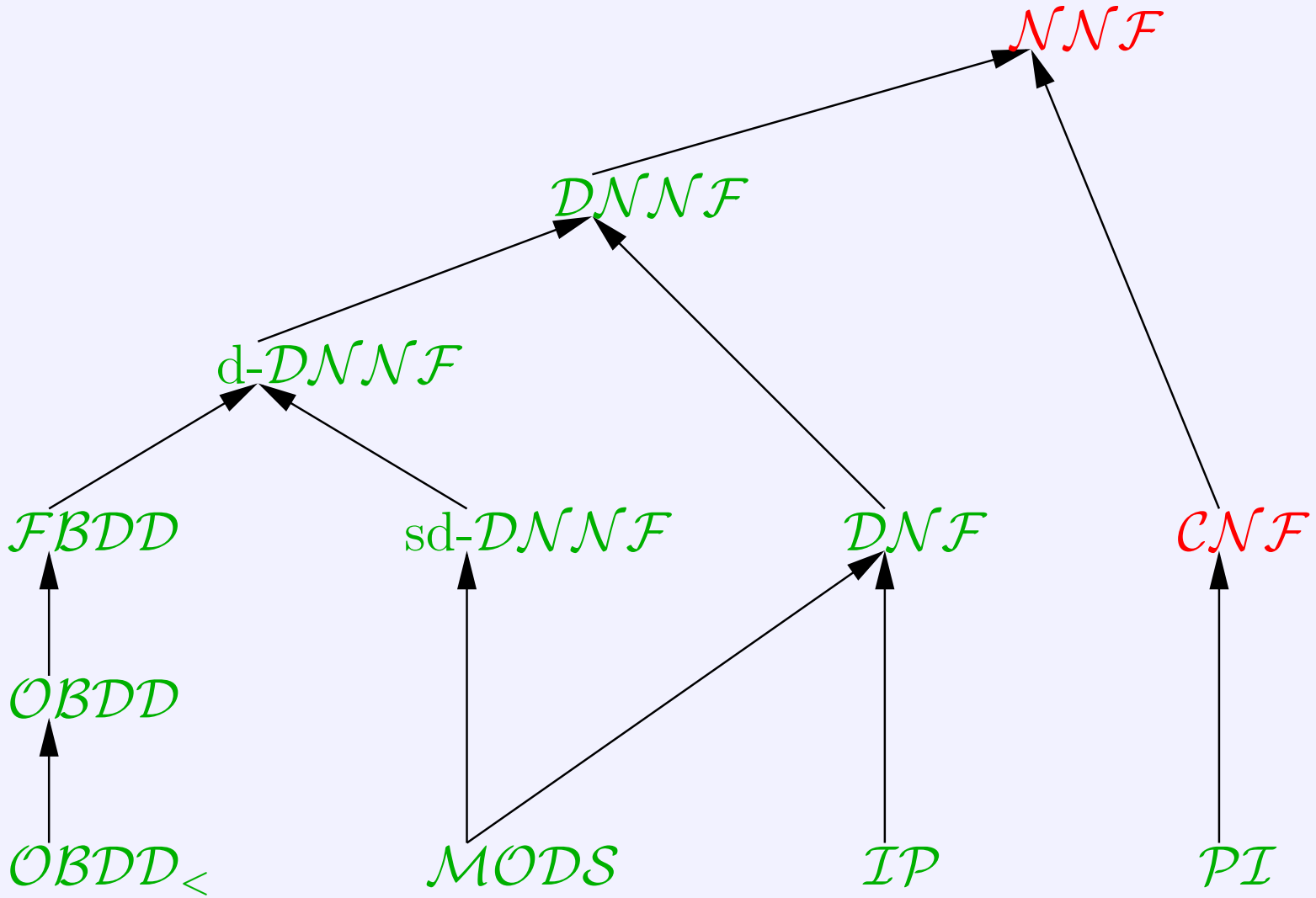
Impliquants premiers de Σ :

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

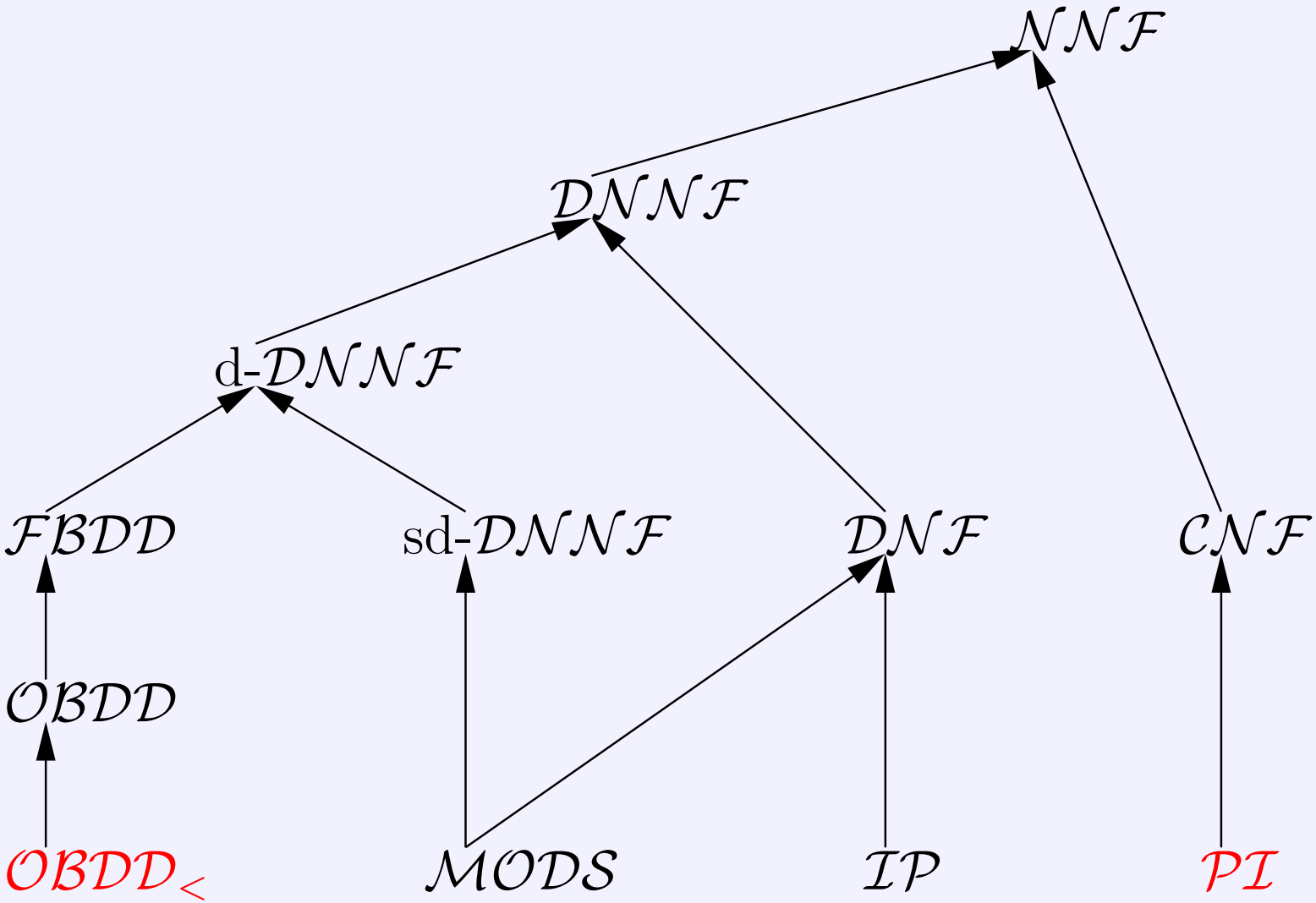
Inclusion des fragments [Darwiche & Marquis 2001]



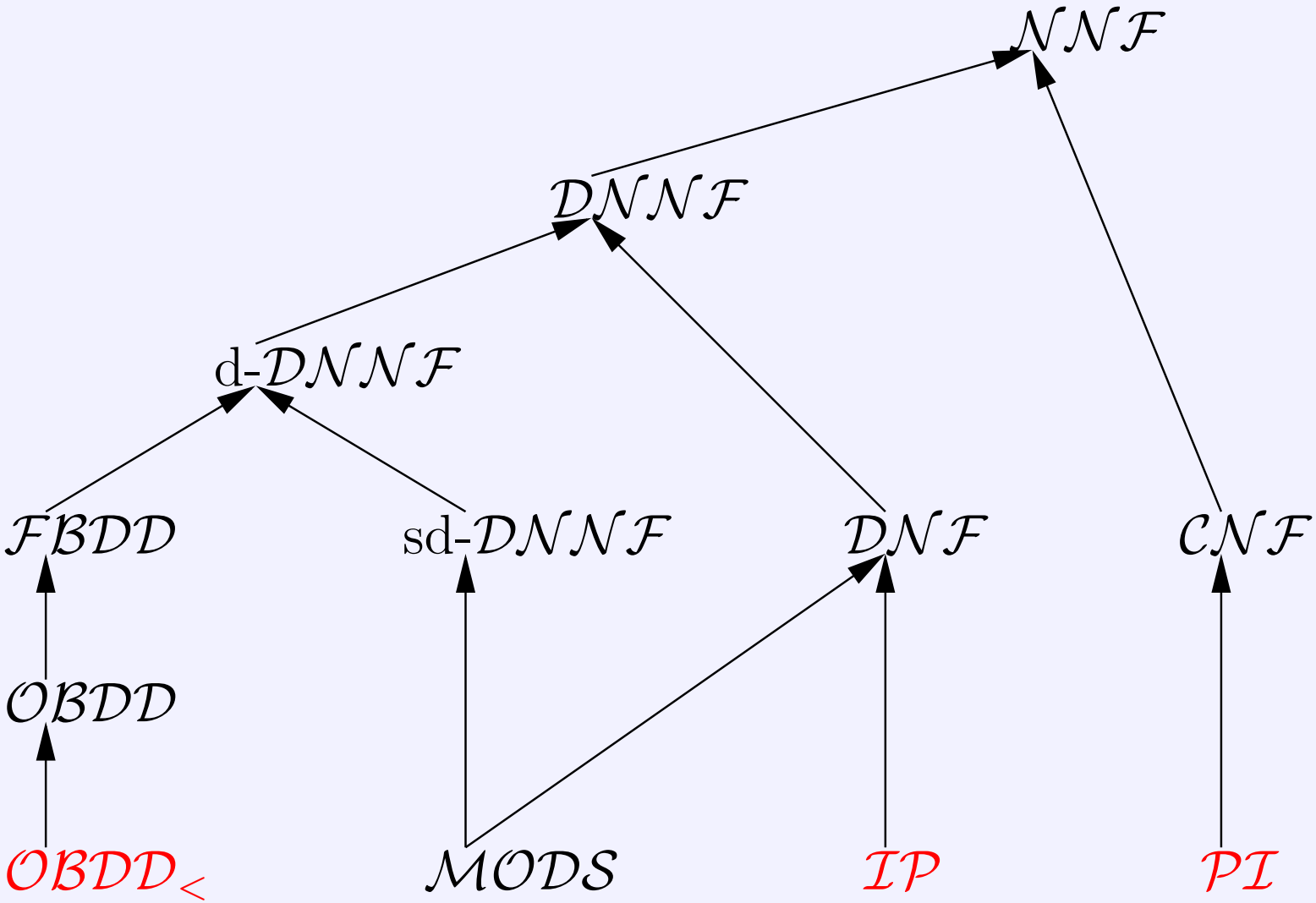
Complexité pour SAT [Darwiche & Marquis 2001]



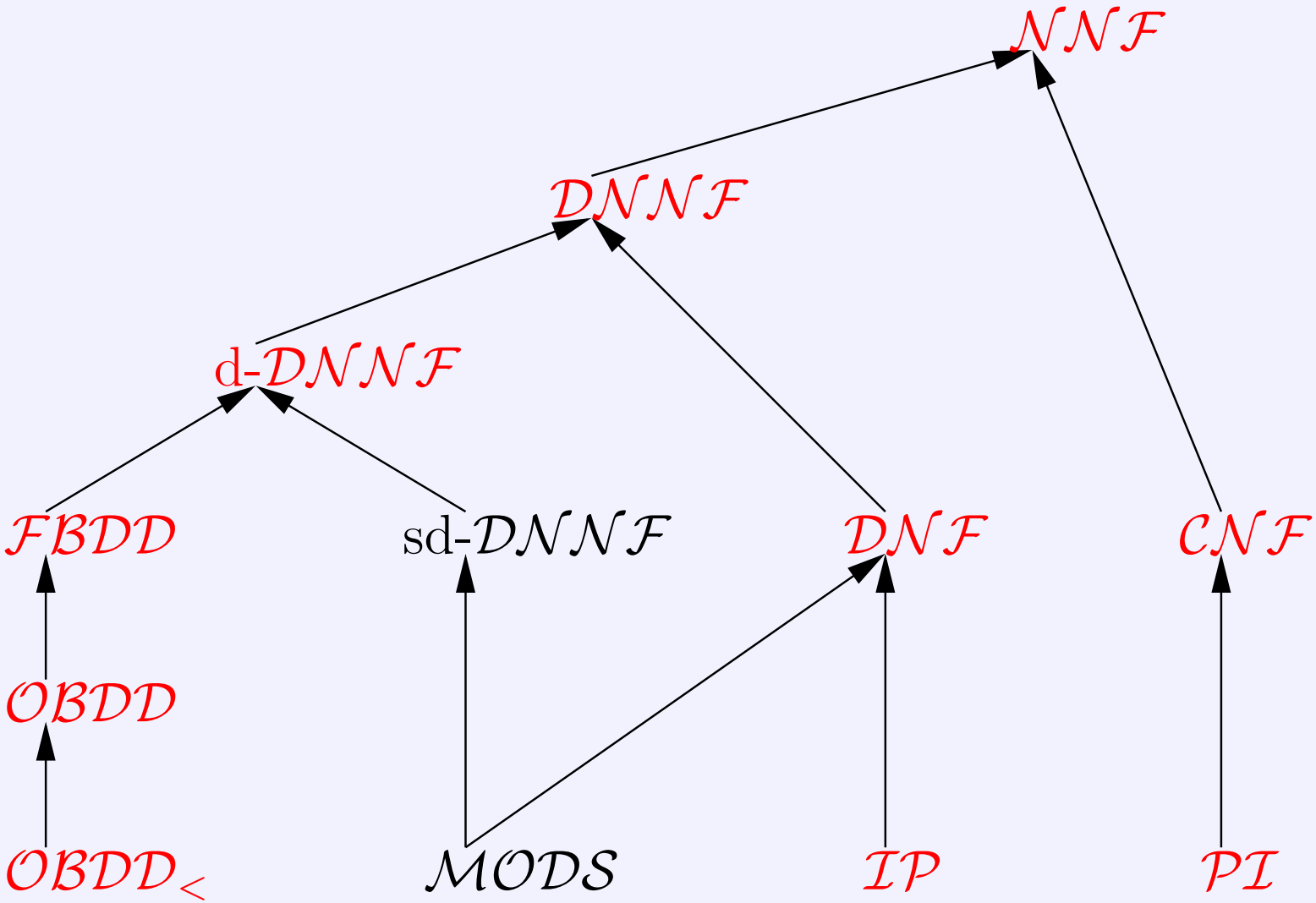
Résultats de complexité pour QBF



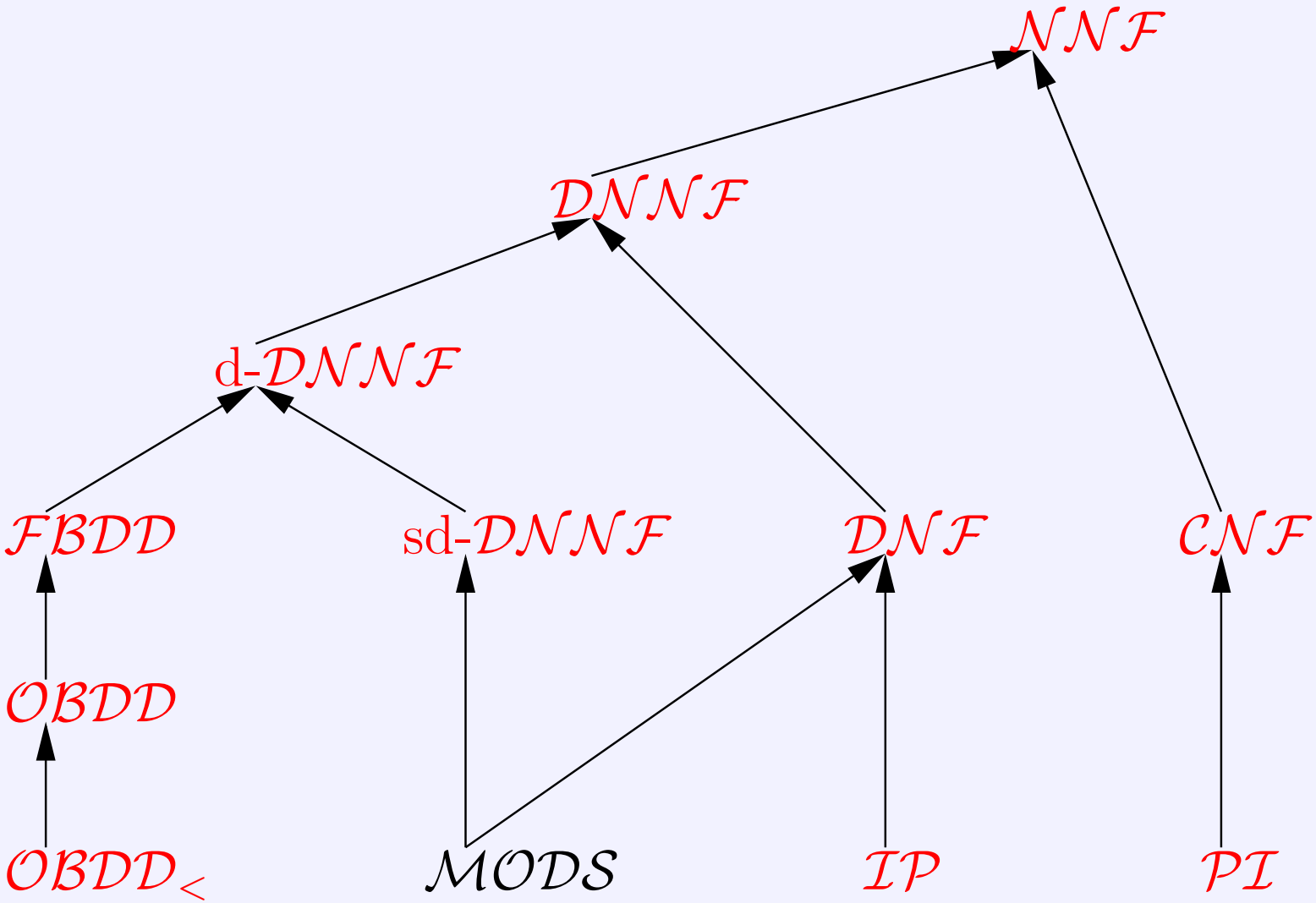
Résultats de complexité pour QBF



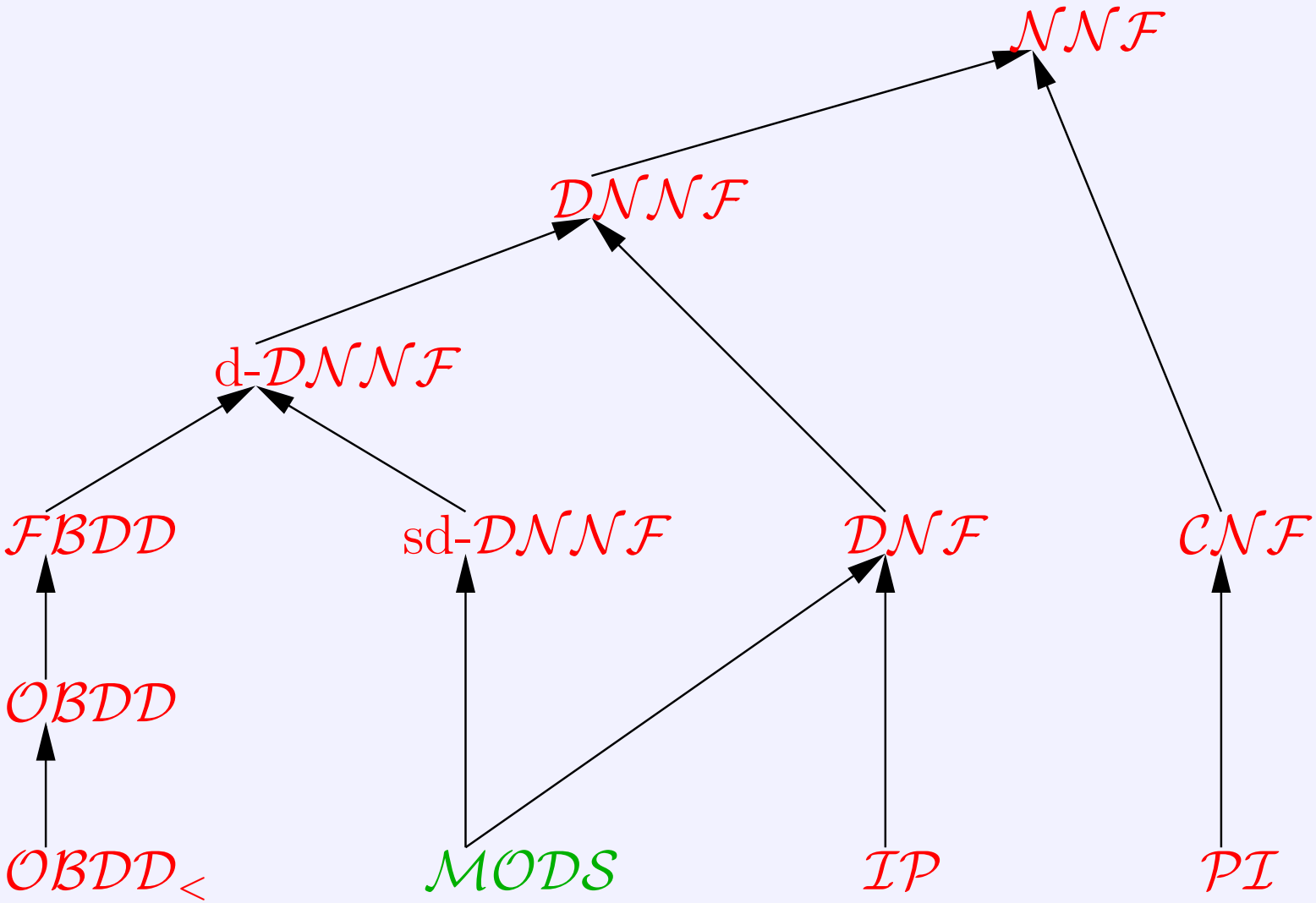
Résultats de complexité pour QBF



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

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

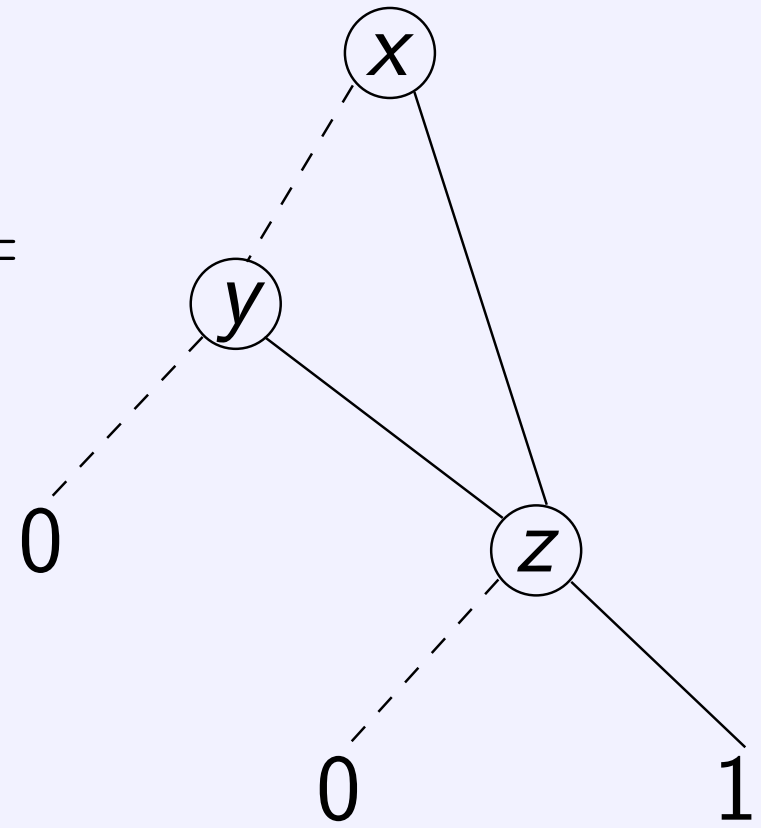
$$\phi \equiv (x \vee y) \wedge z$$

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

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

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

$\phi =$

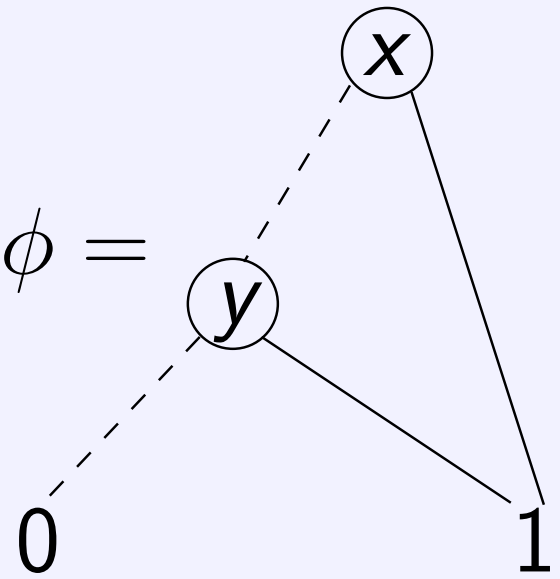


$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

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

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

$\exists z\ .\ \phi =$

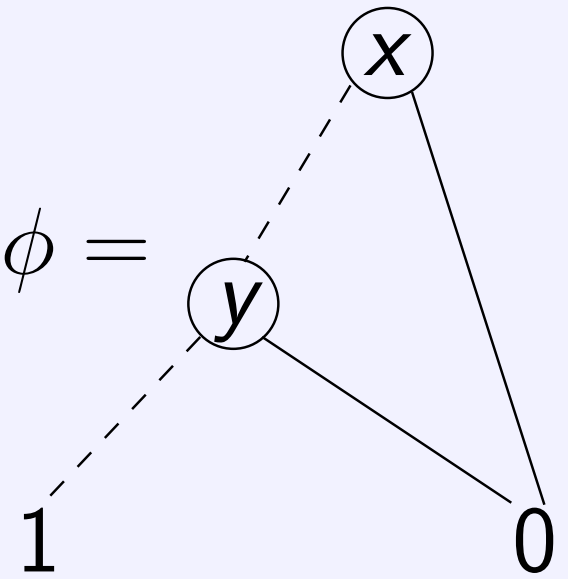


$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

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

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

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

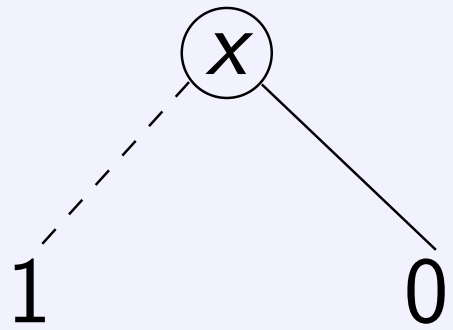


$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

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

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

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

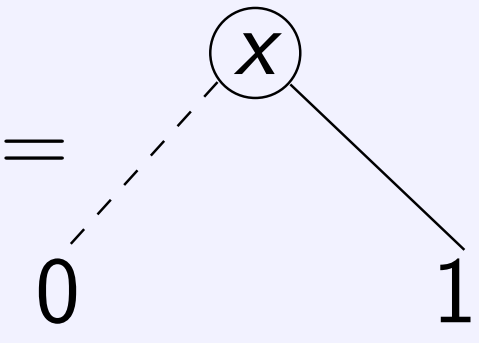


$\mathcal{OBDD}_<$ avec préfixe compatible : un cas polynomial

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

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

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



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

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

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

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

$$\Rightarrow \Sigma \text{ est valide}$$

Méthodes de résolution

- Techniques générales :
  - Q-résolution [Kleine-Büning *et al.* 1995]
  - Q-DPLL [DP 1960, DLL 1962]
- Règles de simplification et parcours de l'espace de recherche :
  - Fausseté et vérité triviale [Cadoli *et al.* 1998]
  - Propagation unitaire ou monotone [Cadoli *et al.* 1998]
  - Inversion de quantificateurs et échantillonnage [Rintanen 2001]
  - Retour arrière (systématique ou intelligent) [Letz 2001, Giunchiglia *et al.* 2003]
  - Apprentissage [Letz 2001, Giunchiglia *et al.* 2002]

Prouveurs à la DPLL [DP 1960, DLL 1962]

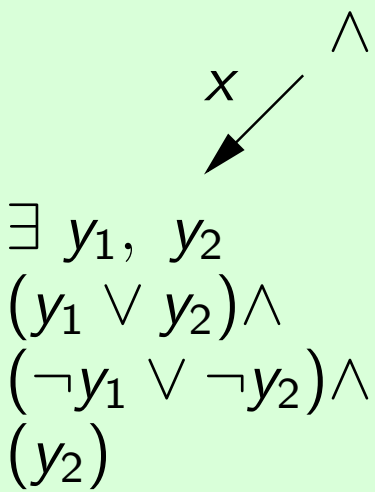
Exemple

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

Prouveurs à la DPLL [DP 1960, DLL 1962]

Exemple

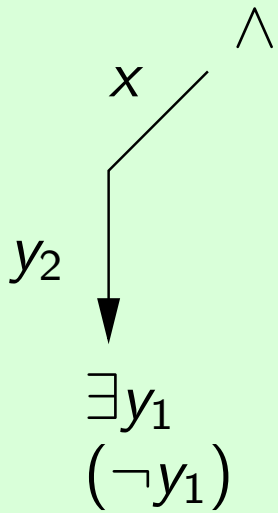
$$\forall\ x\ \exists\ y_1,\ y_2\ .\ [(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge (\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$



Prouveurs à la DPLL [DP 1960, DLL 1962]

Exemple

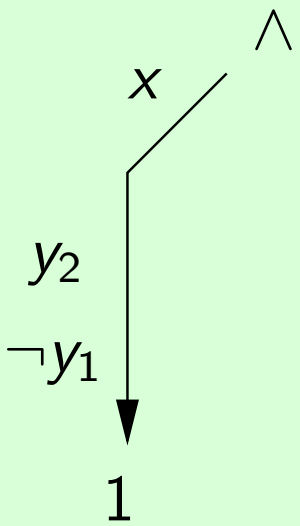
$$\forall\ x\ \exists\ y_1,\ y_2\ .\ [(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge (\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$



Prouveurs à la DPLL [DP 1960, DLL 1962]

Exemple

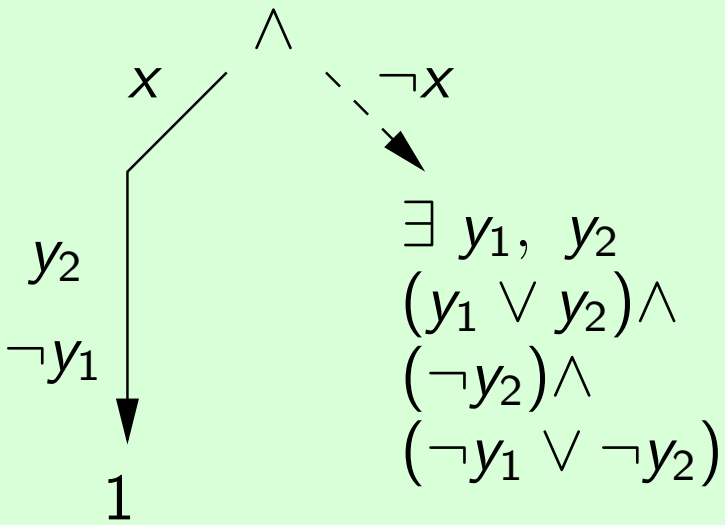
$$\forall\, x\, \exists\, y_1,\ y_2\ .\ [(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge (\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$



Prouveurs à la DPLL [DP 1960, DLL 1962]

Exemple

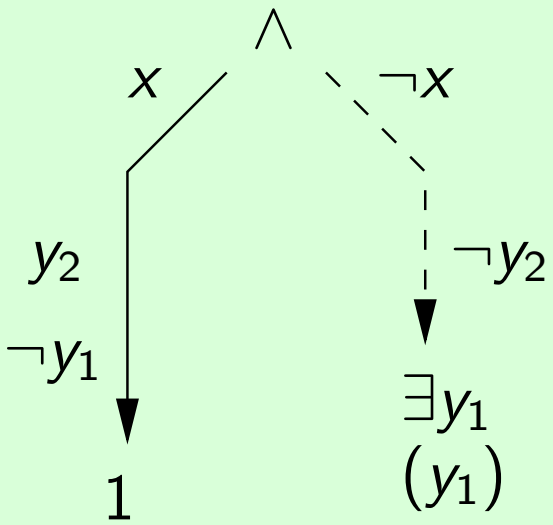
$$\forall\, x\, \exists\, y_1,\ y_2\ .\ [(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge (\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$



Prouveurs à la DPLL [DP 1960, DLL 1962]

Exemple

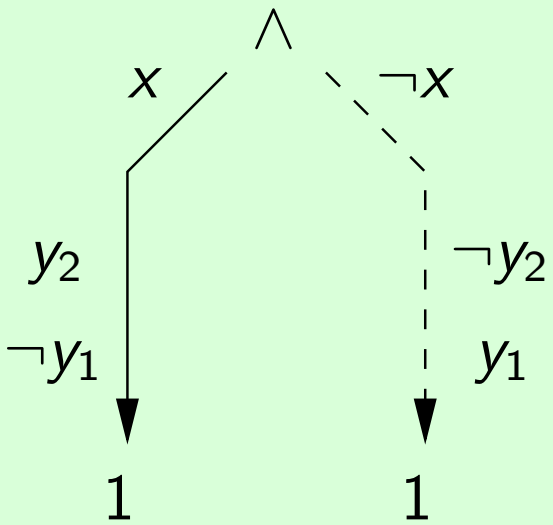
$$\forall\, x\, \exists\, y_1,\ y_2\ .\ [(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge (\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$



Prouveurs à la DPLL [DP 1960, DLL 1962]

Exemple

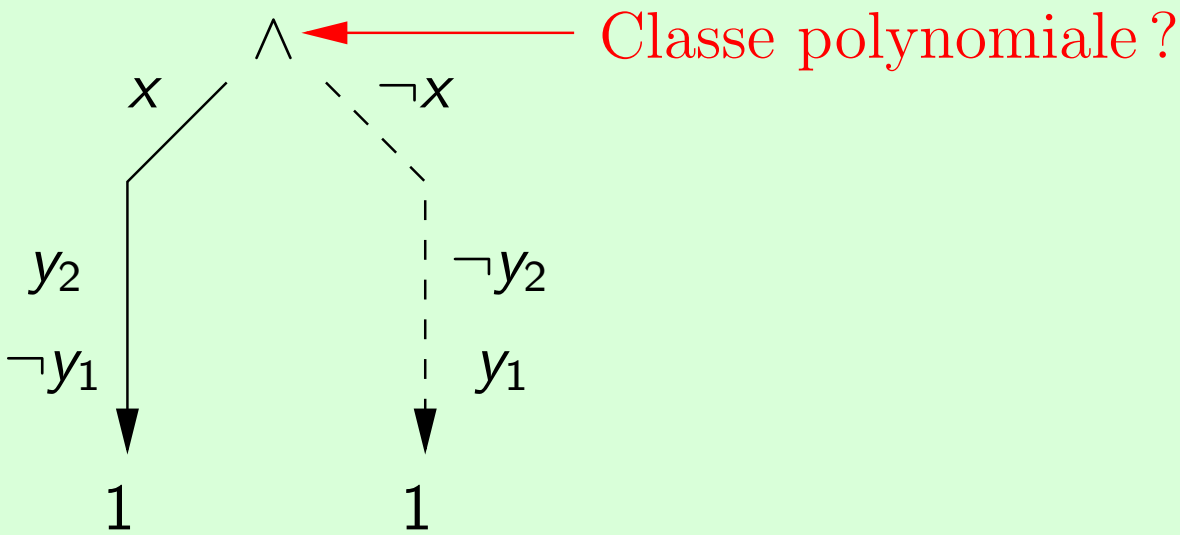
$$\forall\, x\ \exists\, y_1,\ y_2\ .\ [(y_1 \vee y_2) \wedge (\neg y_2 \vee x) \wedge (\neg y_1 \vee \neg y_2) \wedge (y_2 \vee \neg x)]$$



Prouveurs à la DPLL [DP 1960, DLL 1962]

Exemple

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

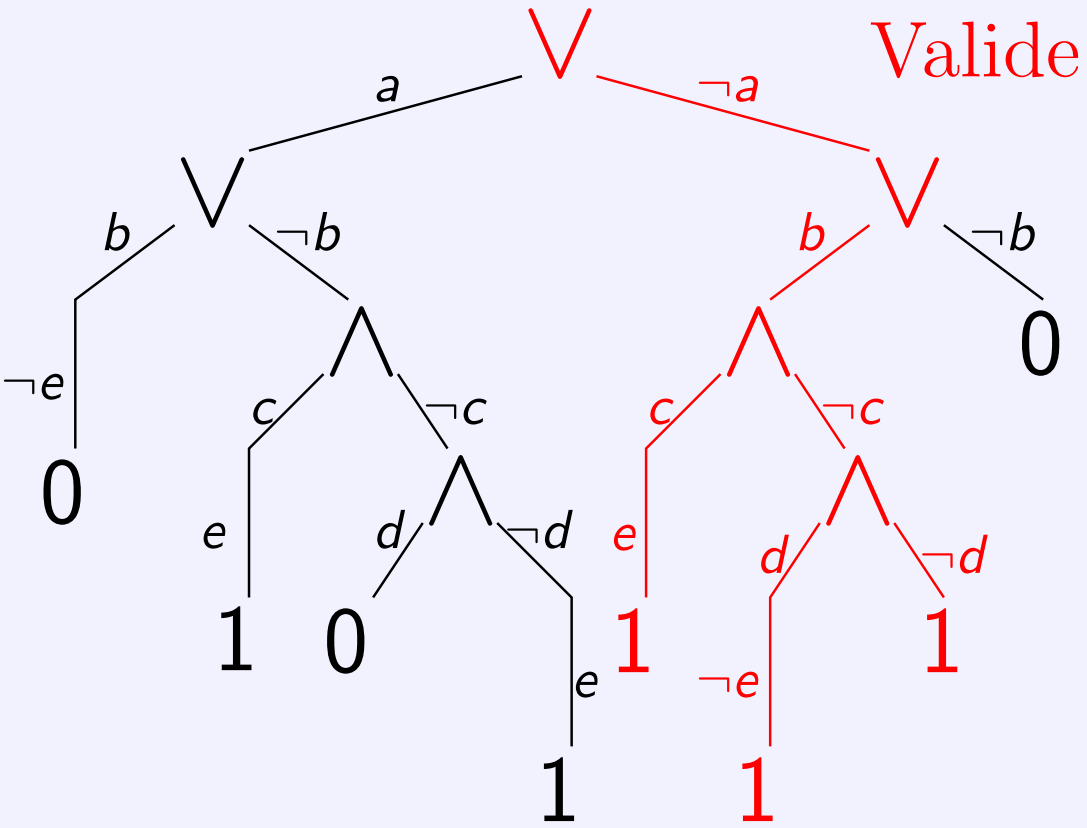


Classes polynomiales : présentation

- 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, ...

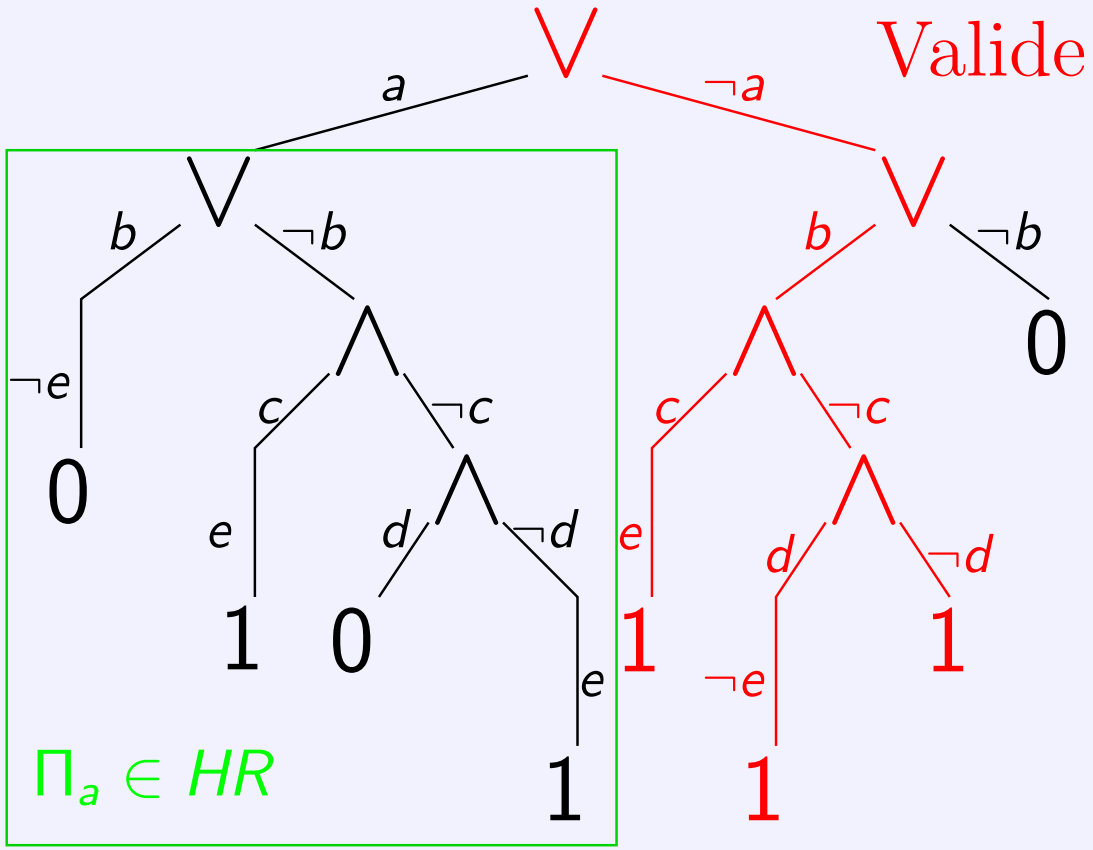
Classes polynomiales : motivation

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



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 de Horn quantifiées ($\mathcal{QHF}$)

Définition ($\mathcal{QHF}$)

- Clause de Horn : clause comprenant au plus un littéral positif
- $\mathcal{QHF}$: $\mathcal{QBF}$ 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 quantifiées (ren$\mathcal{QHF}$)

Définition (*ren$\mathcal{QHF}$*)

- 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

$$\Pi_2 = \forall\ a, b\ \exists\ c, d\ \forall\ e\ .\ \begin{bmatrix} (a \vee b \vee \neg d \vee \neg e) & \wedge \\ (\neg a \vee \neg c \vee \neg e) & \wedge \\ (b \vee c \vee d) & \wedge \\ (\neg a \vee c \vee \neg d) & \end{bmatrix}$$

$$\Pi_1 = \forall\ a, b\ \exists\ c, d\ \forall\ e\ .\ \begin{bmatrix} (a \vee \neg b \vee \neg d \vee \neg e) & \wedge \\ (\neg a \vee c \vee \neg e) & \wedge \\ (\neg b \vee \neg c \vee d) & \wedge \\ (\neg a \vee \neg c \vee \neg d) & \end{bmatrix}$$

Π_2 est Π_1 dans laquelle on a renommé b et c

Reconnaissance de formules Horn renommables

Définition *($\Rightarrow_\Sigma$ et $\textsc{Clos}_\Sigma(l)$)*

- $l \Rightarrow_\Sigma t$ ssi il existe une clause $C \in \Phi$ tq $l \in C$, $\neg t \in C$ et $l \neq \neg t$
- $\textsc{Clos}_\Sigma(l)$ désigne l'ensemble $\{t \ / \ l \Rightarrow_\Sigma^* t\}$

Si $\textsc{Clos}_\Sigma(l)$ est contradictoire (contient un littéral et son contraire) et $l \in R$, alors R n'est pas fermé

Proposition *(Proposition 1.1 de [Hébrard 1994])*

Un renommage R est Horn si et seulement s'il est fermé, i.e. pour tout $l \in R$, $\textsc{Clos}_\Sigma(l) \subseteq R$

Exemple

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

- $\textsc{Clos}_\Sigma(\neg a) = \{\neg a\}$
- $\textsc{Clos}_\Sigma(a) = \{a\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Exemple

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

- $\textsc{Clos}_{\Sigma}(\neg a) = \{\neg a\}$
- $\textsc{Clos}_{\Sigma}(a) = \{a\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Exemple

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

- $\textsc{Clos}_{\Sigma}(\neg a) = \{\neg a, c, e\}$
- $\textsc{Clos}_{\Sigma}(a) = \{a\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Exemple

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

- $\textsc{Clos}_\Sigma(\neg a) = \{a, \neg a, c, d, e\}$
- $\textsc{Clos}_\Sigma(a) = \{a\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Exemple

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

- $\textsc{Clos}_\Sigma(\neg a) = \{a, \neg a, c, d, e\} \equiv \bot$
- $\textsc{Clos}_\Sigma(a) = \{a\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Exemple

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

- $\textsc{Clos}_\Sigma(\neg a) = \{a, \neg a, c, d, e\} \equiv \bot$
- $\textsc{Clos}_\Sigma(a) = \{a\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Exemple

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

- $\textsc{Clos}_\Sigma(\neg a) = \{a, \neg a, c, d, e\} \equiv \bot$
- $\textsc{Clos}_\Sigma(a) = \{a, \neg b, d, e\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Exemple

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

- $\textsc{Clos}_{\Sigma}(\neg a) = \{a, \neg a, c, d, e\} \equiv \bot$
- $\textsc{Clos}_{\Sigma}(a) = \{a, \neg b, \neg c, d, e\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Exemple

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

- $\textsc{Clos}_\Sigma(\neg a) = \{a, \neg a, c, d, e\} \equiv \bot$
- $\textsc{Clos}_\Sigma(a) = \{a, \neg b, \neg c, d, e\}$
- Un Horn renommage possible pour Π_2 est $\{a, \neg b, \neg c, d, e\}$

Qbfl Version 1.7

- Développé depuis 2003
- `http://www.cril.univ-artois.fr/~letombe/qbfl`
- Extension de DPLL
- Utilise Limmat (1.3) comme oracle SAT (vainqueur compétition SAT 2002)
- Reconnaissance de ren$\mathcal{QHF}$s
- Diverses heuristiques dont Δ

Définitions

Définition *(Distance de contradiction de Horn renommabilité d'un littéral l)*

$$\delta_l = \begin{cases} 0 \text{ si } \nexists v | l \Rightarrow_\Sigma v \\ 1 \text{ si } l \Rightarrow_\Sigma t \text{ et } l \Rightarrow_\Sigma \neg t \\ 1 + min(\{\delta_v | l \Rightarrow_\Sigma v\}) \text{ sinon.} \end{cases}$$

PARCOURS EN LARGEUR

Définition *(Δ)*

- $\Delta_x = 1024 \times \delta_x \times \delta_{\neg x} + \delta_x + \delta_{\neg x}$;
- la variable x est choisie si $x \in X_1$ et, $\forall y \in X_1$, $(x \neq y \Rightarrow \Delta_x \leq \Delta_y)$;
- le littéral x est enfin choisi si $\delta_x > \delta_{\neg x}$, $\neg x$ sinon.

Démarche expérimentale

Sur des instances de classes traitables

- Génération de jeux d'essai
- Validation de la méthode de résolution de Qbfl
- Étude du comportement de deux prouveurs actuels
- Aperçu du comportement des prouveurs soumis à l'évaluation QBF 2005

Sur des instances générales

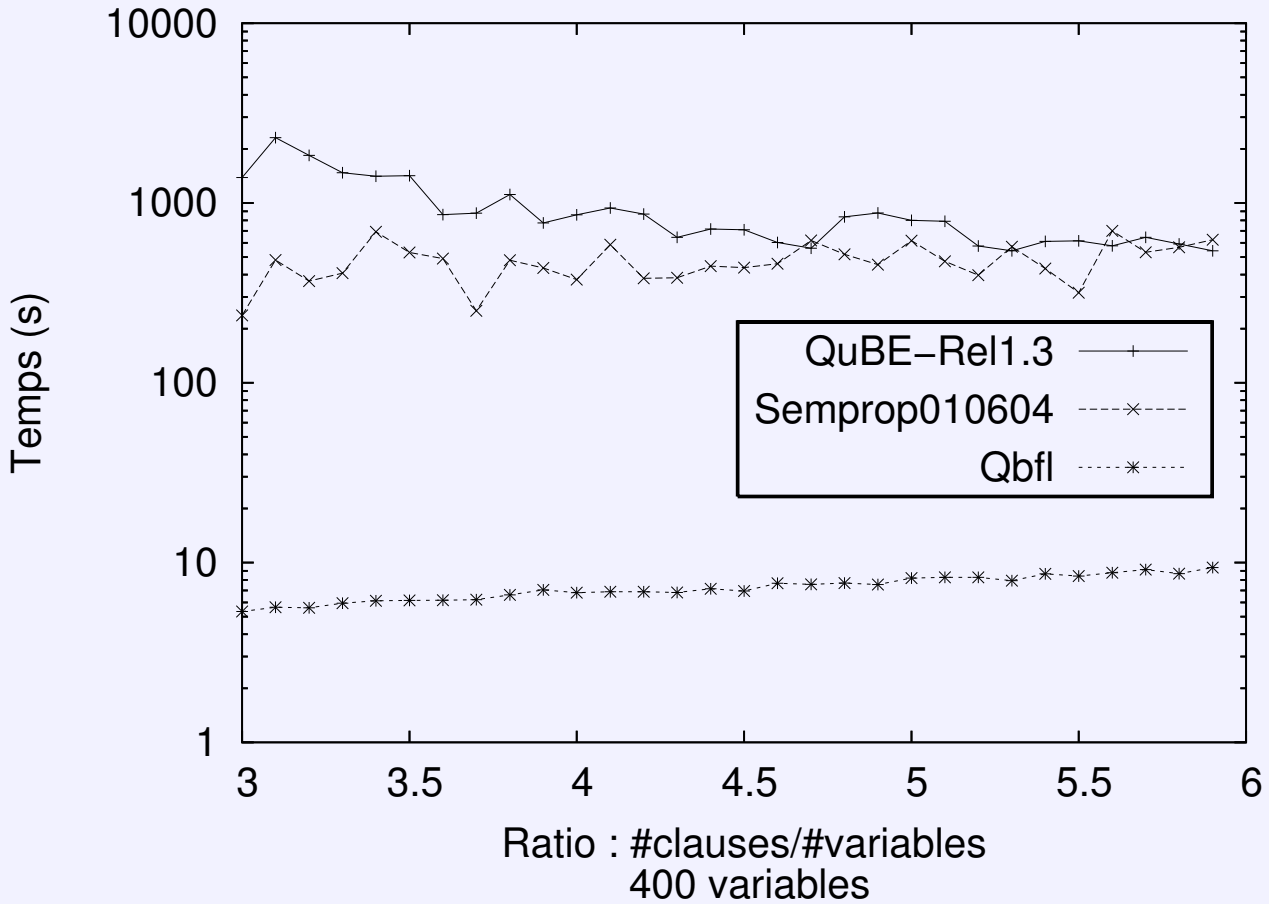
- Comportement de Qbfl muni de l'heuristique Δ sur des jeux d'essai de l'évaluation QBF 2004
- Comportement général de Qbfl

Génération aléatoire d'instances de classes polynomiales

- $\mathcal{QHF}$ et ren$\mathcal{QHF}$ générées aléatoirement
- 2 générateurs (`http://www.cril.univ-artois.fr/~letombe/qbfg`)
- $\mathcal{QHF}$: pour chaque clause
  - taille aléatoire (entre 1 et 50)
  - choix aléatoire du littéral positif
  - choix des littéraux négatifs restants
- ren$\mathcal{QHF}$:
  - nombre aléatoire de variables à renommer
  - choix aléatoire des variables à renommer
  - mêmes autres paramètres que $\mathcal{QHF}$
- Préfixe $\forall X \exists Y$, $2 < |X| < 2/3\ \#V$

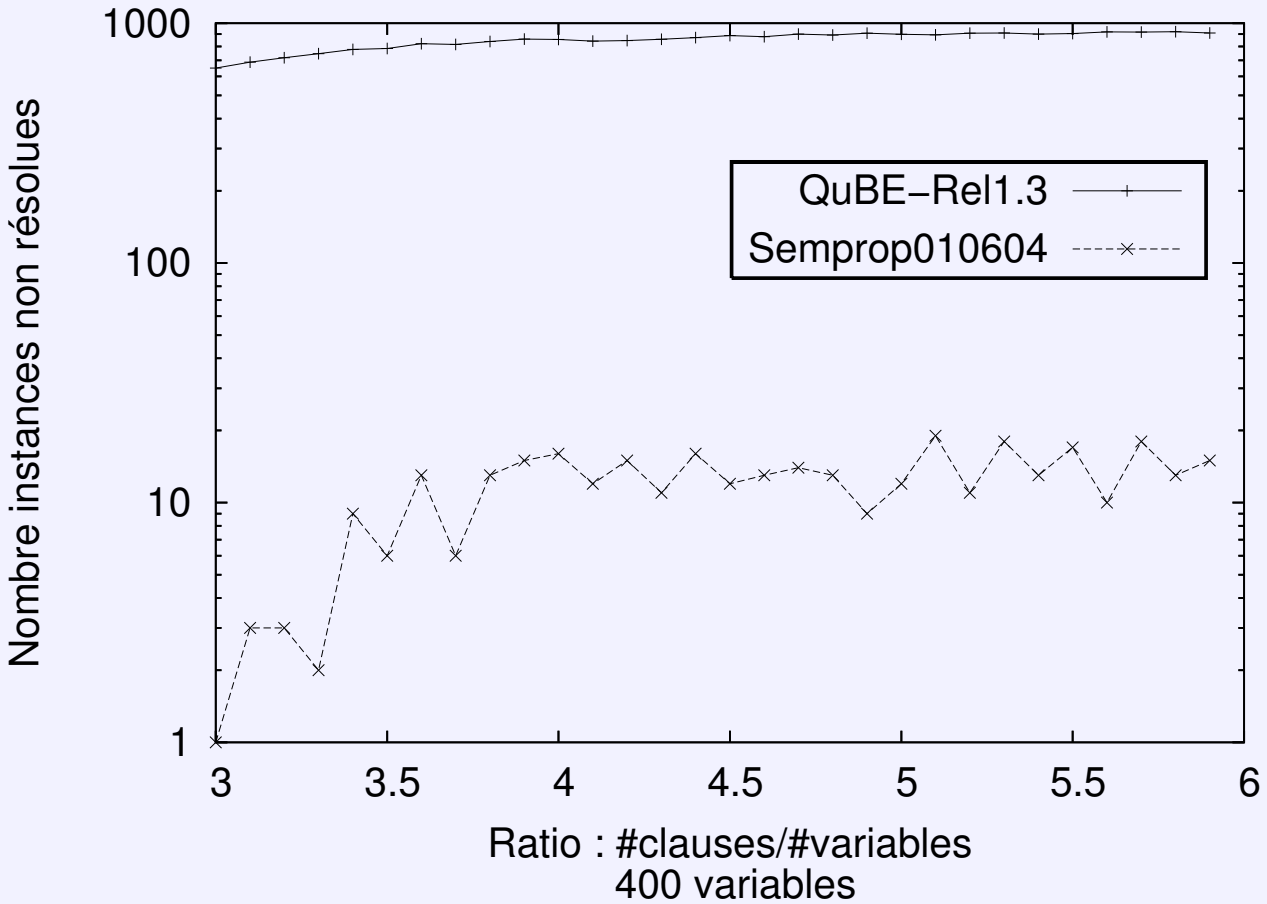
QHF

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



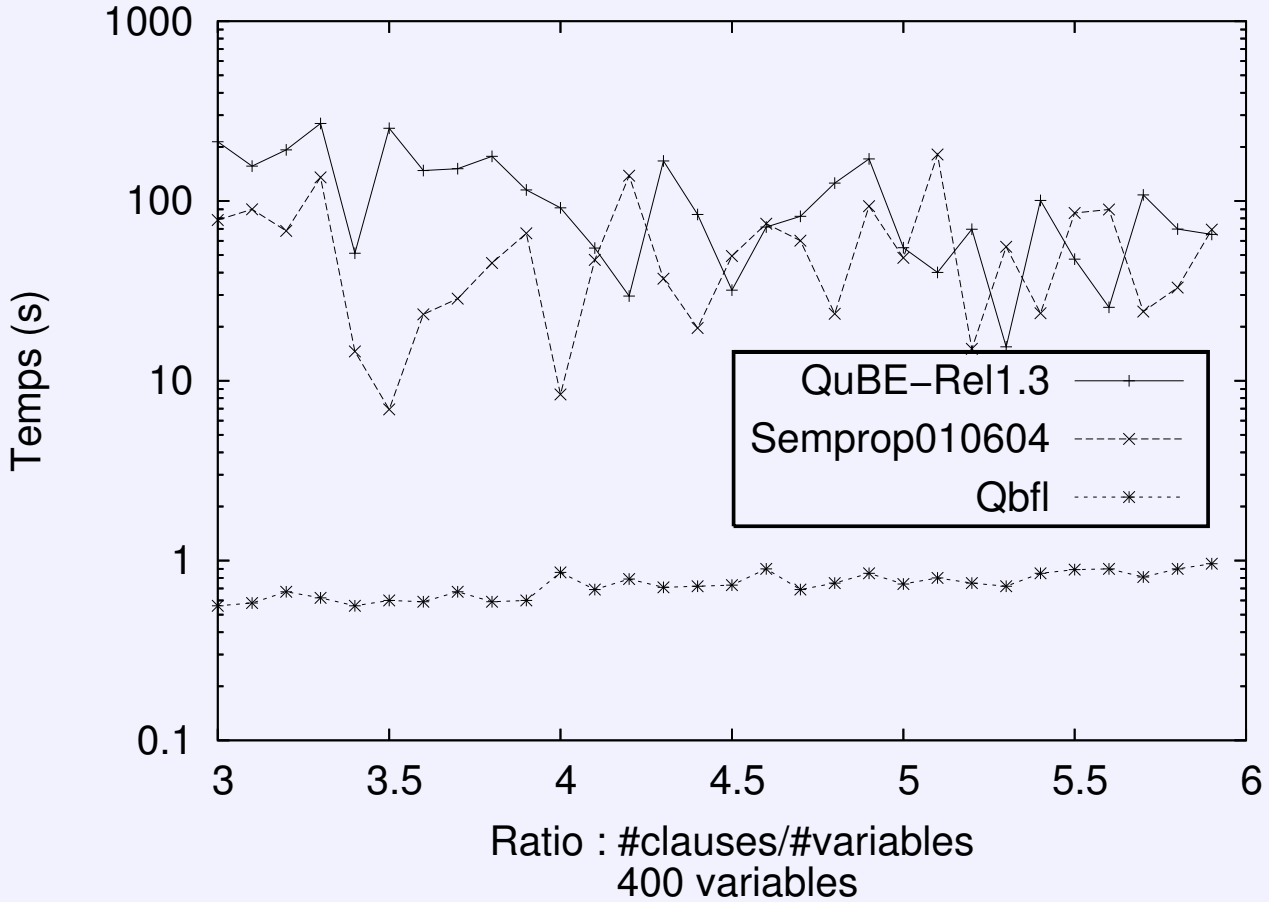
$\mathcal{QHF}$

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



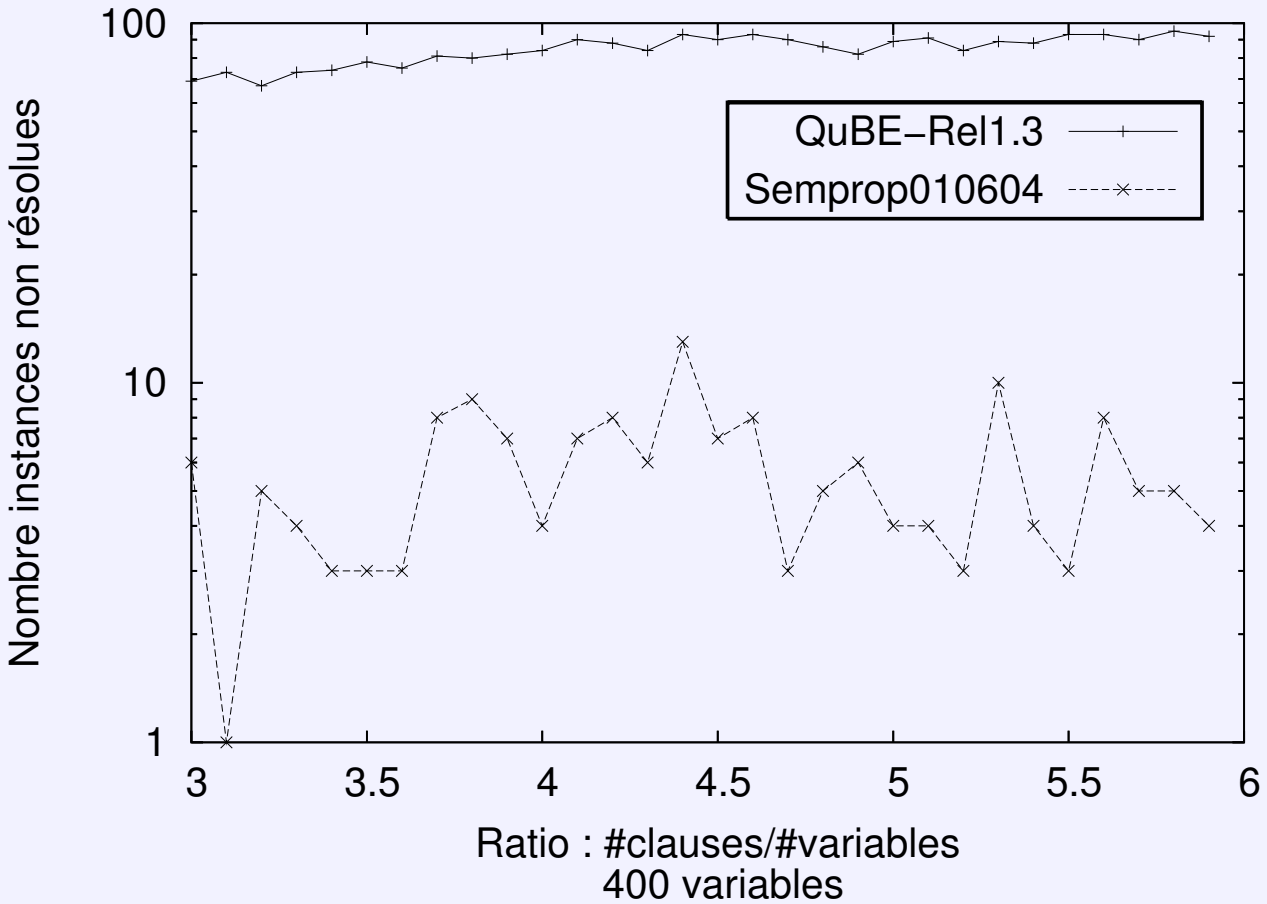
ren$\mathcal{QHF}$

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



ren$\mathcal{QHF}$

- 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 ren$\mathcal{QHF}$

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

Jeux d'essai de l'évaluation QBF 2004

- Jeux d'essai disponibles sur `http://www.qbflib.org/benchmarks.html`
- G. Pan : 9 types de 21 jeux d'essai valides et 9 faux (378 instances)
- Autres types d'instances :
  - A. Ayari (72)
  - C. Castellini (169)
  - M. Mneimneh et K. A. Sakallah (202)
  - J. Rintanen (67)
  - C. Scholl et B. Becker (64)
- Total : 952 instances
- TimeOut : 900 s

Instances de G. Pan

Type d'instance	Jeroslow-Wang				Horn renommable Δ			
	%résolus		%HR		%résolus		%HR	
	k_*_n	k_*_p	k_*_n	k_*_p	k_*_n	k_*_p	k_*_n	k_*_p
k_branch_n/p	4.76	4.76	25.26	24.56	**9.52**	4.76	9.29	9.33
k_d4_n/p	4.76	9.52	13.25	26.33	4.76	**14.28**	5.63	25.34
k_dum_n/p	4.76	4.76	11.69	11.71	**23.80**	**14.28**	11.51	11.40
k_grz_n/p	0	0	-	-	**61.90**	0	11.94	-
k_lin_n/p	9.52	9.52	20.00	11.88	9.52	**19.04**	24.26	8.65
k_path_n/p	9.52	14.28	5.47	7.43	**14.28**	**19.04**	12.12	7.64
k_ph_n/p	23.80	19.04	2.66	10.06	23.80	19.04	11.73	6.89
k_poly_n/p	9.52	4.76	0	12.91	**14.28**	**9.52**	6.66	11.04
k_t4p_n/p	4.76	0	11.37	-	4.76	**4.76**	26.62	26.02
Total v/f	7.93	7.40	11.21	14.98	**18.51**	**11.64**	13.30	13.28
Total	7.67		13.09		**15.07**		13.29	

Jeux d'essai de l'évaluation QBF 2005

Prouveur	# vrai	# faux	Total /3177
ssolve	**1128**	**913**	**2041**
semprop	1008	911	1919
sKizzo v0.5	795	903	1698
WalkQSAT	873	802	1675
QChaffLearn	828	836	1664
sKizzo v0.4	736	889	1625
GRL	843	754	1597
Qbfl	**949**	**584**	**1533**
yquaffle	742	698	1440
openqbf	786	596	1382
quantor	683	319	1002
QMRes	535	126	661
qbfbdd	229	142	371

Conclusion

- Mise en évidence de nombreux fragments intraitables (AAAI'05)
- Quelques résultats de traitabilité (sous des conditions très restrictives)

- Mise en évidence des problèmes des prouveurs QBF actuels sur des instances polynomiales
- Utilisation pratique de restrictions traitables de $\mathcal{QBF}$ dans des prouveurs
- Distance de contradiction δ (SAT'05)
- Amélioration significative de Qbfl sur certaines instances $\mathcal{QBF}$

Perspectives

Théorique :

- Découverte d'autres fragments (polynomiaux) complets ou incomplets
  - Algorithme de détection de ces fragments
  - Algorithme de résolution de ces fragments
- Analyse plus fine du préfixe

Pratique :

- Validation de l'heuristique
- Intégrer de nouvelles classes polynomiales à Qbfl
- Analyse plus complète des instances

Merci de votre attention