Complexité paramétrée (3)
Réductions paramétrées et classes de complexité

Christophe PAUL
(CNRS - LIRMM)

November 5, 2010

1. Réductions paramétriques
   - Rappels et définitions
   - Exemples
2. La W-hiérarchie.
   - Circuits logiques
   - WEIGHTED-SAT
3. Deux exemples
   - Ensemble indépendant
   - Ensemble dominant
4. L'hypothèse de complexité exponentielle (ETH)

Problème paramétré

Soit Σ un alphabet fini.

1. Une **paramétrisation** de Σ^* est une fonction $\kappa : \Sigma^* \longrightarrow \mathbb{N}$ calculable en temps polynomial.

Problème paramétré

Soit Σ un alphabet fini.

1. Une **paramétrisation** de Σ^* est une fonction $\kappa : \Sigma^* \to \mathbb{N}$ calculable en temps polynomial.
2. Un **problème paramétré** (sur Σ) est une paire (Q, κ) tel que $Q \subseteq \Sigma^*$ et κ est une paramétrisation de Σ^*.

   si $x \in \Sigma^*$ est une instance de Q, $\kappa(x)$ est son paramètre.

Classe de complexité FPT

Un problème (Q, κ) est **FPT** (**Fixed Parameterized Tractable**) s'il existe un algorithme $\mathcal{A}$ qui décide Q et dont la complexité est

$$f(\kappa(x)).n^{O(1)}$$

Définition : Réduction paramétrique

Soient (Q, κ) et (Q', κ') deux problèmes paramétrés sur les alphabets respectifs Σ et Σ'.

Une réduction paramétrique (**FPT**) de (Q, κ) vers (Q', κ') est une fonction : $$R : \Sigma^* \rightarrow (\Sigma')^*$$
telle que

1. Pour tout $x \in \Sigma^*$, nous avons $(x \in Q \Leftrightarrow R(x) \in Q')$

On notera $(Q, \kappa) \leqslant_{fpt} (Q', \kappa')$

Définition : Réduction paramétrique

Soient (Q, κ) et (Q', κ') deux problèmes paramétrés sur les alphabets respectifs Σ et Σ'.

Une réduction paramétrique (**FPT**) de (Q, κ) vers (Q', κ') est une fonction : $$R : \Sigma^* \rightarrow (\Sigma')^*$$
telle que

1. Pour tout $x \in \Sigma^*$, nous avons $(x \in Q \Leftrightarrow R(x) \in Q')$
2. R est calculable par un algorithme **FPT**(par rapport à κ) de complexité $f(\kappa(x)).|x|^{O(1)}$

On notera $(Q, \kappa) \leqslant_{fpt} (Q', \kappa')$

Définition : Réduction paramétrique

Soient (Q, κ) et (Q', κ') deux problèmes paramétrés sur les alphabets respectifs Σ et Σ'.

Une réduction paramétrique (**FPT**) de (Q, κ) vers (Q', κ') est une fonction : $$R : \Sigma^* \to (\Sigma')^*$$
telle que

1. Pour tout $x \in \Sigma^*$, nous avons $(x \in Q \Leftrightarrow R(x) \in Q')$
2. R est calculable par un algorithme **FPT**(par rapport à κ) de complexité $f(\kappa(x)).|x|^{O(1)}$
3. Il existe une fonction calculable $g : \mathbb{N} \to \mathbb{N}$ telle que
$$\text{pour tout } x \in \Sigma^*,\ \kappa'(R(x)) \leqslant g(\kappa(x))$$

On notera $(Q, \kappa) \leqslant_{fpt} (Q', \kappa')$

Réduction paramétrée

Si (Q, κ) et (Q', κ') deux problèmes paramétrés tels que:

1. $(Q, \kappa) \leqslant_{fpt} (Q', \kappa')$
2. $(Q', \kappa') \in FPT$

$$\Rightarrow (Q, \kappa) \text{ est un problème FPT.}$$

Réduction paramétrée

Si (Q, κ) et (Q', κ') deux problèmes paramétrés tels que:

1. $(Q, \kappa) \leqslant_{fpt} (Q', \kappa')$
2. $(Q', \kappa') \in FPT$

$$\Rightarrow (Q, \kappa) \text{ est un problème FPT.}$$

Réciproquement

Le problème $(\textsc{Coloration}, k)$ n'est pas **FPT** à moins que $\mathbf{P} = \mathbf{NP}$.

$\Rightarrow$

Réduction paramétrée

Si (Q, κ) et (Q', κ') deux problèmes paramétrés tels que:

1. $(Q, \kappa) \leqslant_{fpt} (Q', \kappa')$
2. $(Q', \kappa') \in FPT$

$\Rightarrow (Q, \kappa)$ est un problème FPT.

Réciproquement

Le problème $(\textsc{Coloration}, k)$ n'est pas **FPT** à moins que **P** $=$ **NP**.

$\Rightarrow$ tout problème (Q, κ) tel que $(\textsc{Coloration}, k) \leqslant_{fpt} (Q, \kappa)$, n'est pas FPT

- par exemple $(\textsc{Clique}, k)$

Multicolored Clique

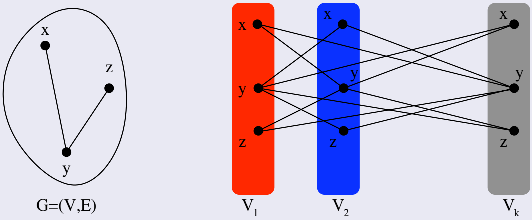
- *Données :* Un graphe $G = (V, E)$ et une coloration propre.
- *Paramètre :* Le nombre de couleurs k.
- *Question :* Existe-t'il une clique K transversale aux couleurs ? (K intersecte chacune des couleurs)

Multicolored Clique

- *Données :* Un graphe $G = (V, E)$ et une coloration propre.
- *Paramètre :* Le nombre de couleurs k.
- *Question :* Existe-t'il une clique K transversale aux couleurs ? (K intersecte chacune des couleurs)

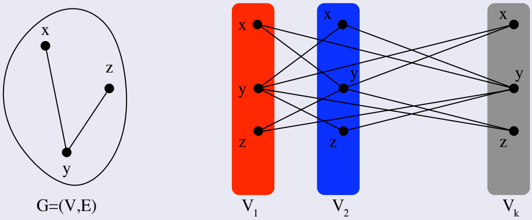
Lemme : $(\textsc{clique}, k) \leqslant_{fpt} (\textsc{Multicolored Clique}, k')$

Lemme : (CLIQUE, k) $\leqslant_{fpt}$ (MULTICOLORED CLIQUE, k')



- G possède une clique de taille k ssi H possède une clique multi-colorée.

Lemme : (CLIQUE, k) $\leqslant_{fpt}$ (MULTICOLORED CLIQUE, k')



- G possède une clique de taille k ssi H possède une clique multi-colorée.
- Transformation en temps polynomial et les paramètres sont identiques.

Set Packing

- *Données :* Une collection $\mathcal{C}$ de sous-ensembles d'un ensemble fini S.
- *Paramètre :* Un entier k.
- *Question :* Existe-t'il une sous-collection $\mathcal{C}' \subseteq \mathcal{C}$ forme d'au moins k sous-ensembles deux à deux disjoints ?

Set Packing

- *Données :* Une collection $\mathcal{C}$ de sous-ensembles d'un ensemble fini S.
- *Paramètre :* Un entier k.
- *Question :* Existe-t'il une sous-collection $\mathcal{C}' \subseteq \mathcal{C}$ forme d'au moins k sous-ensembles deux à deux disjoints ?

Montrer que Ensemble Indépendant $\leqslant_{fpt}$ Set Packing

- Soit $G = (V, E)$ instance de (Ensemble Indépendant,k).

Set Packing

- *Données :* Une collection $\mathcal{C}$ de sous-ensembles d'un ensemble fini S.
- *Paramètre :* Un entier k.
- *Question :* Existe-t'il une sous-collection $\mathcal{C}' \subseteq \mathcal{C}$ forme d'au moins k sous-ensembles deux à deux disjoints ?

Montrer que Ensemble Indépendant $\leqslant_{fpt}$ Set Packing

- Soit $G = (V, E)$ instance de (Ensemble Indépendant,k).

$$\forall v \in V : S_v = \{\{v, u\} \mid u \in N(v)\} \cup \{\{v, v\}\}$$

Set Packing

- *Données :* Une collection $\mathcal{C}$ de sous-ensembles d'un ensemble fini S.
- *Paramètre :* Un entier k.
- *Question :* Existe-t'il une sous-collection $\mathcal{C}' \subseteq \mathcal{C}$ forme d'au moins k sous-ensembles deux à deux disjoints ?

Montrer que Ensemble Indépendant $\leqslant_{fpt}$ Set Packing

- Soit $G = (V, E)$ instance de (Ensemble Indépendant,k).

$$\forall v \in V : S_v = \{\{v, u\} \mid u \in N(v)\} \cup \{\{v, v\}\}$$

$$R(G = (V, E)) = (\mathcal{C} = \{S_v \mid v \in V\}, S = \bigcup_{v \in V} S_v)$$

Set Packing

- *Données :* Une collection $\mathcal{C}$ de sous-ensembles d'un ensemble fini S.
- *Paramètre :* Un entier k.
- *Question :* Existe-t'il une sous-collection $\mathcal{C}' \subseteq \mathcal{C}$ forme d'au moins k sous-ensembles deux à deux disjoints ?

Montrer que Ensemble Indépendant $\leqslant_{fpt}$ Set Packing

- Soit $G = (V, E)$ instance de (Ensemble Indépendant,k).

$$\forall v \in V : S_v = \{\{v, u\} \mid u \in N(v)\} \cup \{\{v, v\}\}$$

$$R(G = (V, E)) = (\mathcal{C} = \{S_v \mid v \in V\}, S = \bigcup_{v \in V} S_v)$$

- $G \in$ (Ens. Indép., k) $\Leftrightarrow R(G) \in$ (Set Packing, k)

SET PACKING

- *Données :* Une collection $\mathcal{C}$ de sous-ensembles d'un ensemble fini S.
- *Paramètre :* Un entier k.
- *Question :* Existe-t'il une sous-collection $\mathcal{C}' \subseteq \mathcal{C}$ forme d'au moins k sous-ensembles deux à deux disjoints ?

Montrer que ENSEMBLE INDÉPENDANT $\leqslant_{fpt}$ SET PACKING

- Soit $G = (V, E)$ instance de (ENSEMBLE INDÉPENDANT,k).
$$\forall v \in V : S_v = \{\{v, u\} \mid u \in N(v)\} \cup \{\{v, v\}\}$$
$$R(G = (V, E)) = (\mathcal{C} = \{S_v \mid v \in V\}, S = \bigcup_{v \in V} S_v)$$
- $G \in$ (ENS. INDÉP., k) $\Leftrightarrow R(G) \in$ (SET PACKING, k)
- La réduction se fait en temps **FPT** (polynomiale en fait) et le nouveau paramètre est le même que l'ancien.

Exercices 1

1. Montrer que la relation $\leqslant_{fpt}$ est réflexive et transitive.
2. Montrer que $(\textsc{Multicolored Clique}, k') \leqslant_{fpt} (\textsc{clique}, k)$
3. Pourquoi la réduction classique de ENSEMBLE INDÉPENDANT vers VERTEX COVER ne permet pas de montrer que ENSEMBLE INDÉPENDANT est **FPT**?

Exercices 1

1. Montrer que la relation $\leqslant_{fpt}$ est réflexive et transitive.
2. Montrer que (MULTICOLORED CLIQUE, k') $\leqslant_{fpt}$ (CLIQUE, k)
3. Pourquoi la réduction classique de ENSEMBLE INDÉPENDANT vers VERTEX COVER ne permet pas de montrer que ENSEMBLE INDÉPENDANT est **FPT**?

Notations

On notera $(Q, \kappa) \sim_{fpt} (Q', \kappa')$ si

$$(Q, \kappa) \leqslant_{fpt} (Q', \kappa') \text{ et } (Q', \kappa') \leqslant_{fpt} (Q, \kappa)$$

Questions

1. Les problèmes paramétrés difficiles sont-ils tous équivalents ?

Questions

1. Les problèmes paramétrés difficiles sont-ils tous équivalents ?
2. Sous quelles hypothèses peut-on affirmer qu'un problème paramétré est difficile ?

Questions

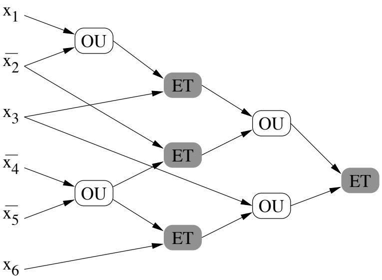
1. Les problèmes paramétrés difficiles sont-ils tous équivalents ?
2. Sous quelles hypothèses peut-on affirmer qu'un problème paramétré est difficile ?
3. Quels sont les problèmes difficiles de bases (tels que SAT pour la complexité classique) ?
4. ...

1. Réductions paramétriques
   - Rappels et définitions
   - Exemples
2. La W-hiérarchie.
   - Circuits logiques
   - WEIGHTED-SAT
3. Deux exemples
   - Ensemble indépendant
   - Ensemble dominant
4. L'hypothèse de complexité exponentielle (ETH)

Circuits logiques

Un *circuit* C de n variables est un DAG tel que:

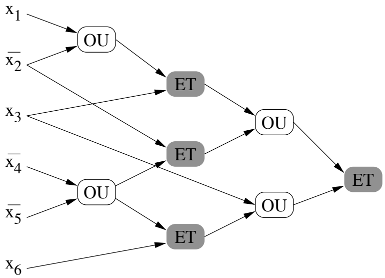
- les sommets de degré entrant 0 (*input gate*) étiquetés par un *littéral* (x_i ou $\bar{x}_i$, $i \in [n]$)
- un sommet de degré sortant 0 (*output gate*)
- les autres sommets sont étiquetés par des OU ou ET



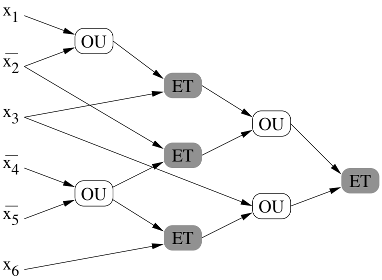
Circuits logiques

Un *circuit* C de n variables est un DAG tel que:

- les sommets de degré entrant 0 (*input gate*) étiquetés par un *littéral* (x_i ou $\bar{x}_i$, $i \in [n]$)
- un sommet de degré sortant 0 (*output gate*)
- les autres sommets sont étiquetés par des OU ou ET

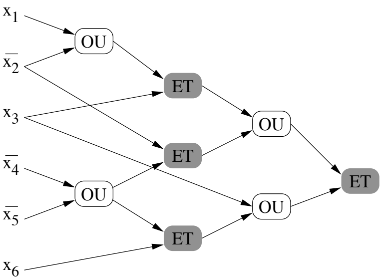


La **taille** de C est le nombre de sommets, et la **profondeur** est la longueur du plus long chemin entre une entrée et la sortie.



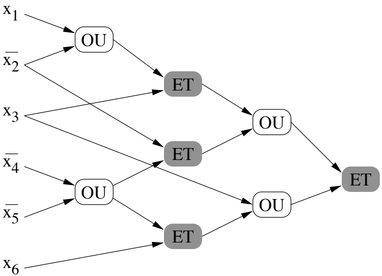
Définitions (suite)

- Un Π_t-**circuit** est un circuit de profondeur t dont la porte de sortie est un ET



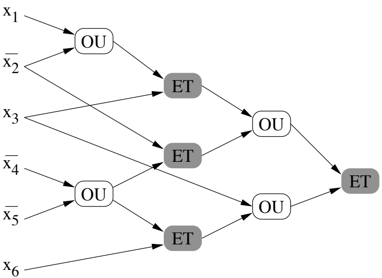
Définitions (suite)

- Un Π_t-**circuit** est un circuit de profondeur t dont la porte de sortie est un ET
- Un circuit est **monotone** (**anti-monotone**) s'il ne contient que des litéraux positifs (négatifs)



Définitions (suite)

- Un Π_t-**circuit** est un circuit de profondeur t dont la porte de sortie est un ET
- Un circuit est **monotone** (**anti-monotone**) s'il ne contient que des litéraux positifs (négatifs)
- Une affectation τ **satisfait** C si le résultat est VRAI



Définitions (suite)

- Un Π_t**-circuit** est un circuit de profondeur t dont la porte de sortie est un ET
- Un circuit est **monotone** (**anti-monotone**) s'il ne contient que des litéraux positifs (négatifs)
- Une affectation τ **satisfait** C si le résultat est VRAI
- Le **poids** d'une affectation τ est le nombre de variables à VRAI.

WEIGHTED-SAT sur les Π_t-circuits

Etant donné un circuit $C \in \Pi_t$, existe-t'il une affectation τ de poids k satisfaisant le circuit C ?

On note $(\textsc{wcs}[t], k)$

WEIGHTED-SAT sur les Π_t-circuits

Etant donné un circuit $C \in \Pi_t$, existe-t'il une affectation τ de poids k satisfaisant le circuit C ?

On note $(\textsc{wcs}[t], k)$

Le problème WEIGHTED-CNF-SAT

Etant donné une formule booléenne CNF (ou 3-CNF) Φ, existe-t'il une affectation de poids k satisfaisant Φ ?

On note $(\textsc{wcnf-sat}, k)$ et $(\textsc{wcnf-3sat}, k)$

WEIGHTED-SAT sur les Π_t-circuits

Etant donné un circuit $C \in \Pi_t$, existe-t'il une affectation τ de poids k satisfaisant le circuit C ?

On note $(\text{WCS}[t], k)$

Le problème WEIGHTED-CNF-SAT

Etant donné une formule booléenne CNF (ou 3-CNF) Φ, existe-t'il une affectation de poids k satisfaisant Φ ?

On note $(\text{WCNF-SAT}, k)$ et $(\text{WCNF-3SAT}, k)$

Hypothèse 1

Aucun des problèmes $(\text{WCNF-3SAT}, k)$ et $(\text{WCS}[t], k)$, pour $t > 1$ n'est FPT

La W-hiérarchie

- La **classe** $W[1]$ contient l'ensemble des problèmes paramétrés (Q, κ) tels que $(Q, \kappa) <_{fpt} (\textsc{wcnf-3sat}, k)$

La W-hiérarchie

- La **classe** $W[1]$ contient l'ensemble des problèmes paramétrés (Q, κ) tels que $(Q, \kappa) <_{fpt} (\textsc{wcnf-3sat}, k)$
- Pour tout $t > 1$, la **classe** $W[t]$ contient l'ensemble des problèmes paramétrés (Q, κ) tels que $(Q, \kappa) <_{fpt} (\textsc{wcs}[t], k)$

La W-hiérarchie

- La **classe** $W[1]$ contient l'ensemble des problèmes paramétrés (Q,κ) tels que $(Q,\kappa) <_{fpt} (\textsc{wcnf-3sat}, k)$
- Pour tout $t > 1$, la **classe** $W[t]$ contient l'ensemble des problèmes paramétrés (Q,κ) tels que $(Q,\kappa) <_{fpt} (\textsc{wcs}[t], k)$
- La **classe** $W[P]$ contient l'ensemble des problèmes paramétrés (Q,κ) tels que $(Q,\kappa) <_{fpt} (\textsc{wcs}, k)$

La W-hiérarchie

- La **classe** $W[1]$ contient l'ensemble des problèmes paramétrés (Q,κ) tels que $(Q,\kappa) <_{fpt} (\textsc{wcnf-3sat}, k)$
- Pour tout $t > 1$, la **classe** $W[t]$ contient l'ensemble des problèmes paramétrés (Q,κ) tels que $(Q,\kappa) <_{fpt} (\textsc{wcs}[t], k)$
- La **classe** $W[P]$ contient l'ensemble des problèmes paramétrés (Q,κ) tels que $(Q,\kappa) <_{fpt} (\textsc{wcs}, k)$

Observation

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq W[t] \subseteq \cdots \subseteq W[P]$$

Un problème (Q, κ) est $\mathcal{C}$-**difficile**, si pour tout problème $(Q', \kappa') \in \mathcal{C}$, on a $(Q,' \kappa') <_{fpt} (Q, \kappa)$
Un problème (Q, κ) est $\mathcal{C}$-**complet** s'il appartient à $\mathcal{C}$ et est $\mathcal{C}$-difficile.

Un problème (Q, κ) est $\mathcal{C}$-**difficile**, si pour tout problème $(Q', \kappa') \in \mathcal{C}$, on a $(Q,' \kappa') <_{fpt} (Q, \kappa)$
Un problème (Q, κ) est $\mathcal{C}$-**complet** s'il appartient à $\mathcal{C}$ et est $\mathcal{C}$-difficile.

Lemme

- $(\textsc{wcnf-3sat}, k)$ est $W[1]$-complet
- $(\textsc{wcs}[t], k)$ est $W[t]$-complet

Un problème (Q,κ) est $\mathcal{C}$-**difficile**, si pour tout problème $(Q',\kappa') \in \mathcal{C}$, on a $(Q,'\kappa') <_{fpt} (Q,\kappa)$
Un problème (Q,κ) est $\mathcal{C}$-**complet** s'il appartient à $\mathcal{C}$ et est $\mathcal{C}$-difficile.

Lemme

- $(\textsc{wcnf-3sat}, k)$ est $W[1]$-complet
- $(\textsc{wcs}[t], k)$ est $W[t]$-complet

Exercices 2

- Montrer que s'il existe un problème $W[t]$-difficile qui soit FPT, alors FPT$= W[t]$
- Montrer que d'après l'hypothèse 1, $\forall t \geqslant 1$, FPT$\neq W[t]$

Exercice 3

Montrer que pour tout $t \geqslant 1$, un problème (Q, κ) est $W[t]$-difficile s'il existe un problème $W[t]$-difficile (Q', κ') tel que $(Q,' \kappa') <_{fpt} (Q, \kappa)$.

Exercice 3

Montrer que pour tout $t \geqslant 1$, un problème (Q, κ) est $W[t]$-difficile s'il existe un problème $W[t]$-difficile (Q', κ') tel que $(Q,' \kappa') <_{fpt} (Q, \kappa)$.

Lemme

Le problème WEIGHTED-ANTIMONOTONE CNF-2SAT (noté (WCNF-2SAT$^-$, k)) est complet pour $W[1]$.
Le problème MONOTONE-WCS[2] (noté (WCS$[2]^+$, k)) est complet pour $W[2]$.

1. Réductions paramétriques
   - Rappels et définitions
   - Exemples
2. La W-hiérarchie.
   - Circuits logiques
   - WEIGHTED-SAT
3. Deux exemples
   - Ensemble indépendant
   - Ensemble dominant
4. L'hypothèse de complexité exponentielle (ETH)

Théorème

Le problème (ENSEMBLE INDÉPENDANT, k) est $W[1]$-complet.

Théorème

Le problème ($\textsc{Ensemble indépendant}, k$) est $W[1]$-complet.

Preuve

- $W[1]$-**difficulté :** Réduction depuis ($\textsc{wcnf-2sat}^-, k'$).

Théorème

Le problème (ENSEMBLE INDÉPENDANT, k) est $W[1]$-complet.

Preuve

- $W[1]$-**difficulté :** Réduction depuis (WCNF-2SAT$^-$, k').
  Soit (Φ, k) une instance de (WCNF-2SAT$^-$, k).
  - variable $x_i \leftrightarrow$ sommet $x_i \in V$
  - clause $(\bar{x}_i \vee \bar{x}_j) \Leftrightarrow x_i x_j \in E$
  - $k' = k$

Théorème

Le problème (ENSEMBLE INDÉPENDANT, k) est $W[1]$-complet.

Preuve

- $W[1]$-**difficulté :** Réduction depuis (WCNF-2SAT$^-$, k').
  Soit (Φ, k) une instance de (WCNF-2SAT$^-$, k).
  - variable $x_i \leftrightarrow$ sommet $x_i \in V$
  - clause $(\bar{x}_i \vee \bar{x}_j) \Leftrightarrow x_i x_j \in E$
  - $k' = k$

  Il existe une affectation positive de poids k' ssi il existe un ensemble indépendant de taille k.

Théorème

Le problème (ENSEMBLE INDÉPENDANT, k) est $W[1]$-complet.

Preuve

- $W[1]$-**difficulté :** Réduction depuis (WCNF-2SAT$^-$, k').
- $W[1]$-**complétude :** On montre que (ENSEMBLE INDÉPENDANT, k) $<_{fpt}$ (WCNF-2SAT$^-$, k')

Théorème

Le problème (ENSEMBLE INDÉPENDANT, k) est $W[1]$-complet.

Preuve

- $W[1]$-**difficulté :** Réduction depuis (WCNF-2SAT$^-$, k').
- $W[1]$-**complétude :** On montre que (ENSEMBLE INDÉPENDANT, k) $<_{fpt}$ (WCNF-2SAT$^-$, k')
  $\longrightarrow$ Même construction.

Théorème

Le problème (ENSEMBLE INDÉPENDANT, k) est $W[1]$-complet.

Preuve

- $W[1]$-**difficulté :** Réduction depuis (WCNF-2SAT$^-$, k').
- $W[1]$-**complétude :** On montre que (ENSEMBLE INDÉPENDANT, k) $<_{fpt}$ (WCNF-2SAT$^-$, k')

Corollaire

Le problème (CLIQUE, k) est $W[1]$-complet.

Théorème

Le problème (ENSEMBLE DOMINANT, k) est $W[2]$-complet.

Théorème

Le problème (ENSEMBLE DOMINANT, k) est $W[2]$-complet.

Preuve

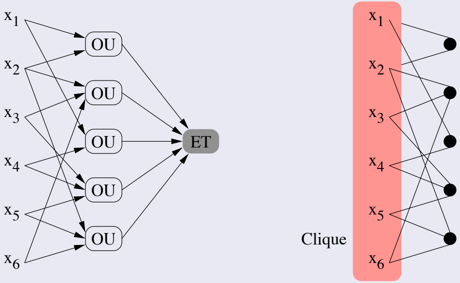
- $W[2]$-**difficulté :** Réduction depuis (WCS$[2]^+$, k').

Théorème

Le problème (ENSEMBLE DOMINANT, k) est $W[2]$-complet.

Preuve

- $W[2]$-**difficulté :** Réduction depuis (WCS$[2]^+, k'$).
  Soit C un Π_2-circuit monotone.

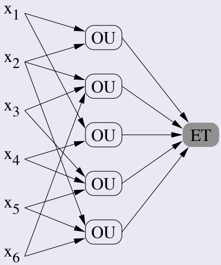


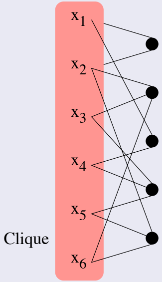
Théorème

Le problème (ENSEMBLE DOMINANT, k) est $W[2]$-complet.

Preuve

- $W[2]$-**difficulté :** Réduction depuis $(\text{WCS}[2]^+, k')$.
  Soit C un Π_2-circuit monotone.





C admet une affectation positive de poids k ssi G_c admet un ensemble dominant de taille k

Théorème

Le problème $(\textsc{Ensemble dominant}, k)$ est $W[2]$-complet.

Preuve

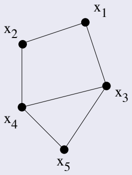
- $W[2]$-**difficulté :** Réduction depuis $(\textsc{wcs}[2]^+, k')$.
- $W[2]$-**complétude :** On montre que $(\textsc{Ensemble dominant}, k) <_{fpt} (\textsc{wcs}[2]^+, k')$

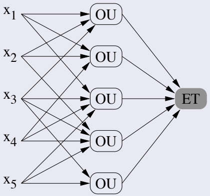
Théorème

Le problème (ENSEMBLE DOMINANT, k) est $W[2]$-complet.

Preuve

- $W[2]$-**difficulté :** Réduction depuis $(\text{WCS}[2]^+, k')$.
- $W[2]$-**complétude :** On montre que $(\text{ENSEMBLE DOMINANT}, k) <_{fpt} (\text{WCS}[2]^+, k')$
  Soit un graphe G, on construit un circuit C_G Π_2-monotone.



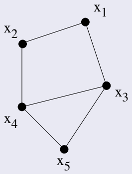


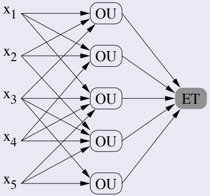
Théorème

Le problème (ENSEMBLE DOMINANT, k) est $W[2]$-complet.

Preuve

- $W[2]$-**difficulté :** Réduction depuis (WCS$[2]^+, k'$).
- $W[2]$-**complétude :** On montre que (ENSEMBLE DOMINANT, k) $<_{fpt}$ (WCS$[2]^+, k'$)
  Soit un graphe G, on construit un circuit C_G Π_2-monotone.





C_G admet une affectation positive de poids k ssi G admet un ensemble dominant de taille k

Un problème Q est **sous-exponentiel** s'il admet un algorithme de complexité $2^{o(n)}$ (par exemple $2^{\sqrt{n}}$).

Un problème Q est **sous-exponentiel** s'il admet un algorithme de complexité $2^{o(n)}$ (par exemple $2^{\sqrt{n}}$).

Classe SNP

La **classe SNP** contient tous les problèmes exprimables par une formule existentielle du second-ordre dont la partie du premier ordre est universelle.

La classe **SNP-contrainte** contient tous les problèmes ... (on s'en fout ;-)

Un problème Q est **sous-exponentiel** s'il admet un algorithme de complexité $2^{o(n)}$ (par exemple $2^{\sqrt{n}}$).

Classe SNP

La **classe SNP** contient tous les problèmes exprimables par une formule existentielle du second-ordre dont la partie du premier ordre est universelle.
La classe **SNP-contrainte** contient tous les problèmes . . . (on s'en fout ;-)

Théorème

FPT$\neq W[1]$ sauf si tous les problèmes SNP-contraints admettent un algorithme sous-exponentiel.

Un problème Q est **sous-exponentiel** s'il admet un algorithme de complexité $2^{o(n)}$ (par exemple $2^{\sqrt{n}}$).

Classe SNP

La **classe SNP** contient tous les problèmes exprimables par une formule existentielle du second-ordre dont la partie du premier ordre est universelle.
La classe **SNP-contrainte** contient tous les problèmes ... (on s'en fout ;-)

Théorème

FPT$\neq W[1]$ sauf si tous les problèmes SNP-contraints admettent un algorithme sous-exponentiel.

VERTEX-COVER; 3-COLORATION,... appartiennent à SNP-contraint (ils sont complets pour cette classe!).

Hypothèse de complexité exponentielle (EHT)

Certains problèmes SNP-contraints n'admettent pas d'algorithme sous-exponentiel.

Hypothèse de complexité exponentielle (EHT)

Certains problèmes SNP-contraints n'admettent pas d'algorithme sous-exponentiel.

Théorème

Si le problème ENSEMBLE INDÉPENDANT peut être résolu en temps $n^{o(k)}$, alors le problème VERTEX COVER peut-être résolu en temps $O(2^{o(k)}n^{O(1)})$.

Hypothèse de complexité exponentielle (EHT)

Certains problèmes SNP-contraints n'admettent pas d'algorithme sous-exponentiel.

Théorème

Si le problème ENSEMBLE INDÉPENDANT peut être résolu en temps $n^{o(k)}$, alors le problème VERTEX COVER peut-être résolu en temps $O(2^{o(k)}n^{O(1)})$.

$\Rightarrow$ VERTEX COVER admet un algorithme sous-exponentiel, et donc ETH est fausse (car VERTEX COVER est SNP-complet).

Hypothèse de complexité exponentielle (EHT)

Certains problèmes SNP-contraints n'admettent pas d'algorithme sous-exponentiel.

Théorème

Si le problème ENSEMBLE INDÉPENDANT peut être résolu en temps $n^{o(k)}$, alors le problème VERTEX COVER peut-être résolu en temps $O(2^{o(k)}n^{O(1)})$.

Théorème

Sauf si ETH est fausse, le problème ENSEMBLE INDÉPENDANT ne peut pas être résolu en temps $f(k).n^{o(k)}$ avec f une fonction récursive.

Définition: réduction paramétrée linéaire

Une réduction paramétrée est linéaire si elle réduit une instance $(x, \kappa(x))$ d'un problème (Q, κ) en une instance $(x', \kappa'(x'))$ d'un problème (Q', κ') avec

- $\kappa'(x') = O(\kappa(x))$ et
- $|x'| = |x|^{O(1)}$

Définition: réduction paramétrée linéaire

Une réduction paramétrée est linéaire si elle réduit une instance $(x, \kappa(x))$ d'un problème (Q, κ) en une instance $(x', \kappa'(x'))$ d'un problème (Q', κ') avec

- $\kappa'(x') = O(\kappa(x))$ et
- $|x'| = |x|^{O(1)}$

Lemme

Soit (Q, κ) un problème linéairement réductible en (Q', κ'). Si (Q', κ') admet un algorithme de complexité $f(k)n^{o(k)}$, alors (Q, κ) admet un algorithme $g(k)n^{o(k)}$.

Définition: réduction paramétrée linéaire

Une réduction paramétrée est linéaire si elle réduit une instance $(x, \kappa(x))$ d'un problème (Q, κ) en une instance $(x', \kappa'(x'))$ d'un problème (Q', κ') avec

- $\kappa'(x') = O(\kappa(x))$ et
- $|x'| = |x|^{O(1)}$

Lemme

Soit (Q, κ) un problème linéairement réductible en (Q', κ'). Si (Q', κ') admet un algorithme de complexité $f(k)n^{o(k)}$, alors (Q, κ) admet un algorithme $g(k)n^{o(k)}$.

Corollaire

Sauf si ETH est fausse, aucun problème $W_l[1]$-difficile ne peut être résolu en temps $f(k)n^{o(k)}$.
($\textsc{wcnf-2sat}^{-} \in W_l[1]$-difficile)

Théorème

Le problème VERTEX COVER peut être résolu en temps $O(2^{o(k)}n^{O(1)})$ ssi l'ETH est fausse.

Preuve

Théorème

Le problème VERTEX COVER peut être résolu en temps $O(2^{o(k)}n^{O(1)})$ ssi l'ETH est fausse.

Preuve

- $\Rightarrow$ puisque VERTEX COVER est complet pour SNP-contraint, alors tous les problèmes de cette classe admettraient un tel algorithme. Donc ETH serait fausse.

Théorème

Le problème VERTEX COVER peut être résolu en temps $O(2^{o(k)}n^{O(1)})$ ssi l'ETH est fausse.

Preuve

- $\Rightarrow$ puisque VERTEX COVER est complet pour SNP-contraint, alors tous les problèmes de cette classe admettraient un tel algorithme. Donc ETH serait fausse.
- $\Leftarrow$ Si ETH est fausse, alors VERTEX COVER admet un algorithme de complexité $O(2^{o(n)})$. En kernalisant, une instance de VERTEX COVER, on obtiendrait donc un algorithme de complexité $O(2^{o(k)}n^{O(1)})$

Théorème

Pour tout entier $t \geqslant 1$, si $W[t]$ =FPT implique $W[t+1]$ =FPT, alors

- soit la complexité du problème $\text{WCS}[t+1]$ est $n^{\Omega(k)}$;
- soit $\text{WCS}[t+1]$ est FPT.