

Complejidad Parametrizada (Parte 2)

Iván Rubio Venegas

Pontificia Universidad Católica

iirubio@uc.cl

9 de Junio, 2016

1 Repaso de conceptos

- Lenguaje parametrizado
- Clase FPT
- Clase para-NP
- Clase XP
- Reducciones fpt

2 Jerarquía W

- Motivación
- Jerarquía W
- Problemas completos

3 Clase $W[P]$

- Clase $W[P]$
- Relación con otras clases

Lenguaje Parametrizado

Consideremos el alfabeto Σ .

Una parametrización $\kappa : \Sigma^* \rightarrow \mathbb{N}$ es una función computable en tiempo polinomial.

Un language parametrizado es un par (L, κ) , donde $L \subseteq \Sigma^*$ y κ es una parametrización.

- e.g. (SAT, κ) con $\kappa(\varphi) = |\text{var}(\varphi)|$

Fixed-Parameter Tractable

Una MT M es *fixed-parameter tractable* con respecto a κ (κ -fpt) si existe una función computable $f : \mathbb{N} \rightarrow \mathbb{N}$ y un polinomio p tal que el tiempo de ejecución de M con input w es a lo más $f(\kappa(w)) \cdot p(|w|)$.

Un lenguaje parametrizado (L, κ) es *fixed-parameter tractable* (fpt) si existe una MT M que sea κ -fpt y tal que $L(M) = L$.

- Cada vez que hablemos de un lenguaje, tendrá asignada una parametrización κ .

Definimos la clase FPT como el conjunto de todos los lenguajes que sean fpt.

Una MT *no determinista* M es κ -fpt si existe una función computable $f : \mathbb{N} \rightarrow \mathbb{N}$ y un polinomio p tal que el tiempo de ejecución de M con input w es a lo más $f(\kappa(w)) \cdot p(|w|)$.

Un lenguaje parametrizado (L, κ) pertenece a para-NP si existe una MT no-determinista M que sea κ -fpt y tal que $L(M) = L$

Un lenguaje (L, κ) pertenece a XP si existe una función computable $f : \mathbb{N} \rightarrow \mathbb{N}$ y una MT M que con input w decide si $w \in L$ en tiempo

$$|w|^{f(\kappa(w))} + f(\kappa(w))$$

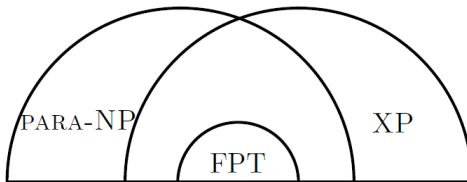
- ¿Qué complejidad tendría el lenguaje de la i -slice de (L, κ) ?

Es claro que $\text{FPT} \subseteq \text{XP}$.

- ¿Puede ser que $\text{FPT} = \text{XP}$?

¿En qué quedamos?

Relaciones entre las clases vistas hasta este momento



Consideremos los lenguajes parametrizados (L_1, κ_1) y (L_2, κ_2) . Un mapeo $R : \Sigma^* \rightarrow \Sigma^*$ se dice que es reducción-fpt de (L_1, κ_1) a (L_2, κ_2) si:

- ① Para todo $w \in \Sigma^*$, $w \in L_1 \Leftrightarrow R(w) \in L_2$
- ② R es computable por una κ -fpt MT. Es decir, existe una función computable f y un polinomio p tal que $R(w)$ es computable en tiempo $f(\kappa_1(w)) \cdot p(|w|)$.
- ③ Existe una función computable $g : \mathbb{N} \rightarrow \mathbb{N}$ tal que $\kappa_2(R(w)) \leq g(\kappa_1(w))$ para todo $w \in \Sigma^*$
 - Pronto veremos para qué sirve esta condición

Si existe dicha reducción, diremos que (L_1, κ_1) es fpt-reducible a (L_2, κ_2) , o $(L_1, \kappa_1) \leq^{fpt} (L_2, \kappa_2)$

Nuevas clases de complejidad

Existen problemas naturales que las clases vistas hasta ahora no capturan. Para el problema parametrizado p -CLIQUE no se conoce una fpt-reducción a ninguna de estas tres clases. Por esto, surge la definición de un nuevo conjunto de clases de complejidad parametrizada.

Antes de poder definir formalmente la jerarquía W , se tienen que definir algunos conceptos de lógica.

Definición

Π_t y Σ_t se definen inductivamente de la siguiente forma:

- Π_0 y Σ_0 corresponden a las clases de fórmulas sin cuantificadores.
- Para $t \geq 0$
 - Σ_{t+1} es la clase de fórmulas de la forma $\exists x_1 \dots \exists x_k \varphi$ donde $\varphi \in \Pi_t$
 - Π_{t+1} es la clase de fórmulas de la forma $\forall x_1 \dots \forall x_k \varphi$ donde $\varphi \in \Sigma_t$

Ejemplo

$\exists x_1 \exists x_2 \forall y \exists z \forall u_1 \forall u_2 \forall u_3 (R(x_1, y, z) \vee R(x_2, u_3, u_2) \vee \neg R(y, u_1, x_2))$ es una Σ_4 -fórmula.

Sea $\varphi(X)$ una fórmula en lógica de primer orden con una variable relacional libre X de aridad s . Se define $L(\varphi(X))$ como el lenguaje de todos los pares $(\mathfrak{A}, k)$ donde $\mathfrak{A}$ es una estructura de primer orden para la cual existe una relación $S \subseteq \text{dom}(\mathfrak{A})^s$ de cardinalidad k tal que $\mathfrak{A} \models \varphi(S)$.

Definición

Dada una fórmula en LPO $\varphi(X)$ con una variable relacional libre X , el problema parametrizado $P\text{-}WD_\varphi$ se define como $(L(\varphi(X)), \kappa)$, donde $\kappa((\mathfrak{A}, k)) = k$

Definición

Para la clase Φ de las fórmulas en LPO con una variable relacional libre, $P\text{-}WD_\Phi$ corresponde al conjunto $\bigcup_{\varphi \in \Phi} P\text{-}WD_\varphi$

Definición

$\Pi_t(X)$ es el conjunto de fórmulas de forma Π_t descrita anteriormente donde la fórmula φ sin cuantificadores incluye adicionalmente la variable relacional X .

Jerarquía W

Sea $t \in \mathbb{N}$. $W[t]$ es el conjunto de problemas fpt-reducibles a algún problema en $P\text{-}WD_{\Pi_t(X)}$. Esto se denota como $[P\text{-}WD_{\Pi_t(X)}]^{fpt}$

Teorema

$P - CLIQUE$ es *fpt – completo* para $W[1]$.

Ejercicio

$P-CLIQUE \in W[1]$.

$$\varphi(X) = \forall x \forall y (X(x) \wedge X(y) \wedge x \neq y \rightarrow E(x, y))$$

Ejercicio

$P - DOMINATING - SET \in W[2]$.

$$\varphi(X) = \forall x \exists y (\neg X(x) \rightarrow (E(x, y) \wedge X(y)))$$

Definición

- 1 Sea Σ un alfabeto y $\kappa : \Sigma^* \rightarrow \mathbb{N}$ una parametrización. Una máquina de Turing $\mathbb{M}$ con input el alfabeto Σ es llamada κ - *restricted* si existen funciones computables $f, h : \mathbb{N} \rightarrow \mathbb{N}$ y un polinomio $p \in \mathbb{N}_0[X]$ tal que en cada ejecución con input $x \in \Sigma^*$ la máquina $\mathbb{M}$ lleva a cabo a lo más $f(k) \cdot p(n)$ pasos, a lo más $h(k) \cdot \log n$ de ellos siendo no determinísticos. Siendo $n := |x|$ y $k := \kappa(x)$.
- 2 $W[P]$ es la clase de todos los problemas parametrizados (Q, κ) que pueden ser decididos por una máquina de Turing no determinista κ - *restricted*.

Relación entre $W[P]$, FPT, XP y para-NP

Proposición

$$FPT \subseteq W[P] \subseteq XP \cap para - NP$$

Demostración

- $FPT \subseteq W[P]$ y $W[P] \subseteq para - NP$ son triviales.
- $W[P] \subseteq XP$ se cumple porque la simulación de $h(k) \cdot \log n$ pasos no deterministas de una máquina de Turing con s estados por un algoritmo determinista, requiere tiempo $O(s^{h(k) \cdot \log n}) = n^{O(h(k))}$.

Relación entre $W[P]$ y la jerarquía W

Proposición

$p\text{-}WD - FO \subseteq W[P]$. Por lo tanto $W[t] \subseteq W[P]$ para todo $t \geq 1$.

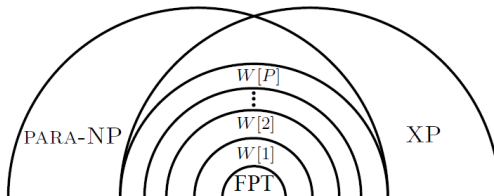
Demostración

Sea $\varphi(X)$ una fórmula en LPO, donde X tiene aridad s . Un algoritmo para el problema $p\text{-}WD_\varphi$ es el siguiente:

- 1 Dada una estructura $\mathfrak{A}$ y $k \in \mathbb{N}$, de forma no determinista adivina k veces una tupla en A^s
- 2 Deterministicamente verifica que el conjunto S de tuplas tiene cardinalidad k y $\mathfrak{A} \models \varphi(S)$.
- 3 Adivinar S requiere $s \cdot k \cdot \log |A|$ bits no deterministas.
- 4 La verificación $\mathfrak{A} \models \varphi(S)$ requiere tiempo polinomial.
- 5 Por lo tanto, $p\text{-}WD_\varphi$ está en $W[P]$

Relación entre $W[P]$ y la jerarquía W

Relaciones entre las clases vistas



-  [Flum, J., Grohe, M \(2006\)](#)
Parameterized Complexity Theory (Libro)
-  [Ugarte, M. \(2012\) \(Diapositivas\)](#)
An introduction to Parameterized Complexity Theory
-  [Grez, A. \(2016\)](#)
Complejidad Parametrizada (Parte 1) (Diapositivas)

¿Preguntas?