Complejidad Parametrizada de Consultas a Bases de Datos

Mauricio Vinés Neuwirth
Departamento de Ciencias de la Computación
Pontificia Universidad Católica de Chile
mvines@ing.puc.cl

Temas

- Introducción. Conceptos preliminares: Lógica de Primer Orden.
- El problema de Chequeo de Modelos.
- Introducción a la Complejidad Paramétrica: La case FPT.
- Intratabilidad: WSAT y la Jerarquía W.
- QBF y la Jerarquía AW.
- Chequeo de Modelo y Jerarquía A.

Introducción

La evaluación de consultas a bases de datos relacionales es un problema complejo.

- ◇ Evaluar consultas de Primer Orden es PSPACE-completo.
- ◇ Evaluar consultas conjuntivas básicas es NP-completo.

Se tiene una *complejidad combinada*: depende del **tamaño de la base de datos** y del **largo de la consulta**.

Pero, mirando resultados prácticos:

- ⇒ La mayor complejidad es causada por consultas "largas".
- ⇒ En la práctica, se emplean consultas pequeñas.

Introducción (cont)

Los algoritmos más eficientes para responder consultas de Primer Orden, o bien, consultas conjuntivas toman un tiempo $n^{\Theta(L)}$.

⇒ n es tamaño de la base de datos (número de registros).

⇒ L es el tamaño de la consulta.

¿Que pasa si ignoramos el tamaño de la consulta?

Supongamos que L es una constante fija...

La complejidad sólo dependería del tamaño de la base de datos (*complejidad de datos*).

⇒ La *complejidad de datos* toma tiempo polinomial.

⇒ Pero para valores relativamente pequeños de L la complejidad es inaceptable.

¿Alguna otra forma de ver el problema?

Breve Interludio de Lógica de Primer Orden

Definiciones:

- ◇ Un *vocabulario* es un conjunto de símbolos de relación, de aridad fija. Corresponde al esquema de la base de datos. Se denota $\mathfrak{L}$.

- ◇ Una estructura $\mathfrak{A}$ (compatible con $\mathfrak{L}$) consta de:
  - Un universo A
  - Una relación r-aria $R^r \subseteq A^r$, para cada símbolo de relación $R \in \mathfrak{L}$, de aridad r.

  Una estructura corresponde a una instancia de la base de datos.

  Todas las estructuras que se consideran aquí tienen universo finito.

- ◇ Una fórumla (atómica) es una expresión del tipo $x = y$ o bien $R^r x_1, x_2, \ldots, x_r$, para R de aridad r.

Interludio de Lógica de Primer Orden (cont)

◇ Si φ y ψ son fórmulas, también lo son $(\neg\varphi)$, $(\varphi \star \psi)$ con $\star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, $\forall x\varphi$, $\exists x\varphi$

◇ $voc(\varphi) \subseteq \mathfrak{L}$ es el conjunto de relaciones que aparecen en φ.

◇ Las *variables ligadas* de φ son aquellas que están cuantificadas.

◇ Las *variables libres* de φ son aquellas que no lo están.

◇ Si $\mathfrak{A}$ es una estructura, y $a_1, a_2, \ldots, a_n \in A$, y $\varphi(x_1, x_2, \ldots x_n)$ es una fórmula, tal que $voc(\varphi) \subseteq \mathfrak{L}$, entonces denotamos con $\mathfrak{A} \models \varphi(a_1, \ldots a_n)$ que $\mathfrak{A}$ **satisface a** φ si se interpreta cada variable x_i por a_i, respectivamente.

Interludio de Lógica de Primer Orden (cont)

◇ Una fórmula está en *Forma Normal Prenex* si es que es de la forma:

$$Q_1 x_1 \ldots Q_k x_k \varphi$$

Con $Q_1, \ldots Q_k \in \{\forall, \exists\}$, y si es que φ sólo tiene variables libres.

◇ Para $k \geq 1$, se define la clase Σ_k la clase de fórmulas en Forma Normal Prenex con $k-1$ alternación de cuantificadores, partiendo con uno existencial.

Análogamente, se define Π_k cuando existe la misma cantidad de alternación de cuantificadores, pero partiendo con uno universal.

En el ámbito de las Bases de Datos

◇ Se define $\varphi(\mathfrak{A}) := \{a_1, \ldots a_n \in A | \mathfrak{A} \models \varphi(a_1, \ldots a_n)\}$
Si φ corresponde a una consulta, $\varphi(\mathfrak{A})$ corresponde a la respuesta.

Si φ no tiene variables libres, se puede considerar lo siguiente:

– $\mathfrak{A} \models \varphi \Leftrightarrow \varphi(\mathfrak{A}) =$ **verdadero**.

– Si $voc(\mathfrak{A}) \nsubseteq \mathfrak{L}$, entonces $\varphi(\mathfrak{A}) = \varnothing$

◇ Una *consulta conjuntiva* es una fórmula de la forma

$$\exists x_1 \ldots \exists x_k (\varphi_1 \wedge \ldots \wedge \varphi_n)$$

◇ El mapeo $\mathfrak{A} \mapsto \varphi(\mathfrak{A})$ es la consulta definida por φ. Interesa el problema de evaluación para varias clases de fórmulas Φ.

Consultas a Bases de Datos y Chequeo de Modelo

Con las herramientas lógicas que disponemos, se puede plantear el problema de evaluar una consulta a una base de datos formalmente:

EVAL(Φ)		
	Entrada:	Estructura $\mathfrak{A}$, fórmula $\varphi \in \Phi$
	Problema:	Computar $\varphi(\mathfrak{A})$

Para efectos de la complejidad computacional del problema de evaluación de consultas, se puede considerar el problema de desición equivalente, el *Problema de chequeo de modelo*:

MC(Φ)		
	Entrada:	Estructura $\mathfrak{A}$, fórmula $\varphi \in \Phi$
	Problema:	Decidir $\varphi(\mathfrak{A}) \neq \varnothing$

Comentarios

- Existen algoritmos de *model-checking* en otras lógicas cuya complejidad es $2^{\Theta(L)} \cdot n$.
- Si bien es exponencial, para valores pequeños de L es más aceptable que $n^{\Theta(L)}$

¿Existirá una cota así para el problema de evaluación de consultas (o el chequeo de modelos en Lógica de Primer Orden)?

Este tipo de preguntas ha dado origen a la *Teoría de Complejidad Parametrizada.*

Complejidad Parametrizada: Definiciones

Un problema parametrizado es es un conjunto $P \subseteq \Sigma^* \times \Pi^*$, donde Σ y Π son alfabetos finitos.

Si $(x, y) \in \Sigma^* \times \Pi^*$ es una instancia de un problema parametrizado, x corresponde a la *entrada* e y al *parámetro*.

Por ejemplo:

p-Clique

Entrada:	Grafo G
Parámetro:	$k \in \mathbb{N}$
Problema:	Decidir si G tiene un Clique de tamaño k

Complejidad Parametrizada: Definiciones (cont)

Es útil dar una definición paramétrica del model-checking:

p-MC(Φ)		
	Entrada:	Estructura $\mathfrak{A}$
	Parámetro:	$\varphi \in \Phi$
	Problema:	Decidir si $\varphi(\mathfrak{A}) \neq \varnothing$

- Se pueden definir versiones parametrizadas de varios problemas conocidos...
- Por ejemplo, para algoritmos de grafos, costos, tamaños de subgrafos, pueden ser parámetros.
- En general, se pueden parametrizar problemas clásicos en la clase P aunque no sea evidente cuál es el parámetro adecuado.

Lo Tratable en versión Parametrizada

Ahora interesa saber que resultados de complejidad computacional se pueden obtener en este nuevo marco teórico.

Primero veremos la clase de los problemas tratables (FPT), para luego pasar a las jerarquías de problemas intratables.

Definición: Un problema parametrizado $P \in \Sigma^* \times \Pi^*$ es *Tratable con parámetro Fijo* si:

- existe una función computable $f : \mathbb{N} \to \mathbb{N}$,
- una constante $c \in \mathbb{N}$
- y un algoritmo tal que:
  dado un par $(x, y) \in \Sigma^* \times \Pi^*$,
  decide si $(x, y) \in P$ en a lo más $f(|y|) \cdot |x|^c$.

La Clase FPT

$P \in FPT \Leftrightarrow P$ es tratable con parámetro fijo.

Una forma alternativa de escribir la definición de Tratabilidad con Parámetro fijo es la siguiente:

"Un problema parametrizado $P \in \Sigma^* \times \Pi^*$ es *Tratable con parámetro Fijo* si y sólo si existen algoritmos $\mathfrak{D}$ y $\mathfrak{E}$ y constantes c y d, tales que:

- Para cada parámetro $y \in \Pi^*, \mathfrak{D}$ produce un string $\mathfrak{D}(y) \in \Pi^*$.
- Para la entrada $(x, \mathfrak{D}(y))$, $\mathfrak{E}$ decide si $(x, y) \in P$ en a lo más $d \cdot |x|^e$ pasos."

Según lo anterior, podríamos decir que el problema de chequeo de modelo es tratable con parámetro fijo si y sólo si existe un *algoritmo de optimización de consulta* de complejidad arbitraria y un *algritmo de evaluación* que evalúe la consulta optimizada en tiempo polinomial.

Reducciones Parametrizadas

Sean P y P' problemas paramétricos (no necesariamente con mismos alfabetos).

Una m-reducción de P a P' es un algoritmo que computa para cada instancia $(x, y) \in P$, una instancia $(x', y') \in P'$, en tiempo $f(|y|) \cdot |x1^c$ tal que
$|y| \leq g(|y|)$ y

$$(x, y) \in P \Longleftrightarrow (x', y') \in P'$$

f,g funciones computables cualesquiera, definidas $\mathbb{N} \rightarrow \mathbb{N}$ y c constante en $\mathbb{N}$.

Si existe tal reducción, entonces P es m-reducible a P' y se denota $P \leq_m^{fp} P'$.

<u>Observaciones:</u>

◇ Para abreviar $(P \leq_m^{fp} P' \wedge P' \leq_m^{fp} P)$
escribimos $P \equiv_m^{fp} P'$.

◇ $[P]_m^{fp}$ corresponde al conjunto de los problemas parametrizados que son m-reducibles a P.

Reducciones Paramétricas: Ejemplo

$$\boxed{\text{P-CLIQUE} \leq_m^{fp} \text{P-MC(CQ)}}$$

p-MC(CQ) es el problema de chequeo de modelo parametrizado para consultas conjuntivas.

Mapeamos cada instancia (G, k) de p-Clique a la instancia (G, φ_k) de p-MC(CQ) donde:

$$\varphi_k := \exists x_1 \ldots \exists x_k \bigwedge_{1 \leq j \leq k} E x_i x_j$$

$E x_i x_j$ corresponde al predicado “Existe una arista entre los nodos x_i y x_j ”.

Chequeo de Modelos, nuevamente

Una versión del problema del chequeo de modelo parametrizado, más acorde con nuestro interés en su comlejidad computacional sería la siguiente:

P-MC'(Φ)		
	Entrada:	Estructura $\mathfrak{A}$, fórmula $\varphi \in \Phi$
	Parámetro:	$\|\|\varphi\|\|$
	Problema:	Decidir si $\varphi(\mathfrak{A}) \neq \emptyset$

Aqui $||\varphi||$ denota el largo de la fórumla φ.

Se puede demostrar que:

$$p - MC'(\Phi) \equiv_m^{fp} p - MC(\Phi)$$

¿Que sigue?

Ahora nos interesa otra pregunta, ¿es tratable con parámetro fijo el problema de chequeo de modelo?

Por ejemplo, *p-Coloración de Grafos*, parametrizado por el número de colores, no es tratable con parámetro fijo a menos que $P = NP$ (Notar que 3-Coloración de Grafos es NP-completo).

Aparentemente p-Clique tampoco sería tratable en términos de parámetro fijo.

Existen varias clases de intratabilidad parametrizada.

Estas se definen en términos de la satisfacibilidad de **fórmulas proposicionales**.

Intratabilidad Parametrizada: WSAT

- Cada fórumla proposicional θ se puede representar como un árbol $\mathfrak{T}_\theta$.

- Haremos una distinción entre $\wedge$ (*conjunción pequeña*) y $\bigwedge$ (*conjunción grande*).
- La primera corresponde a conjunciones de dos fórumlas, mientras que la segunda corresponde a la conjunción entre conjuntos finitos, arbitrarios de fórmulas.

- Asimismo, se distingue entre $\vee$ y $\bigvee$ (*pequeña* y *gran* disyunción, respectivamente).
- La profundidad del árbol corresponde al máximo número de nodos etiquetados con conectivas, desde la raíz hasta una hoja.

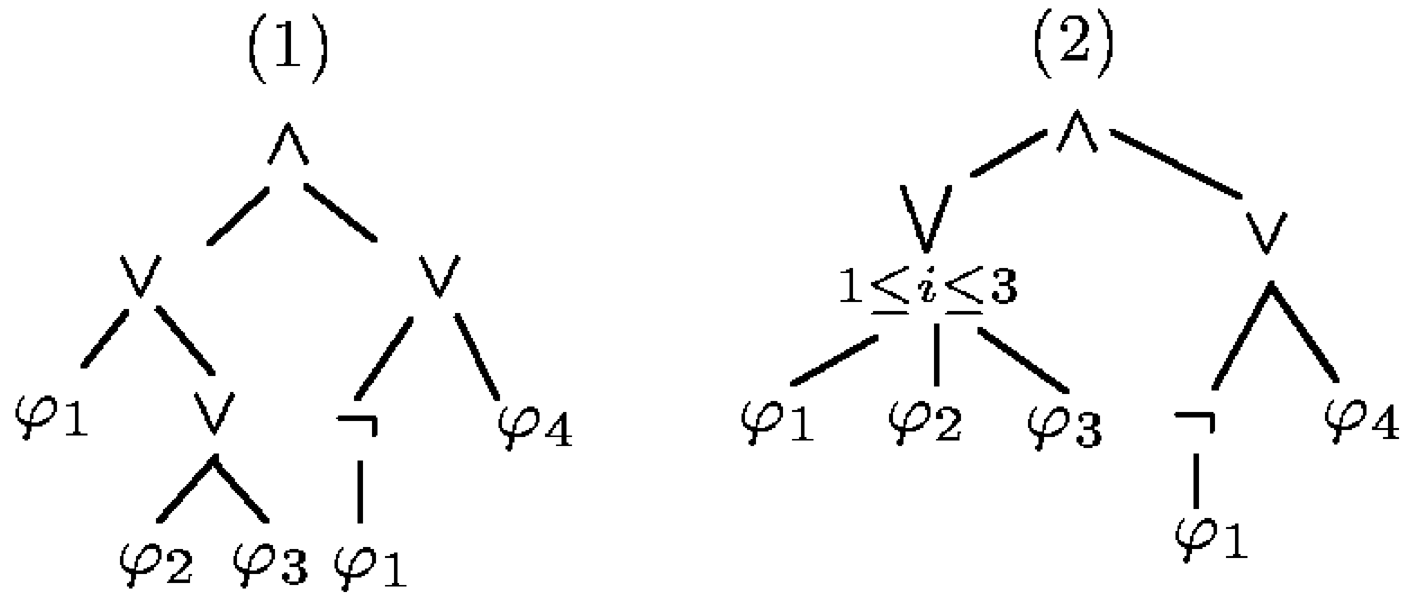
- Diremos que una fórmula es pequeña si es que contiene sólo *conjunciones y disyunciones pequeñas.*

Ejemplos de fórmulas como árboles

Por ejemplo las fórmulas

$$(1)\ (\varphi_1 \vee \varphi_2 \vee \varphi_3) \wedge (\neg\varphi_1 \vee \varphi_4)$$
$$(2)\ (\bigvee_{1\leq i\leq 3} \varphi_i) \wedge (\neg\varphi_1 \vee \varphi_4)$$

Se representarían como los siguientes árboles:



Clases de fórmulas proposicionales

Definimos $\Gamma_0 = \Delta_0$, como la clase de las fórmulas pequeñas.

Para $t \geq 1$ definitmos Γ_t como la clase de todas las *conjunciones grandes* de fórmulas en Δ_{t-1}.

Asimismo, Δ_t es la clase de todas las *disyunciones grandes* de fórmulas en la clase Γ_{t-1}.

Notar que una fórmula en Γ_t es equivalente a una en Γ_0, pero esto aumenta la profundidad de la fórmula.

Se define $\Gamma_{t,d}$ como la clase de todas las fórmulas que están en Γ_t y cuyas subfórmulas tienen profundidad a lo más d (Análogamente se define para $\Delta_{t,d}$).

El *peso* de una asignación es el número de variables proposicionales que la asignación hace verdaderas.

WSAT y la Jerarquía W

Estamos en condiciones de definir el problema de *satisfacibilidad con pesos* para cualquier clase de fórmula proposicional Θ.

WSAT(Θ)

Entrada:	$\theta \in \Theta$
Parámetro:	$k \in \mathbb{N}$
Problema:	Decidir si θ tiene una asignación que de peso k la satisfaga .

Se define la **Jerarquía W**:

$$\text{Para cada } t \geq 1 \; W[t] := \bigcup_{d \geq 0} [WSAT(\Gamma_{t,d})]_m^{fp}$$

La Jerarquía W

Un problema parametrizado está en la jerarquía W si es que existe un $d \geq 0$ tal que el problema es reducible parametrizadamente al problema de satisfacibilidad con pesos para $\Gamma_{t,d}$.

Un resultado muy importante, para efectos del objetivo inicial es:

Teorema:

p-MC(CQ) es W[1]-completo bajo m-reducciones parametrizadas.

La demostración de este teorema no es fácil y se requiere de varios lemas...

Se presentarán las ideas que van guiando la demostración.

p-MC(CQ) es W[1]-completo

1. p-MC(CQ) $\leq_m^{fp}$ p-MC(CQ[binary]).

   Aquí $p - MC(CQ[binary])$ corresponde al problema de chequeo de modelo con un vocabulario a lo más binario. Para demostrar este lema se requiere construir una estructura $\mathfrak{A}'$ (con vocabulario binario) a partir de una estructura $\mathfrak{A}$ del problema original, agregando nuevos elementos al dominio y construyendo las relaciones ad-hoc.

2. p-MC(CQ[$\mathfrak{L}$]$\equiv_m^{fp}$p-MC(CQ)

   para cualquier vocabulario $\mathfrak{L}$ que sea al menos binario.

   La idea es llegar a que

$$p - MC(CQ[binary]) \equiv_m^{fp} p - MC(CQ)$$

p-MC(CQ) es W[1]-completo (cont)

3. p-MC(CQ[binary])$\in W[1]$.

   Aquí es necesario definir una fórmula $\theta \in \Gamma_{1,1}$, tal que $\mathfrak{A} \models \varphi$ si y sólo si φ tiene una asignación que la satisface con peso k.

4. Con las demostraciones anteriores se puede demostrar que

   $[p - MC(CQ)]_m^{fp} \subseteq W[1]$.

5. WSAT($\Gamma_{1,d}$)$\leq_m^{fp}$p-MC(CQ).

6. Con lo anterior se demuestra que $W[1] \subseteq [p - MC(CQ)]_m^{fp}$.

Algunos resultados interesantes

◇ p-Clique es W[1]-completo bajo m-reducciones parametrizadas.

Ya demostramos que p-Clique$\leq_m^{fp}$p-MC(CQ) y, dado que se puede demostrar que p-MC(CQ[binary])$\equiv_m^{fp}$p-MC(CQ), bastaría con demostrar que p-MC(CQ[binary])$\leq_m^{fp}$p-Clique.

◇ *p-Conjunto Dominante* es W[2]-completo bajo m-reducciones parametrizadas.
(Corresponde a encontrar un conjunto de nodos, el cual unido con sus vecinos, forme el grafo completo. Los vecinos de un nodo son todos aquellos nodos que están conectados a éste con una arista).

Algunos resultados interesantes(cont)

◇ Para todo $d \geq 0$ tenemos que $WSAT(\Delta_{1,d}) \in FPT$.

◇ Para todo $t \geq 1$ y $d \geq 0$ se tiene que $WSAT(\Delta_{t+1,d} \in W[t])$.

Esta clase corresponde a una jerarquización (o ramificación, si se prefiere) del problema SAT (que se encuentra en NP).

Pero ésta no es la única clase de complejidad parametrizada que nos interesa.

QBF

- Tomemos el problema del la fórmula booleana cuantificada (QBF).

  Una fórmula booleana cuantificada es de la forma $Q_1 x_1 \ldots Q_n x_n \theta$, donde Q_i es un cuantifocador existencial o universal, y θ es una fórmula proposicional en las variables x_i.

- Este problema es PSPACE-completo.

Parametrizar este problema no es fácil...

Parametrización de QBF

Supongamos que podemos considerar j "conjuntos" de cuantificadores (y variables) de modo de generar una fórmula η de la forma:

$$\begin{array}{llllllll} Q_1 x_1^1 & \ldots & Q1 x_{n_1}^1 & & & & & \\ & Q_2 x_1^2 & & \ldots & Q_2 x_{n_2}^2 & & & \\ & & & \ddots & & & & \\ & & & & Q_j x_1^j & \ldots & Q_j x_{n_j}^j & \theta \end{array}$$

Parametrización de QBF (2)

Supongamos que $Q_1 = \exists$, $Q_2 = \forall$, etc.

Es decir, que la fórmula η pertence a la clase Σ_j.

Una fórmula es $(k_1 \ldots k_n)$ satisfacible si existe asignación S_1 de peso k_1 para las variables $x_1^1, \ldots x_{n_1}^1$, tal que para toda asignación S_2 de peso k_2 para las variables $x_1^2, \ldots x_{n_2}^2$ tal que existe ...

... tal que para la asignación $S_1 \cup S_2 \cup \ldots \cup S_j$ satisface θ.

Versión Parametrizada de QBF

Sea Θ una clase de fórmulas proposicionales:

p-QBF(Θ)

Entrada:	Fórmula η de la forma anterior, donde $\theta \in \Theta$
Parámetro:	$(j, k_1, \ldots, k_j) \in \mathbb{N}^*$
Problema:	Decidir si η es $(k_1, \ldots, k_j)$-satisfacible.

A partir de este problema se define la jerarquía AW.

La Jerarquía AW

Para cada $t \geq 1$ se tiene que la clase $AW[t]$ corresponde a los problemas tales que:

- ◇ Son m-reducibles a $p - QBF(\Gamma_{t,d})$.
- ◇ (Recordar que $\Gamma_{t,d}$ corresponde la clase de fórmulas que son conjunciones *grandes* de $\Delta t - 1$, tal que la profundidad de estas últimas es a lo más d.)
- ◇ Se considera la unión para toda profundidad de las subfórmulas.

Más formalmente se puede escribir:

$$\boxed{AW[t] := \bigcup_{d \geq 0} [p - QBF(\Gamma_{t,d})]_m^{fp}}$$

La unión de todos los AW[t] se denomina $AW[*]$.

Resultados sobre la Jerarquía AW

Teorema: La Jerarquía AW colapsa a su primer nivel, es decir,

$$AW[*] = AW[1]$$

El hecho que la jerarquía AW colapse a su primer nivel, indica que las versiones cuantificadas de QBF, para fórmulas con $\vee$ y $\wedge$, tienen la misma complejidad (Recordar que AW se define para $p - QBF(\Gamma_{t,d})$).

Es un resultado esperable, al ser una versión paramétrica de problemas PSPACE-completos.

Teorema: El problema parametrizado del chequeo de modelo para fórmulas de Primer Orden es AW[*]-completo bajo reducciones paramétricas, es decir,

$$[p - MC[FO]]_m^{fp} = AW[*]$$

¿Sucede lo mismo con la clase PROP de **todas** las fórmulas proposicionales?

Comentarios a los resultados anteriores

Hasta ahora, se conocen las siguientes relaciones:

⋆ La unión de las clases $W[t]$ es un subconjunto de los problemas que se pueden reducir a WSAT para la clase de fórmulas proposicionales.

◇ WSAT para fórmulas proposicionales es subconjunto de $AW[*]$.
En el sentido clásico, es equivalente a decir que $NP \subseteq PSPACE$.

⋆ $AW[*]$ es subconjunto de los problemas reducibles a $p - QBF(PROP)$.

Hasta ahora no se ha podido probar que las inclusiones sean estrictas...

Chequeo de Modelos y la Jerarquía A

Hasta ahora, el vocabulario no se consideraba como parte de la complejidad del problema.

No se prestaba atención si era un parámetro del problema o si estaba fijado de antemano.

Ahora, definiremos por $PMC(\Phi[*])$ la unión (para todos los vocabularios) de los problemas reducibles a $p-MC(\Phi[\mathfrak{L}])$, donde $\Phi(\mathfrak{L})$ es la clase de fórmulas Φ, para el vocabulario $\mathfrak{L}$.

Formalmente hablando:

$$PMC(\Phi[*]) := \bigcup_{\mathfrak{L}} [p-MC(\Phi[\mathfrak{L}])]_m^{fp}$$

La Jerarquía A

Se define la jerarquía A para $t \geq 1$:

$$A[t] := PMC(\Sigma_t[*])$$

Lema: Para cada $t \geq 1$ y para cada vocabulario $\mathfrak{L}$ que sea al menos binario, se tiene:

$$PMC(\Sigma_t[*]) \subseteq PMC(\Sigma_{t+1}[*])$$

Nuevamente no se sabe si las inclusiones son estrictas.

Esto nos permite dar una nueva caracterización de W[1]:

$$PMC(\Sigma_1[*]) = W[1]$$

Comentarios sobre La Jerarquía A

- Se tiene que $W[t] \subseteq A[t]$.

- Además, $\bigcup_{t \geq 1} A[t] \subseteq AW[*]$.

- Se puede ver a la Jerarquía A como la versión parametrizada de la Jerarquía polinomial, mientras que la Jerarquía W se puede ver como una ramificación de la clase NP.

- A pesar de lo anterior no hay evidencia suficiente para demostrar o refutar que $W[2] = A[2]$ (Así como no se ha podido demostrar que $NP \subsetneq \Sigma_2^p$).

Ultimos comentarios

- Se puden construir versiones parametrizadas de una interesante gama de problemas, permitiendo estudiar clases de complejidad equivalentes para estos problemas:
  - ⇒ La Jerarquía W corresponde a una ramificación, una clasificación más detallada de los problemas en NP.
  - ⇒ La Jerarquía AW corresponde a la clase PSPACE y dicha jerarquía colapsa sobre su primer nivel.
  - ⇒ La jerarquía A corresponde a la jerarquía polinomial tradicional.
- El problema del chequeo de modelos (consultas a bases de datos) resulta ser completo, en distintas versiones para distintas clases.
- Se conocen relaciones de inclusión entre las clases de problemas en W[t] y A[t] y A[*] y AW[*], es decir, se dan todas las inclusiones esperadas, pero no se han logrado descubrir otras más que aporten a las dudas tradicionales.

¿Preguntas?