

Consultas a Bases de Datos expresadas con extensiones de FO

Gagliardi, Edilma Olinda - Herrera, Norma Edith - Reyes, Nora Susana
{oli, nherrera, nreyes}@unsl.edu.ar
Proyecto de Investigación "Teoría de la Computación"
dirigido por el Dr. José María Turull Torres
Dpto. de Informática, Universidad Nacional de San Luis

Dado que el poder expresivo de una Lógica de Primer Orden (FO) está restringido a una subclase de la clase de Queries Computables (CQ), estamos utilizando una lógica extendida con distintos tipos de cuantificadores de Clausura Transitiva y Punto Fijo para la expresión y evaluación de consultas (queries) a Bases de Datos Relacionales, y que caen fuera de lo expresable en FO y estrictamente incluido en la clase CQ.

Por medio de una Base de Datos obtenemos una representación simbólica de una realidad acotada. En nuestro trabajo, para tal representación utilizamos el Modelo Relacional, puesto que tanto en el campo de la Teoría de Base de Datos como en el de las aplicaciones, nos brinda un marco formal adecuado de estudio.

Como es deseable obtener información de la Base de Datos, es necesario contar con un lenguaje de consultas, estas últimas conocidas habitualmente como *queries*. En la formulación de Bases de Datos en términos de la teoría de conjuntos, Modelo Relacional, consideramos los lenguajes relacionales de consultas, tales como el álgebra o el cálculo, los cuales son equivalentes en su poder expresivo a la lógica de Primer Orden (FO).

Aho y Ullman [AU79] demostraron que los denominados queries de clausura no son expresables en los lenguajes relacionales mencionados. Entonces cabe preguntarse cuál es la clase de todos los queries posibles a una Base de Datos, y más aún, si dicha clase puede caracterizarse, cuáles son los lenguajes que realmente computan dicha clase. Esta segunda pregunta dio lugar al nuevo concepto de completitud relacional [CH80], dejando obsoleto el referido por Codd al definir los lenguajes relacionales.

Para ello observamos el Modelo Relacional en el marco de la Teoría de Modelos Finitos [EF95][AHV95], tal que el esquema de la Base de Datos Relacional está definido por un vocabulario relacional finito σ , y la instancia de Base de Datos es una σ -estructura, donde cada relación es de la aridad correspondiente para cada símbolo de σ , definida en un dominio finito dado. Así se definen Clases de Estructuras Relacionales Finitas para Modelos Relacionales, y se definen sus instancias como *Estructuras Relacionales Finitas*. Entonces, desde este punto de vista, vemos un query como una función cuyo dominio es el conjunto de las estructuras relacionales finitas de un cierto vocabulario, y cuyo codominio es el conjunto de las relaciones definidas en el dominio de la estructura relacional finita correspondiente para alguna aridad. En [CH80] se propone como definición de la *clase de queries computables CQ* a la clase de funciones definidas en las clases de estructuras relacionales finitas, para cada vocabulario relacional finito, tales que son funciones recursivas parciales en alguna codificación lineal de las estructuras, y preservan isomorfismos en ellas.

Aumentar FO con cuantificadores incrementa su poder expresivo [Ebi85], permitiendo resolver más queries de la clase CQ, aunque no se alcanza a cubrir la clase completa. Algunos autores los tratan como cuantificadores y otros como operadores de trabajo; nosotros utilizamos la primera terminología dado que nos movemos en el marco de las lógicas.

La motivación de este trabajo es que, como ya se mencionó, el grado de expresividad de FO es sumamente limitado ya que en [CH80] se demostró que está estrictamente contenido en (y es muy inferior a) $DSPACE(\log n)$. Nosotros consideraremos extensiones de FO con el agregado de cuantificadores que nos permitan incorporar distintos tipos de iteraciones en su interpretación semántica, con el fin de aumentar el poder expresivo de FO, aunque no alcancemos a cubrir totalmente la clase CQ.

Se está utilizando un lenguaje de consultas a Bases de Datos Relacionales que computa queries expresables en FO extendido con los cuantificadores Clausura Transitiva(CT), Clausura Transitiva Determinística (CTD)[GMc95], Punto Fijo Determinístico Inflacionario (PFDI), Punto Fijo Determinístico No Inflacionario (PFDN), Punto Fijo No Determinístico Inflacionario (PFNI), Punto Fijo No Determinístico No Inflacionario (PFNN), Punto Fijo Alternado Inflacionario (PFAI), Punto Fijo Alternado No Inflacionario (PFAN)[AVV97].

Al extender la lógica de Primer Orden, con los distintos tipos de cuantificadores, se consigue capturar en cada caso las clases que se mencionan sobre estructuras finitas ordenadas: FO(CTD) captura LOGSPACE, FO(CT) captura NLOGSPACE, FO(PFDI) captura PTIME, FO(PFDN) captura PSPACE, FO(PFNI) captura NPTIME, FO(PFNN) captura PSPACE, FO(PFAI) captura PSPACE, FO(PFAN) captura EXPTIME [GMc95][Imm87][Var82] [AVV97] [GS86]. Por lo tanto, al trabajar con todas estas extensiones de Primer Orden se pueden analizar consultas a Bases de Datos que están en la clase EXPTIME.

Esto es relevante de destacar porque considerando la clase CQ, en general los lenguajes de consultas que se utilizan en la práctica no toman en cuenta el concepto de computabilidad y son incompletos respecto del mismo, o calculan queries no computables en su defecto. Obsérvese que estos problemas de expresividad y computabilidad no son sólo de interés teórico, sino que los mismos pueden causar serios errores en aplicaciones de cierta complejidad. Así podemos hacer analogías con otras aplicaciones semejantes, en donde los lenguajes de consulta disponibles a las Bases de Datos resultan pobres en su expresividad con respecto a la clase CQ, y frente a ciertos queries, no es posible su expresión dado que no cuentan con la capacidad de los cuantificadores analizados.

En virtud de lo expuesto, la formulación de query es una abstracción precisa y rigurosa del tipo de consulta que un programa de computadora debería computar. Se ha demostrado una parte de estos resultados, exponiendo simplemente como con FO no es suficiente para la expresión de queries de cierta categoría. Así podemos observar que sobre estructuras finitas ordenadas, con las extensiones de tales cuantificadores, capturaríamos la clase EXPTIME, mientras que con FO quedamos muy lejos del poder expresivo conseguido con las mismas.

Cabe destacar también que estamos realizando un arduo análisis en determinar algoritmos, cotas y comportamientos de estos cuantificadores, puesto que las consultas bibliográficas y/o por correspondencia electrónica a los autores de referencia, no nos aportaron soluciones inmediatas y claras. En particular, podemos referirnos a los cuantificadores de punto fijo, ya que sobre ellos hay muy poco que pueda encontrarse en la bibliografía respecto del comportamiento de los mismos en aplicaciones computacionales.

Se está analizando también la posibilidad de realizar una optimización de las consultas antes de ser evaluadas con el fin de realizar una evaluación eficiente de las mismas. Un ejemplo del tipo de optimización que se podría realizar es que al calcular el resultado de la aplicación de cualquiera de los cuantificadores de este tipo, si las únicas variables que aparecen en sus fórmulas son las que tiene como parámetro el cuantificador, se podría realizar la evaluación de las mismas en una primera pasada sobre la consulta y luego sustituir toda su expresión por el nombre de una nueva relación que mantenga su resultado como si fuera una relación más de la Base de Datos.

Además se prevé que a posteriori se puedan incluir otros tipos de cuantificadores nuevos o incluir lógicas de orden superior a nuestro lenguaje, lo que sería simple de realizar si se trabaja modularmente al nivel de programación.

Hemos podido observar que en esta temática hay muy poco material de aplicación y por consiguiente hay muchos cuestionamientos no planteados respecto del comportamiento real de los cuantificadores. Por ejemplo, en el caso de los puntos fijos determinar cotas más adecuadas haciendo uso de algo más de información respecto de las secuencias que se generan. O determinar si, por ejemplo hay algún cuantificador de punto fijo alternado que permita acercarse más al comportamiento de una Máquina de Turing Alternada, porque el introducido en [AVV97] no permite la alternación en todos los niveles del árbol de etapas.

Como parte de nuestro trabajo, se está desarrollando un intérprete de consultas a Bases de Datos Relacionales expresadas utilizando fórmulas de FO extendida con los distintos tipos de cuantificadores ya mencionados y que están ausentes en FO.

Este trabajo está vinculado a otros trabajos y con él se está contribuyendo también a la construcción de una biblioteca de programas que implementen distintos tipos de formalismos, con diferentes grados de expresividad para formular consultas a Bases de Datos Relacionales.

Bibliografía

- [AHV95] Abiteboul,S; Hull y Vianu, V.; “*Foundations of Databases*”. Addison-Wesley Publishing Company, 1995.
- [AU79] Aho, A.V. y Ullman, J.D.; “*Universality of Data Retrieval Languages*”; Proc. 6th.ACM Symp. on Principles of Programming Languages, 110-117, 1979.
- [AV95] Abiteboul, S.; Vianu, V.; “*Computing with First Order Logic*”; Journal of Computer and System Sciences, 50:309-335, 1995.
- [AVV97] Abiteboul, S.; Vardi, M. y Vianu, V.; “*Fixpoint Logics, Relational Machines, and Computational Complexity*”, Journal of ACM, 44(1), 30-56.Enero, 1997. Versión Completa Obtenida desde la dirección: http://www_cse.ucsd.edu/users/vianu
- [CH80] Chandra, A.K.; Harel, D.; “*Computable Queries for Relational Data Bases*”. Journal of Computer and System Sciences 21, 156-178. 1980.
- [Ebi85] Ebbinghaus, H; Flum, J.;Thomas, W.; “*Extended Logic: the general framework, in perspectives in mathematical logic*”, Springer-Verlag, 1985.
- [EF95] Ebbinghaus, H; Flum, J.; “*Finite Model Theory*”, Springer-Verlag, 1995.
- [GMc95] Grädel, E. y McColm, G. L.; “*On the Power of Deterministic Closures*” Versión Final, Information and Computation, Vol. 119, 129-135, 1995”
- [GS86] Gurevich, S.y Shelah, S.; “*Fixed Point Extensions of First Order Logic*” Annals of Pure and Applied Logic; 32:265-280, 1986.
- [Imm87] Immerman, Neil; “*Languages that Capture Complexity Classes*”; SIAM Journal of Computing 16:4, 1987, 760-778.
- [Tur96] Turull Torres, José M.; “*Clases de Bases de Datos L-Rígidas y Expresividad de Lenguajes Relacionales Incompletos*”. Tesis Doctoral, UNSL, 1996.
- [Var82] Vardi, M. Y.; “*The Complexity of Relational Queries Languages*”Proc. 14th ACM Symp. On Theory of Computing, 137-146, 1982.