Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Datalog +/-, una interfaz tolerante a la inconsistencia

Tesis de Licenciatura en Ciencias de la Computación

Pablo Víctor Fromer

Director: Vanina Martinez
Codirector: Ricardo Rodriguez
Buenos Aires, 2019

Índice general

1. INTRODUCCIÓN

1.1. Motivación Opcion 1

Los lenguajes ontológicos, los sistemas basados en reglas y sus integraciones han jugado un rol central en el desarrollo de la Web Semántica... Antes del surgimiento de Datalog +/- no existía literatura que explique cómo generalizar reglas y dependencias de bases de datos de manera que puedan expresar axiomas ontológicos. De esta manera no estaba cubierto el interés en la comunidad de la Web Semántica por conseguir formalismos altamente escalables que pudieran hacer que la red se beneficie de las tecnologías y las optimizaciones ya implementadas en los motores de bases de datos.

Datalog +/- es una serie de variantes del lenguaje Datalog que son particularmente adecuadas para responder consultas sobre ontologías de manera eficiente. Estas variantes extienden Datalog con la posibilidad de cuantificar existencialmente a las variables en las implicaciones de las reglas y otra serie de ventajas, mientras que al mismo tiempo restringen el lenguaje de manera sintáctica con el objetivo de asegurar tratabilidad.

Por otro lado, tanto en el campo de las bases de datos como en el de la Web Semántica es ampliamente reconocido que la inconsistencia es un problema que no puede ser ignorado. Al momento de integrar información proveniente de diversas fuentes, ya sea para popular una ontología o para responder una consulta, las restricciones de integridad son muy propensas a ser violadas en la práctica. En este trabajo proponemos una implementación de la semántica IAR, la cual describe un método para tratar ....

Finalmente presentamos una herramienta que tiene el propósito de acercar a la comunidad una implementación de Datalog+/-, con el objetivo de poder acceder a este lenguaje de manera fácil, a través de Internet...

1.2. Datalog +/-

1.2.1. Consultas y bases de datos

Con respecto a los ingredientes elementales, se asumen constantes, nulos y variables de la siuguiente manera; estos son los arguentos de las fórmulas atómicas en las bases de datos, consultas y dependencias. Se asume (i) un universo infinito de constantes Δ (las cuáles forman el dominio de la base de datos), (ii) un conjunto infinito de nulos etiquetados Δ_N (representando valores desconocidos), y (iii) un conjunto infinito de variables X (que se utilizan en las consultas y en las dependencias). Cada constante representa un valor diferente, mientras que nulos distintos podrían representar el mismo valor. Se asume un orden lexicográfico en $\Delta \cup \Delta_N$, donde los símbolos de Δ_N siguen a los de Δ. Se denota por X a una sequencia de variables X_1, ..., X_K con $k > 0$.

Se definen las fórmulas atómicas, que ocurren en bases de datos, consultas, y dependencias, y que se construyen en base a los nombres de las relaciones y los términos. Se asume un *esquema relacional* R, el cual constituye un conjunto finito de *nombres de relación*, o *símbolos de predicado*. La posición *P[i]* identifica al i-esimo argumento de un predicado P. Un *término* t es una constante, un nulo o variable. Una *fórmula atómica* (o *átomo*) a tiene la forma $P(t_1,...,t_n)$, donde P es un predicado n-ario, y $t_1,...,t_n$ son términos. Se denotan

por *pred*(a) y *dom*(a) su predicado y al conjunto de sus arguemntos, respectivamente. Esta notación se extiende de manera natural para conjuntos y conjunciones de átomos.

Podemos definir ahora la noción de base de datos relativa a un esquema relacional, junto con la sintaxis y la semántica de las consultas conjuntivas y las consultas conjuntivas booleanas a una base de datos. Una *instancia de base de datos* D para un esquema relacional R es un conjunto de átomos (posiblemente infinito) con predicados de R y argumentos de Δ. Una consulta conjuntiva sobre R tiene la forma $Q(\mathbf{X}) = \exists \mathbf{Y} \Phi(\mathbf{X}, \mathbf{Y})$, donde $\Phi(\mathbf{X}, \mathbf{Y})$ es una conjunción de átomos con las variables $\mathbf{X}$ y $\mathbf{Y}$, y eventualmente constantes, pero sin nulos. Una consulta conjuntiva *Booleana* sobre R es una consulta conjuntiva de la forma $Q() = \exists \mathbf{Y} \Phi(\mathbf{X}, \mathbf{Y})$, donde todas las variables están cuantificadas existencialmente. El conjunto de todas las respuestas a una consulta conjuntiva $Q(\mathbf{X}) = \exists \mathbf{Y} \Phi(\mathbf{X}, \mathbf{Y})$ sobre una base de datos es el conjunto de todas las tuplas $\mathbf{t}$ sobre Δ para las cuales existe un homomorfismo $\mu : \mathbf{X} \cup \mathbf{X} \to \Delta \cup \Delta_N$ tales que $\mu(\Phi(\mathbf{X}, \mathbf{Y})) \subseteq D$ y $\mu(\mathbf{X}) = D$

1.2.2. Dependencias generadoras de tuplas (TGDs)

Las dependencias generadoras de tuplas (TGDs) son restricciones sobre una base de datos en la forma general de las reglas de Datalog..... Dado un schema relacional R, una TGD σ es una fórmula de primer orden de la forma $\forall X \forall Y \Phi(X, Y) \to \exists Z \Psi(X, Z)$, donde $\Psi(X, Z)$ y $\Phi(X, Y)$ son conjunciones de átomos sobre R, llamadas el *cuerpo* y la *cabeza* de σ, denotadas *cuerpo*(σ) y *cabeza*(σ) respectivamente. Tal σ es satisfecha en una base de datos D para R ssi, toda vez que exista un homomorfismo h que mapea los atomos de $\Phi(X, Y)$ a atomos de D, existe una extensión $h\prime$ de h que mapea los atomos de $\Psi(X, Z)$ a átomos de D. Todos los conjuntos de $TGDs$ son finitos.

1.2.3. Respondiendo consultas bajo TGDs

La evaluación de una consulta conjuntiva o una consulta conjuntiva booleana sobre una base de datos bajo un conjunto de TGDs se define como se describre a continuación. Dada una base de datos D para R, y un conjunto de TGDs Σ sobre R, el conjunto de modelos de D y Σ, el cual denotamos por $mods(D, \Sigma)$, es el conjunto (posiblemente infinito) de todas las bases de datos B tales que (i) $D \subseteq B$ y (ii) toda $\sigma \in \Sigma$ es satisfecha en B. El conjunto de todas las respuestas para una consulta conjuntiva Q, denotado por $ans(Q, D, \Sigma)$, es el conjunto de todas las tuplas $\mathbf{a}$ tales que $\mathbf{a} \in Q(B)$ para todo $B \in mods(D, \Sigma)$. La respuesta para una consulta conjuntiva booleana Q a D es *Sí*, denotada por $D \cup \Sigma \models Q$, si solo si, $ans(Q, D, \Sigma) \neq \emptyset$. Notar que responser consultas bajo TGDs para el caso general es indecidible [1], aún cuando el esquema y las TGDs son fijas [2]. (Agregar referencias a esto)

1.2.4. El chase

El *chase* es un procedimiento para reparar una base de datos con respecto a un conjunto de dependencias, de manera tal que el resultado del chase satisfaga esas dependencias. Por *chase* nos referimos tanto al procedimiento como a su resultado. El TGD chase trabaja sobre una base de datos a través de aplicar las TGDs como se explica a continuación. Considere una base de datos D para un esquema relacional R, y una TGD σ sobre R de la forma $\Phi(X, Y) \to \exists Z \Psi(X, Z)$. Entonces, σ es *aplicable* a D si existe un homomorfismo h que mapee los átomos de $\Phi(X, Y)$ a átomos de de D. Sea σ aplicable a D, y sea h_1 un

homomorfismo que extiende h de la siguiente manera: para cada $X_i \in \mathbf{X}, h_1(X_i) = h(X_i)$; para cada $Z_j \in \mathbf{Z}, h_1(Z_j) = z_j$, siendo z_j un nulo "fresco", es decir, $z_j \in \Delta_N, z_j$ no ocurre en D, y z_j sigue lexicográficamente todos los otros nulos introducidos previamente. Al aplicar σ a D, se agrega a D el átomo $h_1(\Psi(X, Z))$, si no se encuntra en D previamente. El algoritmo del chase para una base de datos D y un conjunto de TGDs Σ consiste en una aplicación exhaustiva de la regla anterior en anchura (como traduzco breadh-first fashion?), lo cual trae como resultado un chase (posiblemente infinito) para D y Σ. Formalmente, el chase de nivel 0 para D y Σ, denotado por $chase^0(D, \Sigma)$, se define como D, asignándole a cada átomo de D el nivel de derivación 0. Paara cada $k \geq 1$, el chase de nivel k de D y Σ, denotado $chase^k(D, \Sigma)$, se construye de la siguiente manera: sean $I_1, ..., I_n$ todas las posibles imágenes de los cuerpos de las TGDs en Σ relativas a algún homomorfismo tal que (i) $I_1, ..., I_n \subseteq chase^{k-1}(D, \Sigma)$ y (ii) el nivel más alto de todo átomo en cada I_i es $k - 1$; entonces, se aplica cada posible TGD sobre $chase^{k-1}(D, \Sigma)$, escogiendo las TGDs y los homomorfismos en un orden lineal y lexicográfico respectivamente, y asignándole a cada átomo nuevo el nivel de derivación k. El chase de D relativo a Σ, denotado $chase(D, \Sigma)$, se define entonces como el limite de $chase^k(D, \Sigma)$ para $K \to \infty$.

El chase (posiblemente infinito) es un modelo universal, es decir que existe un homomorfismo del $chase(D, \Sigma)$ a cada $B \in mods(D, \Sigma)$. Este resultado implica que las consultas conjuntivas boolenas sobre D y Σ pueden ser evaluadas en el chase para D y Σ, es decir que $D \cup \Sigma \models Q$ es equivalente a $chase(D, \Sigma) \models Q$.

2. RESULTADOS FORMALES

3. IMPLEMENTACIÓN

Bibliografía

[1] C. Beeri y M. Y. Vardi. *The implication problem for data dependencies*. En *Proc. ICALP-1981*, pp. 73–85, 1981.

[2] A. Calı, G. Gottlob, y M. Kifer. Taming the infinite chase: Query answering under expressive relational constraints. En *Proc. KR-2008, pp. 70–80, 2008.*