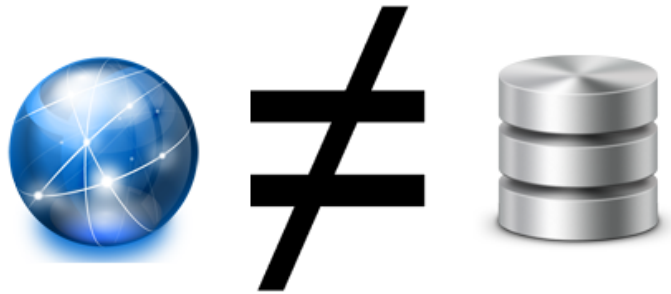
CC71X - La Web de Datos
Formal Models of Web Queries

Felipe Bravo Márquez

2 de noviembre de 2010

Introducción

- Desde la aparición de la Web, se han desarrollado diversas herramientas y técnicas para recuperar información en ella.
- Muchas de éstos trabajos se han basado en la metáfora de tratar a la Web como una **base de datos**.
- Principalmente para poder adaptar lenguajes de consulta establecidos como **SQL** o **Datalog** a la Web.
- La Web **no es una base de datos** y consultarla es diferente a consultar a una base de datos convencional.
- Las mayores diferencias radican en la **falta de control de concurrencia** y las **capacidades limitadas de acceso**.



Falta de control de Concurrencia en la Web

- La falta de control de concurrencia impide que las consultas puedan ser computadas de manera eficiente.

- El dueño de un documento puede realizarle un **lock** para prevenir que otros lo accedan mientras lo modifica.
- Ningún otro usuario puede realizarlo y **no existen mecanismos de transacciones**.
- Muchas consultas son imposibles de computar en tiempo finito.
- Por ejemplo, preguntar por los documentos alcanzables directa o indirectamente a partir de un punto de partida requeriría **navegar por una cadena de documentos** en la cual se van agregando nuevos documentos más rápidamente a como se descubren.
- La computación de la consulta nunca terminaría.

Capacidades limitadas de acceso

- Asumamos una Web capaz de crear vistas estáticas de ésta en el procesamiento de consultas.
- Aún no contamos con una base de datos tradicional puesto que el acceso a documentos es netamente **navegacional**.
- Sólo se puede acceder a un documento ya sea sabiendo su *URL* o a través de otro documento que lo apunte.

¿Cómo lo hacen los motores de búsqueda?

- Por medio de la navegación de sitios crean índices (invertidos) sobre el contenido de los documentos.
- Las búsquedas se procesan en el índice construido por medio de una navegaciones off-line.
- Estos índices no proveen una recuperación asociativa real de la Web, pues no garantizan tener indexada la Web completa.
- Básicamente, los motores de búsqueda son formularios que proveen acceso a datos pre-computados.

Limitaciones de la Web

Ejemplo de consulta imposible de evaluar

- Consultas del tipo " existen documentos apuntando al documento d " son **imposibles** de evaluar.
- Incluso en una **Web estática** no hay forma de examinar todos los documentos y asegurar no haber perdido alguno.

Otras limitaciones de la Web

- La Web es heterogénea, autónoma y carente de estructura
- Estas limitaciones no son consideradas en este estudio.

A continuación se plantea un modelo para tratar a la Web como una **base de datos finita**, asumiéndola como **estática** para evitar considerar los problemas derivados a la falta de control de concurrencia.

Definiciones Preliminares

- La Web es una colección de documentos heterogéneos y distribuidos que se conectan por medio de **hipervínculos** (links).
- La Web es un grafo cuyos nodos son *objetos Web* identificados por un *Uniform Resource Locator* y posee cierto contenido arbitrario dependiente del tipo (HTML, Postcript, image, ..etc)
- Se asocia un objeto Web con una tupla en una relación virtual de Nodo

$$N[id, title, content, type, length, modif, \ldots]$$

EL id representa una *URL* y es la llave de la relación, el resto de los atributos están generalmente presentes en un documento HTML pero pueden ser nulos.

- Un link se representa como una tupla de relación de Link

$$L[source, destination, d\_offset, \ldots]$$

donde $source$ y $destination$ son Oid's (Object ids).

Definiciones Preliminares [2]

Formularios

- Los formularios son documentos Web especiales que permiten a los usuarios completar ciertos parámetros y obtener **otro documento** como resultado.
- Son documentos cuyos out-links son parametrizados.
- Un **link parametrizado** L_p es una relación con la misma estructura que una relación de links L más un atributo adicional $param\_set$.

$$P[param\_set : t_1, param\_name : t_2, param\_val : t_3]$$

Relaciones bases y Acceso a datos

- Existen datos adicionales usados al interactuar con la Web como bookmarks, archivos locales ,..etc.
- Pueden ser usados para **inicializar un acceso a la Web** como por ejemplo iniciar la navegación siguiendo algunos bookmarks definidos.
- Estos datos pueden ser vistos como un conjunto de **relaciones base** que contienen *información adicional relevante*.
- Las relaciones base son completamente accesibles mediante un lenguaje de consulta relacional
- Mientras que las relaciones N, L, L_P y P sólo pueden ser accedidas ya sea especificando explícitamente una URL o siguiendo algún link.
- Los **índices invertidos** de los motores de búsqueda no reflejan el estado de la Web al momento de una consulta, o un estado completo y consistente de la Web para algún momento.

Finitud de la Web

- En el estudio, a diferencia de [Abiteboul and Vianu, 1997] se considera a la Web como finita por muy grande que sea.
- Documentos HTML (sin considerar formularios) poseen una cantidad finita de out-links.

Formularios y Finitud

- Se pueden modelar la cantidad de combinaciones de parámetros de un formulario como finita.
- Si bien, existen formularios que aceptan entradas arbitrarias y pueden generar infinitos documentos, éstos se ignoran en el trabajo.

Se propone a continuación un modelo de datos formal donde se **omiten** en un principio tanto los **formularios** como los links **parametrizados**.

Web database Schema

- Un **esquema de base de datos Web** W es un esquema de base de datos relacionales con un conjunto finito de **relaciones base** $DB = \{R_1, \ldots, R_n\}$ y dos esquemas relacionales adicionales:
  1. Un esquema N que contiene los objetos $node$
  2. Un esquema L que contiene los objetos $link$.

Web Database

Web database

- Llamamos **Web database** W a una instancia de esquema de base de datos Web donde:
  1. W mapea un conjunto finito de tuplas con cada relación $R_i \in DB$
  2. Posee un conjunto finito de objetos $node$ con N
  3. Posee un conjunto finito de $links$ con L y existe una restricción referencial entre el atributo $source$ de L y el atributo id de N donde
  4. la restricción referencial no aplica para el atributo $destination$ pues un documento Web puede apuntar a documentos(URLs) no existentes.

Web Machine

- Para bases de datos relacionales tradicionales el usuario tiene control total sobre sus relaciones
- La computación de las consultas se puede abstraer mediante una máquina de Turing tradicional.
- En la Web, el acceso a los datos se ve limitado por la navegación.
- Esta limitación se formaliza con las *Web Machines*

Web Machine[2]

Web Machine

- Una Web Machine es una máquina de Turing aumentada con un **oráculo**.
- Tiene dos cintas de entrada
  1. Una cita ordinaria donde las entradas de relaciones base DB están codificadas,
  2. Una cinta de oráculo donde las relaciones N y L están codificadas.
- Tiene dos cintas de trabajo:
  1. Una cinta de trabajo ordinaria
  2. Una cinta de trabajo del oráculo
- Finalmente tiene una cinta de salida donde se escribe el resultado de la computación.

Web Machine [3]

En cada etapa de la computación una Web machine puede hacer una de las siguientes operaciones:

1. Leer desde la cinta de entrada ordinaria o de alguna de las cintas de trabajo (no puede leer desde la cinta de entrada del oráculo)
2. Escribir en la cinta de salida o en alguna de las cintas de trabajo
3. Llamar al oráculo

Web Machine [4]

Cuando la Web machine llama al Oráculo

- El oráculo lee el contenido de su cinta de trabajo e interpreta su contenido como un *node id*
- Si existe el *node id* en su cinta de entrada, copia a su cinta de trabajo la tupla de *node* relevante (codificada) y las tuplas de todos los nodos de sus links de salida.
- Si no existe el objeto *node* en su cinta de entrada, el oráculo escribe el símbolo $\perp$ en la cinta.

Web Machine [5]

- La máquina abstrae la idea de que los *nodos* sólo pueden ser accedidos a partir de su id.
- Para navegar a partir de una *URL* dada, se **llama** primero al oráculo por aquella *URL*
- Luego, éste retorna entre otras cosas, los ids de los **documentos apuntados** por el documento dado.
- Posteriormente, puede ser llamado **nuevamente** para acceder a los datos de aquellos ids apuntados, y así sucesivamente.

Total relational Queries

- Una **consulta relacional total** es un mapeo genérico Q de **instancias** pertenecientes a un esquema de base de datos a instancias de otro esquema.
- Donde existe una **máquina de Turing** tal que, dada una instancia I del primer esquema y cualquier codificación $enc(I)$ en su **cinta de entrada**, la máquina se detiene con $enc(Q(I))$ en su **cinta de salida**.

Web Query

- Se define a una **consulta Web** como una **consulta relacional** Q que mapea instancias de una **base de datos Web** a tuplas de valores sobre las instancias mapeadas.
- Donde existe además, una **Web machine** que computa Q sobre cualquier Web.
- Dada una base de datos Web $W = (DB, N, L)$ y cualquier codificación enc, cuando la Web machine comienza con $enc(DB)$ en su cinta de entrada ordinaria y con $enc(N)$, $enc(L)$ en su cinta de entrada de oráculo, la máquina se **detiene** con la salida $enc(Q(W))$ en su cinta de salida.

Web Query [2]

Ejemplos de consultas Web

- Lista los títulos de los **nodos alcanzables** desde el nodo con $oid\ o$.
- Encuentra todos los nodos con distancia menor o igual a 3 al nodo con $oid\ o$.
- Encuentra todos los **nodos alcanzables** a partir del nodo con $oid\ o$ (computable bajo el supuesto de Web finita)

- Como el acceso a los nodos es navegacional, las siguientes consultas no son computables en un contexto Web:
  1. Encuentra todos los nodos.
  2. Encuentra todos los nodos que referencian al nodo con $oid\ o$.
  3. Encuentra todos los nodos sin links de entrada.

Relación de equivalencia entre Web Queries

- Sean dos instancias W,W' sobre el mismo esquema y S un conjunto de valores
- Se dice que $W =_S W'$ si y solo si
  1. W, W son equivalentes en contenido en todas sus relaciones base
  2. El contenido de las relaciones $Nodo$ y $Link$ para W y W' son **idénticos** cuando se restringe a los nodos con $id \in S$ más los nodos alcanzables directa o indirectamente por aquellos nodos en W o W' junto a los links salientes de los nodos.

Web Queries y Queries relacionales

Theorem

*Una **consulta relacional** Q que mapea una Web database a un conjunto de tuplas sobre los valores en la Web database es una consulta Web **ssi***

- *Q es una consulta Web para todos los pares de entradas $W = (DB, N, L)$ y $W' = (DB, N', L')$ tal que $W =_S W'$, siendo S el conjunto de valores mencionado en DB y dándose que $Q(W) = Q(W')$.*

- Las consultas sólo se interesan en la sub-Web alcanzable usando los datos existentes en las **relaciones base** más los datos encontrados en el camino navegado, los otros nodos son ignorados
- Las consultas se computan recuperando primero todos los documentos alcanzables a partir de las relaciones base, y luego evaluando la consulta sobre éstos.

Queries Eventualmente Computables

- Consultas relaciones del tipo “lista todos los documentos que referencian a o” no son Web Queries
- No esta garantizado que la computación termine
- Sin embargo, es posible encontrar tuplas en la salida, a pesar de que nunca sabremos si estamos listos
- Para el caso del ejemplo se podrían enumerar todas las posibles URL’s, luego para cada uno testear si corresponde a un nodo existente y luego si el nodo apunta a o
- Llamamos a estas consultas como **eventualmente computables**

Queries Eventualmente Computables [2]

Una consulta Web es eventualmente computable si una Web machine puede producir eventualmente una tupla en la salida.

Theorem

Una consulta Web es eventualmente computable ssi es monótona con respecto a la adición de nodos inalcanzables a la Web.

- Las consultas *"lista todas los documentos referenciando o"* o *"lista los títulos de todos los artículos"* son eventualmente computables
- La consulta *"encuentra todos los nodos que nunca son apuntados"* no es eventualmente computable, puesto que no hay manera de asegurar que se revisaron todos los nodos y por lo tanto que el nodo no tenga links que lo referencien.

Web Calculus

- Web calculus es una extensión y una abstracción del lenguaje de consulta *WebSQL* descrito en [Mendelzon et al., 1996]
- *WebSQL* integra la recuperación de información basada en contenidos (como los motores de búsqueda) con recuperación estructurada y topológica.

Ejemplo

Recuperar documentos que contengan el string “database” que pueden ser alcanzados por un sitio particular por medio de caminos de largo ≤ 2 sin salirse del servidor local.

SELECT	$d.url$, $d.title$
FROM	Document d SUCH THAT
	“www.cs.toronto.edu” $= \mid \Rightarrow \mid \Rightarrow\Rightarrow d$
WHERE	$d.title$ CONTAINS “database”;

La expresión regular $= \mid \Rightarrow \mid \Rightarrow\Rightarrow d$ restringe al camino a empezar en la URL señalada y de tener cero, uno o dos “local” links.

Web calculus [2]

Para una esquema de base de datos Web (DB, N, L), se define al Web calculus como el conjunto de fórmulas de primer orden en el siguiente vocabulario:

- Un símbolo de predicados R_i para cada relación base DB con la misma aridad que la relación
- Símbolos de predicado N y L con la misma aridad que las relaciones correspondientes
- Un predicado ternario $Path(n_1, R, n_2)$ donde existe un camino entre n_1 y n_2 por medio relaciones L
- Un predicado binario de contención del tipo $n\ contains\ s$ donde n es un oid y s un string donde el $body$ del documento n contiene a s.

Web calculus [3]

Es necesario restringir la sintaxis para que sólo consultas **computables** o **eventualmente computables** sean expresables

Ejemplo

La consulta no eventualmente computable **encuentra todos los nodos que nunca son apuntados** se podría representar como

$$\{x | N(x, \dots) \wedge \forall y (N(y, \dots) \rightarrow \neg L(y, x, \dots))\}$$

Se necesitan restricciones sintácticas del Web calculus para asegurar que las consultas definidas por el lenguaje sean computables. A continuación se define el **Web safe calculus**

Web safe calculus

Se requieren dos tipos de restricciones para evitar fórmulas que expresen consultas Web **no-computables**:

1. Es necesario asegurar que el primer argumento para predicados tipo N, L o $Path$ estén asociados a conjuntos de nodos conocidos.
2. Al igual que en el cálculo relacional hay que asegurar que los argumentos de un átomo negado sean instanciados y que los términos de una disjunción usen los mismos conjuntos de variables.

Web safe calculus [2]

- Una fórmula en **safe Web calculus** es una fórmula en **Web calculus** si tiene una de las siguientes formas.
- En la lista siguiente, a es siempre una constante, $x, x_1, \ldots, x_n, y, y_1, \ldots, y_m$ son variables o constantes y $\phi(x_1, \ldots x_n)$ y $\phi'(x_1, \ldots x_n)$ son safe-fórmulas

Safe-fórmulas

- $N(a, x_1, \ldots, x_n), L(a, x_1, \ldots, y_n), Path(a, R, x), R_i(x_1, \ldots, x_n), x = a$
- $(\phi(x_1, \ldots x_n) \vee \phi'(x_1, \ldots x_n))$, $(\phi(x_1, \ldots x_n) \wedge \phi'(y_1, \ldots y_m))$, $(\phi(x_1, \ldots x_n) \wedge \neg\phi'(x_{i_1}, \ldots x_{i_j}))$
- $(\phi(x_1, \ldots x_n) \wedge x_i = x_j)$, $(\phi(x_1, \ldots x_n) \wedge x_i \neq x_j)$, $(\phi(x_1, \ldots x_n) \wedge x_i = y)$, $(\phi(x_1, \ldots x_n) \wedge x_i \; contains \; x_j)$, $(\phi(x_1, \ldots x_n) \wedge x_i \; contains \; a)$, $(\phi(x_1, \ldots x_n) \wedge L(x_i, y_1, \ldots y_m))$, $(\phi(x_1, \ldots x_n) \wedge N(x_i, y_1, \ldots y_m))$, $(\phi(x_1, \ldots x_n) \wedge Path(x_i, R, y))$
- $\exists x_i \phi(x_1, \ldots x_n)$

Web safe calculus [3]

Theorem

Una fórmula Web calculus expresa una consulta Web si y solo si es equivalente a una formula en Web safe calculus

Para las consultas eventualmente computables se define un **semi-safe Web calculus**

Web dinámica

- Si dejamos de ver la Web como estática, consideramos que cambia en el tiempo consultas del tipo “**Encuentra documentos alcanzables desde mi página**’ ya no es computable
- Pueden aparecer nuevos documentos mientras navego para computar una consulta
- Se requiere un nuevo tipo de Web machine.

Web machines dinámicas

Web machines dinámicas

Una Web machine dinámica es una variación de Web machine pensada para una Web dinámica

- Corre sobre una secuencia infinita de esquemas de bases de datos Web (Node,Link). Cada esquema representa a la Web en distintas instancias de tiempo
- El oráculo puede cambiar de una base de datos a la siguiente en cualquier momento (oráculo dinámico)

Web query dinámica

Una Web query dinámica es un mapeo no-determinístico de secuencias de relaciones (posiblemente infinitas).

Se mantienen la separación de consultas computables, eventualmente computables y no computables por máquinas Web dinámicas.

References I

Abiteboul, S. and Vianu, V. (1997).
Queries and computation on the web.
In *ICDT*, pages 262–275.

Mendelzon, A. O., Mihaila, G. A., and Milo, T. (1996).
Querying the world wide web.
In *PDIS '96*, pages 80–91.