Foundations and Applications of Path Operations
– Propuesta de Tesis de Doctorado –

Roberto Antonio García Segura (Estudiante)
Renzo Angles Rojas (Profesor guía)
Doctorado en sistemas de ingeniería
Universidad de Talca

22 de marzo de 2022

1. Introducción

Un grafo es un modelo de datos abstracto muy versátil, que permite modelar problemas simples y complejos en distintos dominios de aplicación, incluyendo financial services, manufacturing, government, data regulation and privacy, marketing, AI and machine learning research [24]. En su definición más básica, un grafo [13] esta compuesto de un conjunto de nodos (o vértices), y de un conjunto de aristas. Cada arista vincula un par de nodos, y puede ser dirigida o no dirigida, es decir la relación entre los nodos puede o no tener una dirección.

Una base de datos orientada a grafos es aquella donde las estructuras de datos para los esquemas y las instancias son modeladas como un grafo o una generalización de estos [10]. En una base de datos orientada a grafos los nodos representan las entidades, mientras que las aristas representan las relaciones entre las entidades.

Dentro de los sistemas de gestión de bases de datos orientadas a grafos usados en la actualidad tenemos: Neo4J [18], Microsoft Azure Cosmos DB [2], OrientDB [19], entre otros [9]. Y dentro de las ventajas tenemos: su rendimiento cuando se utilizan datos altamente conectados; su naturaleza aditiva y sus resultados que muestran los datos con una mayor expresividad y simplicidad [21]. Adicionalmente, cuando una base de datos es creada y poblada, una consulta se puede realizar tal como se ve un grafo, es decir, como un subgrafo, lo que permite evitar el uso del operador de reunión como en una base de datos relacional [23].

Un componente importante para cualquier base de datos es un lenguaje de consulta, el cual consiste en una colección de operadores o reglas de inferencia que pueden ser aplicadas a cualquier instancia valida de una base de datos, lo anterior,

con el objetivo de consultar y manipular datos en cualquier combinación deseada [7]. Actualmente existen múltiples lenguajes de consulta para grafos como por ejemplo: Cypher [17], G-Core [8], Gremlin [1], GSQL [4], PGQL [5], Tiger Graph [6], SPARQL [20], entre otros. Por otro lado, existe un grupo de trabajo que está trabajando en un lenguaje de consulta estándar para grafos denominado GQL [3], el cual actualmente esta en etapa de diseño.

Una funcionalidad que caracteriza a los lenguajes de consulta para grafos son las consultas de caminos (en inglés, "path queries"). Un camino ("path" en inglés) corresponde a una secuencia de aristas en un grafo. El objetivo de una consulta de caminos es obtener todos los caminos que conectan a dos nodos en un grafo. La búsqueda de caminos es una operación compleja de procesar y ha sido ampliamente estudiada tanto en la teoría como en la práctica [9].

Por ejemplo, consideremos una red social representada como un grafo, donde los vértices representan a las personas y las aristas representan a las relaciones entre las personas. Un tipo de consulta de caminos consiste en ver si una persona A esta conectada con una persona B, lo cual implicaría verificar las secuencias de amigos ("paths") que conectan a ambas personas.

Otro ejemplo práctico consiste en usar grafos para representar transacciones financieras entre personas u organizaciones, esto con el fin de detectar fraudes.

2. Planteamiento del problema

Actualmente los lenguajes de consulta para grafos están restringidos a buscar caminos pero no retornarlos como respuesta, ni poder operar posteriormente con ellos. Específicamente, el resultado de una consulta de caminos consiste en los nodos origen y destino que están conectados por el camino, pero dicho resultado no contiene a los nodos ni aristas que conforman el camino. En consecuencia, no es posible realizar operaciones adicionales sobre los caminos resultantes completos.

El hecho de no poder retornar los caminos y poder trabajar con ellos, reduce la capacidad de expresar consultas más complejas de manera declarativa. Por ejemplo las consultas de análisis de grafos (ej. centralidad, page rank, etc) son expresadas actualmente como algoritmos en un lenguaje de programación. Esto se observa en los algoritmos para calcular las medidas de un grafo, como es el largo, el centro, la excentricidad, entre otras [13].

La presente propuesta de tesis se enfoca en estudiar las operaciones entre caminos. Esto quiere decir que una vez encontrado un camino, poder realizar operaciones básicas de bases de datos sobre este como son juntar, filtrar entre otras [11].

En la actualidad existen algunas operaciones de manipulación de caminos co-

mo las presentadas en [22], las cuales se limitan a juntar caminos, pero en la actualidad no se ha seguido desarrollando, y tampoco han sido aplicadas en algún lenguaje de consulta.

En respuesta a los problemas descritos anteriormente proponemos estudiar un conjunto de operaciones de caminos, que permitan expresar consultas de análisis de grafos en dominios de aplicación actuales, como por ejemplo consultas de teoría de grafos, redes sociales (distancia social, centro en una red), detección de fraude (transacciones fraudulentas), redes de transporte (optimización en redes por ejemplo).

2.1. Operaciones para caminos

En la literatura actual existen algunos estudios relacionados con operaciones de caminos. Gondran [12] define una estructura algebraica general para operaciones de caminos, la cual define operadores de suma y multiplicación, con la finalidad de simplificar los problemas de caminos de mayor complejidad. Posteriormente Manger [15] define dos operadores, reunión y producto. El álgebra propuesta busca ser más sencilla y computacionalmente eficiente. Por otro lado Naudziunas y Griffin [16] definen un lenguaje de dominio específico para la especificación de álgebras de caminos, lo anterior con la finalidad de que el álgebra y las pruebas sean automatizadas. Finalmente Rodríguez y Neubauer [22] presentan algunas operaciones para caminos, que permiten juntar caminos con el fin de cruzar grafos, mediante el uso de autómatas.

Por otro lado, algunos sistemas de bases de datos han comenzado a implementar algunas operaciones de caminos, como es el caso de Neo4J [18], el que posee algunas operaciones básicas, como son crear, combinar, dividir y obtener los elementos de un camino.

Dentro de las operaciones que podrían ser adaptadas para el procesamiento de caminos, están las operaciones básicas encontradas en una base de datos como son: proyección, selección, unión, diferencia, intersección, reunión y producto cruz [11]. Con el fin de introducir el desarrollo de las operaciones mencionadas anteriormente, se mostrará mediante el uso del operador de reunión un análisis semántico para mostrar su poder expresivo y complejidad según la semántica aplicada. Adicionalmente, mediante el uso de algunos ejemplos, se mostrará el resultado de cada una de las semánticas propuestas para el operador.

Operador de reunión

La reunión es un operador que permite reunir objetos. En el caso de las bases de datos relacionales, el operador de reunión permite reunir tuplas de dos tablas. En este caso, se definirá como un operador que permitirá reunir caminos entre

dos conjuntos de caminos. Para esto debemos definir el significado de reunir dos caminos, y después generalizarlo a conjuntos de caminos.

Por simplicidad, asumiremos que un camino se representa como una secuencia de nodos. Por ejemplo, la secuencia (n_1, n_2, n_3) representa un camino compuesto de tres nodos. Dados dos caminos P_1 y P_2, consideremos las siguientes semánticas para la operación de reunión entre dicho par de caminos:

1. Si el nodo final de P_1 es igual al nodo inicial de P_2 entonces se concatenan P_1 con P_2. Por ejemplo, la reunión de (n_1, n_2, n_3) con (n_3, n_4, n_5), resultará en el camino $(n_1, n_2, n_3, n_4, n_5)$.

2. Si el nodo final de P_1 es igual a algún nodo de P_2 entonces se concatenan P_1 con P_2, descartando parte del camino P_2, según sea la posición del nodo en común. Por ejemplo la reunión de (n_1, n_2, n_3) con (n_2, n_3, n_4), resultará en el camino (n_1, n_2, n_3, n_4).

3. Si algún nodo de P_1 es igual al nodo inicial de P_2 entonces se concatenan P_1 con P_2, descartando parte del camino P_1, según sea la posición del nodo en común. Por ejemplo la reunión de (n_1, n_2, n_3) con (n_2, n_4, n_5), resultará en el camino (n_1, n_2, n_4, n_5).

4. Si algún nodo de P_1 es igual a algún nodo de P_2 entonces se concatenan P_1 con P_2, descartando parte de los caminos, según sea la posición del nodo en común. Por ejemplo la reunión de (n_1, n_2, n_3) con (n_4, n_2, n_5) mediante el nodo n_2, resultará en el camino (n_1, n_2, n_5). Esto se replicaría para todos los nodos en común entre P_1 y P_2.

5. Si se desea concatenar dos caminos P_1 y P_2, manteniendo los nodos de cada camino, se crea una nueva arista que une ambos caminos. Por ejemplo la reunión de (n_1, n_2, n_3) con (n_1, n_2, n_4), resultará en el camino $(n_1, n_2, n_3, n_1, n_2, n_4)$.

Cabe destacar que las definiciones anteriores no son las únicas que se podrían definir, ya que podría existir por ejemplo el caso de realizar la reunión entre posiciones en especifico, o entre los nodos finales de cada camino. Por otro lado, las semánticas anteriores pueden ser empleadas para definir la reunión entre dos conjuntos de caminos. En este caso, sería necesario comparar cada uno de los caminos del primer conjunto con los del segundo conjunto, aplicando alguna de las semánticas anteriores.

2.2. Dominios de aplicación

Dentro de los dominios de aplicación donde la manipulación de caminos puede ser necesaria encontramos: teoría de grafos, redes de transporte, optimización,

fraude o redes sociales. Para cada dominio de aplicación describiremos algunos casos de uso, es decir, consultas que no pueden ser expresadas por los lenguajes de consulta existentes, pero podrían expresarse a través de operaciones entre caminos.

Teoría de grafos En teoría de grafos existe el concepto de medida, el cual nos permite obtener información de un grafo. Dentro de las medidas de un grafo podemos mencionar [13]: largo (número de aristas), distancia (número de aristas entre dos vértices), excentricidad (distancia máxima entre un par de vértices), diámetro (excentricidad máxima de todos los vértices), radio (excentricidad mínima de todos los vértices) y centro (vértices que tienen la misma excentricidad que el radio).

Las medidas antes expuestas podrían ser calculadas mediante el uso de operaciones de caminos, como serían proyección o selección en el caso del largo y la distancia de un grafo u operaciones agregadas para el caso de la excentricidad y sus derivadas.

El uso de estas medidas en un grafo tiene múltiples aplicaciones. Los valores obtenidos pueden ser utilizados para estimar la eficiencia en ciertos procesos de comunicación o transporte. Cabe mencionar que las medidas en grafos pueden ser generalizadas y utilizadas en otros casos de uso.

Redes de transporte Las redes de transporte permiten al usuario obtener rutas entre dos o más destinos según su configuración. Como una red de transporte puede ser fácilmente abstraída a un grafo, donde los nodos representan las ciudades, puntos de interés u otros objetos, las aristas representan las conexiones entre ellos. Un ejemplo muy común para este caso consiste en encontrar un camino de A a B para dos personas, pero que no se crucen.

Optimización Este dominio tiene relación con el área de gestión de operaciones, donde se busca maximizar o minimizar una o mas funciones objetivo de un problema en particular. Dentro de los problemas tratados en optimización están los de optimizar caminos, como son los de el vendedor viajero, ruteo de vehículos, donde se busca minimizar el costo de un camino. Este costo puede pertenecer a una propiedad del camino, como por ejemplo el largo. En este caso una operación de selección podría resultar útil para la resolución del problema.

Fraude El dominio de aplicación relacionado con fraude tiene como ventaja el impacto en la industria bancaria o de investigación debido a que podría permitir obtener información para evitar fraudes bancarios, fraude en entrega de productos, entre otros. Dentro de los datos utilizados en este dominio se encuentran: uso de

cuentas de usuario, uso de tarjetas de crédito, duración de las ordenes de compra, entre otros.

Redes sociales Las redes sociales son un dominio de aplicación que posee dos importantes características, la sencillez de sus datos y la cantidad de información. Sencillez debido a la facilidad de abstraer una red social en una estructura de grafo, donde los nodos representarían a las personas y las aristas sus relaciones de amistad entre otras. Existen distintos tipos de redes sociales, fuera de las redes de amistad, dentro de las cuales se tienen: redes de teléfonos, redes de correo electrónico, redes de colaboración, redes de información, redes de infraestructura, redes biológicas, entre otras [14].

El uso de operaciones de caminos sobre un red social podría permitir resolver distintos problemas, como por ejemplo, obtener los entes influenciadores en la red o análisis de puentes entre dos redes.

3. Preguntas de investigación

- ¿Qué operaciones entre caminos se pueden definir?
- ¿Qué propiedades tendrían las operaciones entre caminos?
- ¿Cuál será el poder expresivo de las operaciones de caminos?
- ¿Cuál será la complejidad computacional de las operaciones entre caminos?

4. Objetivos

4.1. Objetivo general

Desarrollar los fundamentos de las bases de datos para manipulación de caminos y mostrar su aplicación en casos de uso reales.

4.2. Objetivos específicos

- (O1) Definir un conjunto de operaciones para caminos tomando como base casos de uso reales.
- (O2) Estudiar las propiedades y características de las operaciones para caminos.
- (O3) Implementar las operaciones para caminos como parte de un lenguaje de consulta declarativo para grafos.

- (O4) Evaluar las operaciones para caminos en distintos dominios de aplicación.

5. Alcances y limitaciones

Alcances

- Las operaciones que se definirán serán más expresivas que las operaciones básicas de los lenguajes de consulta para grafos actuales.
- Las operaciones seleccionadas estarán asociadas con los casos de uso que se definirán.
- Se implementará un lenguaje de consulta experimental que permitirá expresar y probar las operaciones para caminos que se definirán.

Limitaciones

- No se espera definir todas las posibles operaciones entre caminos. Se pondrá énfasis en aquellas que tengan aplicación práctica (en base a los casos de uso).
- Las operaciones de caminos no serán implementadas de manera nativa en un sistema de gestión de bases de datos para grafos. En lugar de ello, se empleará un sistema existente.
- No se desarrollarán técnicas de optimización de consultas avanzadas para procesar las operaciones entre caminos.

6. Metodología

Con el fin de cumplir con los objetivos propuestos, se han previsto realizar tareas en base a los objetivos definidos, las cuales serán realizadas durante un periodo de 2 años.

Objetivo 1: Definir un conjunto de operaciones para caminos tomando como base casos de uso reales.

- (T1.1) Identificar un conjunto de casos de uso reales que requieran aplicar operaciones de caminos.
- (T1.2) Estudiar operaciones existentes en bases de datos que puedan ser extrapoladas a operaciones para caminos.

- (T1.3) Definir operaciones de caminos unitarias.

- (T1.4) Definir operaciones entre pares de caminos.

- (T1.5) Definir operaciones entre conjuntos de caminos.

- (T1.6) Definir operaciones entre multiconjuntos de caminos.

- (T1.7) Analizar la relación entre las operaciones para caminos y los casos de uso.

Objetivo 2: Estudiar las propiedades y características de las operaciones para caminos.

- (T2.1) Estudiar las propiedades de las operaciones definidas.

- (T2.2) Estudiar la complejidad computacional de las operaciones definidas.

- (T2.3) Estudiar el poder expresivo de las operaciones definidas.

- (T2.4) Comparar las operaciones definidas con el estado del arte.

Objetivo 3: Implementar las operaciones de caminos como parte de un lenguaje de consulta para grafos.

- (T3.1) Seleccionar el lenguaje de consulta para grafos que será extendido con las operaciones para caminos.

- (T3.2) Extender la sintaxis del lenguaje de consulta seleccionado para soportar las operaciones para caminos.

- (T3.3) Implementar las operaciones de caminos sobre un motor de gestión de base de datos.

- (T3.4) Evaluar el funcionamiento de las operaciones implementadas.

Objetivo 4: Evaluar las operaciones de caminos en distintos dominios de aplicación.

- (T4.1) Definir un ambiente experimental para probar las consultas de caminos.

- (T4.2) Seleccionar dominios de aplicación para evaluar las operaciones de caminos.

- (T4.3) Seleccionar los casos de prueba para cada dominio de aplicación.

- (T4.4) Diseñar y evaluar las consultas correspondientes a los casos de prueba.

7. Planificación

La planificación sera presentada como una lista de entregables agrupada por año. En la Tabla 1 se encuentra el plan de actividades en base a las tareas de los objetivos. Adicionalmente a continuación se detallara una lista de entregables la cual sera dividida durante los dos años.

	Año 1												Año 2											
Tarea/Mes	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
T1.1																								
T1.2																								
T1.3																								
T1.4																								
T1.5																								
T1.6																								
T1.7																								
T2.1																								
T2.2																								
T2.3																								
T2.4																								
T3.1																								
T3.2																								
T3.3																								
T3.4																								
T4.1																								
T4.2																								
T4.3																								
T4.4																								

Tabla 1: Plan de actividades

Entregables año 1:

- (E1.1) Artículo de conferencia o workshop.

Entregables año 2:

- (E2.1) Software: Lenguaje de consulta experimental extendido con operaciones de caminos.

- (E2.2) Artículo de revista: Extensión de un lenguaje de consulta para grafos, con operaciones de caminos.

- (E2.3) Artículo de revista: Operaciones entre caminos.

- (E2.4) Tesis doctoral.

Referencias

[1] Apache TinkerPop. https://tinkerpop.apache.org/gremlin.html.

[2] Azure Cosmos DB: servicio de base de datos multimodelo — Microsoft Azure. https://azure.microsoft.com/es-es/services/cosmos-db/.

[3] Graph Query Language GQL. https://www.gqlstandards.org/home.

[4] GSQL - TigerGraph. https://www.tigergraph.com/gsql/.

[5] PGQL — Property Graph Query Language. https://pgql-lang.org/.

[6] TigerGraph. https://www.tigergraph.com/.

[7] Renzo Angles. A comparison of current graph database models. Technical report, 2012.

[8] Renzo Angles, Marcelo Arenas, Pablo Barceló, Peter Boncz, George Fletcher, Claudio Gutierrez, Tobias Lindaaker, Marcus Paradies, Stefan Plantikow, Juan Sequeda, Oskar Van Rest, and Hannes Voigt. G-CORE a core for future graph query languages. In Eldawy A Bernstein P Das G. Jermaine C., editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD '18, pages 1421–1432, New York, New York, USA, 2018. ACM Press.

[9] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. Foundations of modern query languages for graph databases. *ACM Computing Surveys*, 50(5), nov 2017.

[10] Renzo Angles and Claudio Gutierrez. An Introduction to Graph Data Management. Technical report, 2018.

[11] E. F. Codd. Data models in database management. *ACM SIGMOD Record*, 11(2):112–114, 1981.

[12] M. Gondran. Path Algebra and Algorithms. In *Combinatorial Programming: Methods and Applications*, pages 137–148. Springer Netherlands, Dordrecht, 1975.

[13] Dieter Jungnickel. *Graphs, Networks and Algorithms*, volume 5 of *Algorithms and Computation in Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[14] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining Social-Network Graphs*. 2020.

[15] Robert Manger. A new path algebra for finding paths in graphs. *Proceedings of the International Conference on Information Technology Interfaces, ITI*, pages 657–662, 2004.

[16] Vilius Naudžiunas and Timothy G. Griffin. A domain-specific language for the specification of path algebras. Technical report, 2011.

[17] Neo4j. Cypher Graph Query Language. https://neo4j.com/cypher-graph-query-language/, 2020.

[18] Neo4j. Neo4j Graph Platform – The Leader in Graph Databases. https://neo4j.com/, 2020.

[19] OrientDB. Graph Database — Multi-Model Database — OrientDB. https://orientdb.com/, 2019.

[20] Eric Prud'hommeaux and Andy Seaborne. SPARQL query language for RDF. https://www.w3.org/TR/rdf-sparql-query/, 2011.

[21] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases*.

[22] Marko A. Rodriguez and Peter Neubauer. A path algebra for multi-relational graphs. In *Proceedings - International Conference on Data Engineering*, pages 128–131, 2011.

[23] Pramod J. Sadalage and Martin Fowler. *NoSQL distilled : a brief guide to the emerging world of polyglot persistence*. Addison-Wesley, 2012.

[24] Vadim Zverovich. *Modern Applications of Graph Theory*. Oxford University Press, New York, NY, 2021.