Lenguajes y modelos subyacentes a los grafos de conocimiento

Renzo Angles
rangles@utalca.cl
https://orcid.org/0000-0002-6740-9711
Departamento de Ciencias de la Computación, Facultad de Ingeniería
Universidad de Talca, Curicó, Chile

Recibido: 07/10/2022

https://doi.org/10.26439/ciis2022.6065

Resumen

Un grafo de conocimiento es una gran base de datos que integra información desde distintas fuentes de datos, esto con el objetivo de poder extraer conocimiento y transformarlo en valor para los usuarios. Dicha base de datos es representada como un grafo donde los nodos representan entidades, y cada arista representa una relación entre dos nodos, o un atributo de un nodo. El objetivo de este artículo es presentar una revisión de los modelos de datos que se usan para representar grafos de conocimiento, y los lenguajes de consulta que permiten extraer la información explícita e implícita contenida en dichos grafos.

Palabras Clave: Grafo de conocimiento, Modelos de datos basado en grafos, Lenguaje de consulta para grafos.

Languages and models underlying knowledge graphs

Abstract

A knowledge graph is a large database that integrates information from different data sources, this with the objective of supporting knowledge extraction, and allowing the transformation of such knowledge into value for the users. Such database is represented as a graph where the nodes represent entities, and each edge represents a relationship between nodes, or the attribute of a node. The objective of this article is to review the data models used to represent knowledge graphs, and the query languages that allow to extract explicit and implicit information contained in such graphs.

Keywords: Knowledge graph, Graph Data Model, Graph Query Language.

Cómo citar

Angles, R. (2022). Lenguajes y modelos subyacentes a los grafos de conocimiento. *Actas del Congreso Internacional de Ingeniería de Sistemas 2022: Entornos híbridos en la pospandemia: posibilidades para las nuevas tecnologías,* e6065. https://doi.org/10.26439/ciis2022.6065

1. Introducción

Los términos "dato", "información" y "conocimiento" son fundamentales en áreas del conocimiento de gran interés científico e impacto tecnológico en la actualidad. Entre dichas áreas se encuentran: Bases de Datos (*Databases*), Datos Masivos (*Big Data*), Web Semántica (*Semantic Web*), Inteligencia Artificial, entre otros. Para explicar dichos conceptos, consideremos la siguiente frase: “Rosa Rosales cortó una rosa, que roja es la rosa de Rosa Rosales.”

En términos muy generales, un dato puede definirse como una palabra sin significado obvio, o una palabra ambigua (es decir, puede entenderse o interpretarse de diversas maneras). Por ejemplo, la palabra "rosa" puede tener varios significados: una flor, un nombre, un color, una marca, o incluso un lugar. Ahora, si acotamos dichos significados a la frase de ejemplo, entonces "rosa" puede ser un nombre o una flor. Luego, para determinar el significado preciso de una ocurrencia de la palabra "rosa" en la frase de ejemplo, debemos analizar las palabras que aparecen a su alrededor. Por ejemplo, si consideramos el fragmento "Rosa Rosales", podemos inferir que "Rosa" es un nombre; en el caso de "que roja es la rosa", podemos saber que se refiere a una flor. Es decir, el significado de un dato puede dilucidarse analizando las relaciones que tiene con otros datos. En este sentido, la información se puede definir como un conjunto de datos cuyo significado se infiere de sus relaciones y un contexto. Finalmente, el término "conocimiento" puede definirse como la información implícita que puede extraerse al procesar la información explícita o existente. Por ejemplo, el número de veces que cada palabra aparece, y la palabra que aparece el mayor número de veces, son dos ejemplos de conocimiento que puede extraerse de la frase de ejemplo.

Desde el inicio de la era digital, a finales de los años 1950, se han investigado problemas y desarrollado soluciones para gestionar datos, información y conocimiento. Inicialmente, la preocupación estaba en almacenar (codificar) y procesar (leer y escribir) los datos. Con la aparición del modelo de datos relacional en 1970, se propusieron distintas formas de modelar (o representar) los datos y extraer (o consultar) la información subyacente. La creación de la Web (en 1989) marca el inicio de otra etapa, una donde las personas y los sistemas empiezan a generar información y conocimiento, esto gracias a estándares para transferir datos (HTTP), generar información (HTML), y representar conocimiento (RDF, RDF Schema, OWL). La Web motiva el desarrollo de sistemas informáticos interconectados, los cuales generan información que se caracteriza por su volumen (cantidad), variedad (heterogeneidad) y velocidad (de generación), dando lugar al concepto de Big Data. En este punto, se empiezan a desarrollar plataformas (como Apache Hadoop) que permiten el procesamiento y análisis de

datos masivos, esto empleando técnicas de almacenamiento distribuido de datos y computación paralela. Las organizaciones se dan cuenta del conocimiento implícito existente en los datos que producen, por lo que empiezan a impulsar proyectos de ciencia de datos (Data Science), esto con el objetivo de generar valor empleando métodos estadísticos y técnicas avanzadas de minería de datos. La existencia de grandes cantidades de datos no solo permite extraer conocimiento, también permite que los sistemas de inteligencia artificial aprendan de los datos, logrando así predicciones más precisas. Podríamos decir que actualmente nos encontramos en "La era del conocimiento", ya que estamos desarrollando métodos que nos permitan representar, analizar y generar conocimiento, o mejor dicho, grafos de conocimiento.

2. Grafos de conocimiento

No existe una definición estándar para el término "grafo de conocimiento" (*knowledge graph*), pero presentaremos tres definiciones que consideramos relevantes.

- Un grafo dirigido cuyos nodos son unidades de conocimiento (conceptos) que un estudiante debe adquirir, y cuyas aristas denotan dependencias entre dichas unidades de conocimiento (Scheneider, 1972). Esta es la primera definición del término "grafo de conocimiento", que se incluye en un artículo publicado en 1972.
- Un modelo inteligente (un grafo) que comprende entidades del mundo real y las relaciones entre ellas. Esto corresponde a un fragmento del post[1] que se usó para dar a conocer el Google Knowledge Graph.
- Un grafo de datos destinado a acumular y transmitir conocimiento del mundo real, cuyos nodos representan entidades de interés, y cuyas aristas representan relaciones potencialmente diferentes entre dichas entidades. Esta definición es parte de un artículo (Hogan et al., 2021) que revisa modelos, lenguajes, herramientas y dominios de aplicación asociados a los grafos de conocimiento.

Luego de revisar diversos trabajos en el área, nuestra definición de grafo de conocimiento es la siguiente: "Una gran base de datos que integra información desde distintas fuentes de datos, esto con el objetivo de generar información adicional y conocimiento. Dicha base de datos es representada como un grafo, es decir las entidades se representan como nodos, y las relaciones entre dichas entidades se representan como aristas".

Actualmente existen diversos modelos, lenguajes y métodos que nos permiten representar, analizar (consultar) y generar grafos de conocimiento. A continuación, describimos algunos

[1] https://www.blog.google/ products/search/introducing-knowledge-graph-things-not/

de ellos.

3. Modelos para representar grafos de conocimiento

El desarrollo de un grafo de conocimiento es un proceso complejo, y una de las primeras tareas es el modelado conceptual, lo que consiste en identificar tipos de entidades, tipos de relaciones y tipos de restricciones, y representar dichos tipos usando un modelo de datos (Brodie et al., 1984). De una manera muy general, un modelo de datos se define como una colección de herramientas conceptuales usadas para modelar representaciones de entidades del mundo real y las relaciones entre ellas (Silberschatz et al., 1996). Existen distintos modelos de datos (Beeri, 1988), los cuales pueden agruparse según la estructura abstracta en la cual están basados, como por ejemplo tablas, árboles o grafos.

El modelado conceptual de un grafo de conocimiento puede realizarse usando cualquier modelo de datos, pero es más natural y usual emplear un modelo basado en grafos, ya que está pensado para representar de mejor manera las conexiones entre las entidades. Existen diversos modelos de datos basados en grafos, algunos teóricos (Angles & Gutierrez, 2008) y otros más prácticos (Angles et al., 2017). Los modelos más usados en la actualidad son tres: grafo dirigido etiquetado, grafo con propiedades, y grafo RDF.

Un grafo dirigido etiquetado (directed labeled graph) (Barceló Baeza, 2013) es una estructura compuesta de nodos y aristas donde: los nodos y las aristas pueden tener identificadores y etiquetas; cada arista conecta un par de nodos; las aristas son dirigidas ya que tienen un nodo origen y destino; y pueden existir múltiples aristas entre dos nodos (multigrafo). Desde el punto de vista de modelado de datos, un nodo representa una entidad o un valor, y una arista representa una relación entre dos entidades o un atributo de una entidad. Este modelo es muy usado en métodos estadísticos, minería de datos, inteligencia artificial, y sistemas de procesamiento masivo de datos.

Un grafo con propiedades (property graph) (Angles, 2018) es un grafo dirigido etiquetado, pero tiene una característica extra: cada nodo o arista puede mantener un conjunto (posiblemente vacío) de propiedades, donde una propiedad tiene un nombre (o etiqueta) y un valor. En este modelo, un nodo representa una entidad, una arista representa una relación entre dos entidades, y una propiedad representa una característica específica y propia de una entidad o una relación. Este modelo es muy usado por los sistemas de gestión de bases de datos que soportan el almacenamiento y consulta de de grafos (Angles & Gutierrez, 2018).

RDF (Resource Description Framework) es una estándar para describir recursos Web, el cual define un modelo de datos basado en grafos. En un grafo RDF existen tres tipos de nodos: recursos Web identificados por una especie de URL (llamada IRI), nodos blancos que

representan recursos anónimos, y literales que representan datos o valores concretos (como cadenas, números y fechas). Un triple RDF es una tupla de la forma (sujeto, predicado, objeto), donde el sujeto puede ser un recurso Web o un recurso anónimo; el predicado suele ser un recurso Web (que referencia a un tipo de relación), y el objeto puede ser un recurso Web o un valor concreto. Según lo anterior, el sujeto y el objeto serían nodos en el grafo, y el predicado es el identificador de una relación. Nótese que, en comparación con un grafo con propiedades, una arista puede representar una relación o una propiedad. Un grafo RDF es la manera estándar de representar datos en la Web Semántica.

Existen otros modelos que tratan de sobrellevar las deficiencias de estos modelos (Angles et al., 2022), o han sido diseñados para dominios de aplicación especiales. Por ejemplo, el modelo basado en hipergrafos (hypergraphs) (Iordanov, 2010) permite representar relaciones n-arias con más facilidad, pero su implementación es más compleja. RDF* (Hartig, 2017) extiende el modelo RDF con el objetivo de permitir metadatos sobre las propiedades de un triple RDF. En un trabajo reciente (Lassila et al., 2023), los autores plantean la idea de un único modelo de datos basado en grafos denominado OneGraph, el cual busca unificar RDF y los grafos con propiedades, esto con el objetivo de permitir interoperabilidad entre ambos modelos, y facilitar el desarrollo de lenguajes de consulta.

Actualmente existen diversos sistemas de gestión de datos, que permiten almacenar y consultar grafos de conocimiento (DB-Engines Ranking of Graph DBMS, 2022). Estos sistemas podemos dividirlos en cuatro grupos: sistemas de base de datos para grafos (graph database systems), como Neo4j (The Neo4j Graph Platform, 2022) y TigerGraph (2022), que en su mayoría soportan property graphs; RDF Triple stores, como Apache Jena (2022) y GraphDB (2022), que son diseñados para gestionar grafos RDF; plataformas de procesamiento distribuido de grafos (distributed graph processing frameworks), como Apache Giraph (2022) y GraphX (2022), que permiten manipular grafos dirigidos etiquetados de gran tamaño; y sistemas multi-modelo, como Amazon Neptune (2022) and Microsoft Azure Cosmos DB (2022), que soportan múltiples modelos, usualmente RDF y grafos con propiedades.

4. Lenguajes para consultar grafos de conocimiento

Un componente clave de cualquier sistema de gestión de datos es su lenguaje de consulta. En términos generales, un lenguaje de consulta es un lenguaje computacional de alto nivel que permite recuperar los datos almacenados en un sistema de gestión de datos (Samet, 1981). En el caso de los sistemas de gestión de datos basados en grafos, un lenguaje de consulta para grafos (*graph query language*) está diseñado para extraer información usando

operaciones orientadas a grafos, como patrones y consultas de caminos (Angles et al., 2018b).

Existen varios trabajos relacionados a los lenguajes de consulta para grafos, los cuales tratan aspectos como su definición (Angles & Gutierrez, 2008; Angles, 2012), expresividad (es decir, los tipos de consultas que el usuario puede expresar) (Angles et al., 2022), complejidad computacional (es decir, el tiempo requerido para evaluar distintos tipos de consultas) (Barceló Baeza, 2013; Angles et al., 2017; Bonifati & Dumbrava, 2019), implementación (Fletcher, Peters, & Poulovassilis, 2016), y evaluación (Ciglan, Averbuch, & Hluchy, 2012). El libro de Bonifati et al. (2018), presenta una revisión completa sobre consultas orientadas a grafos.

A pesar de los avances en el desarrollo de lenguajes de consulta para grafos, aún no existe un estándar. Sin embargo, en septiembre del 2019 se inició el proyecto GQL (GQL Standard, 2022) cuyo objetivo es definir un lenguaje estándar que se basa en la fusión de cuatro lenguajes existentes: Cypher (Neo4j) (Cypher Query Language, 2022), PGQL (Oracle) (van Rest et al., 2013), G-CORE (LDBC) (Angles et al., 2018a) y GSQL (TigerGraph) (Deutsch et al., 2020).

La principal noción detrás de un lenguaje de consulta para grafos es la búsqueda de patrones (*graph pattern matching*) (Gallager, 2006). Un patrón de grafo simple es un subgrafo donde los nodos y las aristas pueden ser entidades, relaciones y variables. Los patrones simples se pueden combinar usando operadores relacionales, como reunión (join), unión y diferencia, permitiendo definir patrones más complejos (Angles et al., 2022). El resultado de un patrón de grafo puede ser un conjunto de grafos que satisfacen la estructura del patrón, o una tabla donde cada columna es una variable, y cada fila contiene asignaciones de variables a nodos o valores.

Aunque muchas consultas reales pueden ser expresadas como patrones de grafos, es muy frecuente necesitar de mecanismos adicionales que permitan explorar la topología del grafo (Angles et al., 2022). Esto nos lleva a la noción de consulta de camino (*path query*) (Barceló et al., 2012), que tiene que ver con navegar a través del grafo usando patrones de camino. Recordemos que, en teoría de grafos, un camino es una secuencia de nodos y aristas que nos permiten ir desde un nodo origen a un nodo destino (Diestel, 2005). En este sentido, la forma más común de representar consultas de caminos es a través de una expresión de la forma (a,p,b) donde a representa el nodo origen, b el nodo destino, y p es una expresión regular (Mendelzon & Wood, 1995). Los aspectos teóricos asociados a las consultas de caminos se han estudiado de manera amplia (Angles et al., 2022; Bonifati et al., 2018), y los sistemas de

gestión de bases de datos soportan diversos tipos de consultas de caminos. Sin embargo, aún está el desafío de ejecutar eficientemente consultas de caminos en grafos de gran tamaño, esto debido a la complejidad computacional intrínseca del problema.

Si bien los lenguajes de consulta para grafos permiten extraer información desde los grafos de conocimiento, no son lo suficientemente poderosos para expresar consultar complejas (conocimiento) propias del análisis de grafos (graph analytics). Por esta razón, se han desarrollo lenguajes que mezclan operaciones declarativas con programación (ej. Gremlin (Gremlin, 2017)), librerías que proveen algoritmos para grafos (ej. the Neo4j Graph Data Science library (Neo4j, 2020)), y sistemas especiales que permiten procesamiento de grafos a gran escala (ej. Spark GraphX (GraphX, 2022)).

5. Conclusiones

Actualmente, las organizaciones están creando muchos grafos de conocimiento con el objetivo de obtener valor de sus datos. Sin bien existen modelos y lenguajes bien establecidos para representar los grafos de conocimiento, aún existen varios desafíos prácticos relacionados con la consulta y análisis de estos grafos, esto pensando en extraer de manera eficiente el conocimiento oculto.

Además de representar y consultar los datos, necesitamos modelos para representar el conocimiento, además de métodos deductivos e inductivos que permitan generar conocimiento de manera automática. Estos temas son el foco de investigación actual en el área, y representan el estado actual de la era del conocimiento.

Referencias

Amazon Neptune. (2022). https://aws.amazon.com/es/neptune/.

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys (CSUR)*, *40*(1), 1–39.

Angles, R. (2012). A comparison of current graph database models. In *4rd International Workshop on Graph Data Management: Techniques and Applications (GDM) (ICDE Workshop).*

Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., & Vrgocˇ, D. (2017, sep). Foundations of Modern Query Languages for Graph Databases. *ACM Comput. Surv. (CSUR)*, *50*(5).

Angles, R. (2018). The Property Graph Database Model. In *Proc. Alberto Mendelzon International Workshop on foundations of data management (AMW)* (Vol. 2100). CEUR Workshop Proceedings.

Angles, R., & Gutierrez, C. (2018). An Introduction to Graph Data Management. In *Graph data management* (chap. 1). Springer Nature.

Angles, R., Arenas, M., Barceló, P., Boncz, P., Fletcher, G., Gutierrez, C., ... Voigt, H. (2018a). G-CORE: A Core for Future Graph Query Languages. In *Proc. of the International Conference on Management of Data (SIGMOD).*

Angles, R., Reutter, J., & Voigt, H. (2018b). Graph Query Languages. In *Encyclopedia of Big Data Technologies* (pp. 1–8). Cham: Springer International Publishing.

Angles, R., Hogan, A., Lassila, O., Rojas, C., Schwabe, D., Szekely, P., & Vrgocˇ, D. (2022). Multilayer Graphs: A Unified Data Model for Graph Databases. In *Proc. of the 5th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences and Systems (GRADES) and Network Data Analytics (NDA).* New York, NY, USA: ACM.

Apache Giraph. (2022). http://giraph.apache.org.

Apache Jena. (2022). https://jena.apache.org/.

Barceló, P., Libkin, L., Lin, A. W., & Wood, P. T. (2012, December). Expressive Languages for Path Queries over Graph-Structured Data. *ACM Trans. Database Syst., 37*(4), 1–46.

Barceló Baeza, P. (2013). Querying graph databases. In *Proc. of the Symposium on Principles of Database Systems (PODS)* (pp. 175–188). ACM.

Beeri, C. (1988, Aug - Sept). Data Models and Languages for Databases. In *Proceedings of the 2nd International Conference on Database Theory (ICDT)* (Vol. 326, p. 19-40). Springer.

Bonifati, A., & Dumbrava, S. (2019). Graph Queries: From Theory to Practice. *SIGMOD Rec., 47*(4), 5–16.

Bonifati, A., Fletcher, G., Voigt, H., & Yakovets, N. (2018). *Querying Graphs*. Morgan & Claypool Publishers.

Brodie, M. L., Mylopoulos, J., & Schmidt, J. W. (1984). *On Conceptual Modelling*. Springer-Verlag.

Ciglan, M., Averbuch, A., & Hluchy, L. (2012). Benchmarking Traversal Operations over Graph Databases. In *Proc. of the int. conf. on data engineering workshops* (pp. 186–189). IEEE Computer Society.

Cypher Query Language. (2022). http://neo4j.com/developer/cypher-query-language/.

DB-Engines Ranking of Graph DBMS. (2022). http://db-engines.com/en/ranking/graph+dbms.

Deutsch, A., Xu, Y., Wu, M., & Lee, V. E. (2020). Aggregation Support for Modern Graph Analytics in TigerGraph. In *Proc. of the International Conference on Management of Data (SIGMOD)* (pp. 377–392). New York, NY, USA: ACM.

Diestel, R. (2005). *Graph Theory* (Third ed., Vol. 173). Springer-Verlag New York, Inc.

Fletcher, G. H. L., Peters, J., & Poulovassilis, A. (2016). Efficient regular path query evaluation using path indexes. In *Proc. of the 19th international conference on extending database technology* (pp. 636–639). OpenProceedings.org.

Gallagher, B. (2006). Matching structure and semantics: A survey on graph-based pattern matching. In *AAAI Fall Symposium on Capturing and Using Patterns for Evidence Detection* (pp. 45–53).

GQL Standard. (2022). https://www.gqlstandards.org.

GraphDB. (2022). https://graphdb.ontotext.com.

GraphX - Apache API for graphs and graph-parallel computation. (2022). https://spark.apache.org/graphx/.

Gremlin. (2017). http://tinkerpop.apache.org/gremlin.html.

Hartig, O. (2017, June 7-9). Foundations of RDF* and SPARQL* - An Alternative Approach to Statement-Level Metadata in RDF. In *Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web* (Vol. 1912). Uruguay: CEUR Workshop Proceedings.

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., . . . Zimmermann, A. (2021, jul). Knowledge Graphs. *ACM Comput. Surv., 54*(4).

Iordanov, B. (2010). HyperGraphDB: a generalized graph database. In *Proc. of the int. conf. on web-age information management (WAIM)* (pp. 25–36). Springer-Verlag.

Lassila, O., Schmidt, M., Hartig, O., Bebee, B., Bechberger, D., Broekema, W., . . . Thompson, B. (2023). The OneGraph Vision: Challenges of Breaking the Graph Model Lock-In. *Semantic Web, 14*.

Mendelzon, A. O., & Wood, P. T. (1995, December). Finding Regular Simple Paths in Graph Databases. *SIAM J. Comput., 24*(6), 1235–1258.

Microsoft Azure Cosmos DB. (2022). https://docs.microsoft.com/en- us/azure/cosmos-db/.

Neo4j, I. (2020). *The Neo4j Graph Data Science Library.* https://neo4j.com/docs/graph-data-science/current/.

The Neo4j Graph Platform. (2022). http://neo4j.com/.

Samet, J. (1981). *Query languages - A unified approach* (Tech. Rep.). Heyden University Press, Cambridge, England.: Report of the British Computer Society Query Languages Group.

Schneider, E. W. (1972, April). Course modularization applied: The interface system and its implications for sequence control and data analysis. In *Association for the development of instructional systems (adis).* Chicago, Illinois.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (1996). Data Models. *ACM Computing Surveys*, *28*(1), 105–108.

TigerGraph. (2022). https://www.tigergraph.com.

Van Rest, O., Hong, S., Kim, J., Meng, X., & Chafi, H. (2013). PGQL: a Property Graph Query Language. In *Proc. of the int. workshop on graph data management experiences and systems (GRADES).*

Biografía

Renzo Angles es profesor asociado del Departamento de Ciencias de la Computación de la Universidad de Talca. Obtuvo su grado de bachiller en Ingeniería de Sistemas en la Universidad Católica de Santa María (Arequipa, Perú), y el grado de Doctor en Ciencias mención Computación en la Universidad de Chile. Durante 2013, realizó un postdoctorado en la VU University Amsterdam, y participó en el proyecto europeo "Linked Data Benchmark Council (LDBC)". Sus trabajos de investigación se encuentran en la intersección entre las bases de datos orientadas a grafos y la web semántica. Sus especialidades actuales son la teoría y diseño de lenguajes de consulta para grafos, y el análisis de proteínas. Sitio web personal: http://renzoangles.com.

Bio

Renzo Angles is an Associate Professor at the Department of Computer Science at the Universidad de Talca, Chile. He obtained his bachelor's degree in Systems Engineering from the Universidad Católica de Santa María, Arequipa, Peru, and Ph.D. in Computer Science from Universidad de Chile. During 2013, he did a postdoc at the VU University Amsterdam, and participated in the European project "Linked Data Benchmark Council (LDBC)". His research works are at the intersection between graph-oriented databases and the semantic web. His current specialties are graph query language theory and design, and protein analysis. Personal website: http://renzoangles.com.