Mapping Semantic Networks to Undirected Networks

Marko A. Rodriguez

***Abstract*—There exists an injective, information-preserving function that maps a semantic network (i.e a directed labeled network) to a directed network (i.e. a directed unlabeled network). The edge label in the semantic network is represented as a topological feature of the directed network. Also, there exists an injective function that maps a directed network to an undirected network (i.e. an undirected unlabeled network). The edge directionality in the directed network is represented as a topological feature of the undirected network. Through function composition, there exists an injective function that maps a semantic network to an undirected network. Thus, aside from space constraints, the semantic network construct does not have any modeling functionality that is not possible with either a directed or undirected network representation. Two proofs of this idea will be presented. The first is a proof of the aforementioned function composition concept. The second is a simpler proof involving an undirected binary encoding of a semantic network.

***Keywords*—general-modeling, multi-relational networks, semantic networks

I. Introduction

A network is a popular data structure for representing the relationship between discrete elements [7], [9]. There are various types of networks such as the undirected network (i.e. undirected unlabeled network), the directed network (i.e. directed unlabeled network), and the semantic network (i.e. directed labeled network). In an undirected network, there exists no order to the relationships between the vertices. An undirected network can be denoted $U \subseteq \{V^u \times V^u\}$, where V^u is the vertex set and any edge $\{i, j\} \in U$ denotes an undirected relationship. The directed network provides the concept of edge directionality. A directed network can be represented as $D \subseteq (V^d \times V^d)$, where V^d is the vertex set and any edge $(i, j) \in D$ denotes a directed relationship. All edges in both an undirected and directed network are homogeneous in meaning. In order to represent edge meaning, a semantic network can be used. In a semantic network, an edge connecting any two vertices maintains a label (e.g. character string) that denotes the type of relationship between two vertices. A semantic network can be represented as $S \subseteq \langle V^s \times \Omega \times V^s \rangle$, where V^s is the vertex set, Ω is the set of edge labels, and any edge (called a triple) $\langle i, \omega, j \rangle \in S$ denotes an ordered, labeled relationship.

The semantic network is perhaps best known as a modeling construct from the early days of knowledge representation in the cognitive sciences [13]. However, with the inception of the Semantic Web initiative [5], [4] and with the development of triple-store technology (i.e. semantic network databases) [8], [2], [1], there has been an increase in the use of the semantic network as a data structure for modeling data sets where there exists a heterogeneous set of vertices and edges. This trend has been occurring across various disparate domains such as bioinformatics [10], [12], digital libraries [3], [6], and general computer-science [11]. Because of the use of the labeled edge, the semantic network is seen as the better modeling construct than both the undirected and directed network for such data sets.

However, when ignoring space constraints, there is no modeling gain by using a semantic network representation as opposed to a directed network representation. Moreover, there is no modeling gain over using an undirected network representation. Through a series of information-preserving, injective mappings [1], this article demonstrates that it is possible to model a semantic network both as a directed and undirected network. While both the directed and undirected model of a semantic network utilize more vertices and edges in their representation, they ultimately have the ability to capture the same information.

The outline of this article is as follows. Section II presents an injective function to map a semantic network to a directed network. Section III presents an injective function to map a directed network to a semantic network. Finally, through function composition, Section IV presents an injective function to map a semantic network to an undirected network.

II. Mapping a Semantic Network to a Directed Network

This section will present an injective, information-preserving function that maps a semantic network to a directed network. There is a two step process to this function. First, the edge labels of a semantic network are represented as a binary string. Second, each binary string is represented as a unique directed network encoding. Given that a directed network can only represent vertices and directed edges, each edge label of the semantic network is encoded as a topological feature in the directed network.

Let $S \subseteq \langle V^s \times \Omega \times V^s \rangle$ denote a semantic network where V^s is the set of all vertices and Ω is the set of all edge labels. Any triple $\langle i, \omega, j \rangle \in S$ represents a directed edge from vertex i to vertex j with a label of ω. An example semantic network triple is diagrammed in Fig. 1.

There exists the injective function $\lambda : \Omega \rightarrow \{0, 1\}^{\lceil \log_2(|\Omega|) \rceil}$ (a binary encoder) that represents every label in Ω as a unique binary string of length $\lceil \log_2(|\Omega|) \rceil$. While the minimum bits

Marko A. Rodriguez is with the Center for Non-Linear Studies at the Los Alamos National Laboratory, Los Alamos, New Mexico, marko@lanl.gov

[1] An injective function is one such that if $f(a) = f(b)$, then $a = b$.

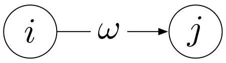

Fig. 1. An edge in a semantic network.

required to make a one-to-one mapping is $\lceil \log_2(|\Omega|) \rceil$, popular examples of other such one-to-one mappings include the ASCII and Unicode functions that map between human language characters and binary strings. Furthermore, there exist the inverse function λ^{-1} that maps a binary string to its original symbolic representation. Note that for labels already represented as unique binary strings, λ and λ^{-1} are identity functions. Given the semantic network edge diagrammed in Fig. 1, the $\lambda(\omega)$ mapping is represented in Fig. 2. Assume that $|\Omega| = 8$ and thus, each $\omega \in \Omega$ requires 3 bits to encode it.

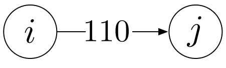

Fig. 2. A example of the $\lambda(\omega)$ mapping.

Next, there exists the injective function $\gamma : \{0,1\}^n \rightarrow \mathcal{D}$ (a directed network encoder), where $\mathcal{D}$ is the family of all directed networks and any $D \in \mathcal{D}$ is denoted $D \subseteq (V^d \times V^d)$. If $B \in \{0,1\}^n$ is the ordered multi-set (or bag) of the n-bit string $\lambda(\omega)$, then

$$\gamma(B) = \bigcup_{n=1}^{n \leq |B|} \begin{cases} (b_n, b_{n+1}) & \text{if } b_n = 0 \wedge n < |B| \\ (b_n, b_{n+1}) \cup (b_n, b_n) & \text{if } b_n = 1 \wedge n < |B| \\ (b_n, b_n) & \text{if } b_n = 1 \wedge n = |B|. \end{cases}$$

If $\lambda(\omega) = (1,1,0)$, then $\gamma(\lambda(\omega))$ is represented as diagrammed in Fig. 3. The number of vertices in D with respects to γ is $\mathcal{O}(\lceil \log_2(|\Omega|) \rceil)$. The number of directed edges in D with respects to γ is $\mathcal{O}(2\lceil \log_2(|\Omega|) \rceil - 1)$.

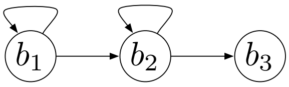

Fig. 3. A directed network representation of the edge label $\lambda(\omega) = (1,1,0)$.

The function γ is information preserving because there also exists the inverse function γ^{-1}. If $q \in \{V^d\}^n$ is the single non-looping path in D that traverses every vertex in V^d (i.e. the only Hamiltonian path), then

$$\gamma^{-1}(D) = \biguplus_{n=1}^{n \leq |q|} \begin{cases} 1 & \text{if } (q_n, q_n) \in D \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $\lambda^{-1}(\gamma^{-1}(\gamma(\lambda(\omega)))) = \omega$. From a set of functions that transform a symbolic edge label to a directed network encoding, it is possible to represent an entire semantic network as a a single directed network. In other words, given $\gamma \circ \lambda$, $S \subseteq \langle V^s \times \mathcal{D} \times V^f \rangle$.

Proposition 1 (Semantic-to-Directed Injection): A semantic network can be modeled as a directed network without loss of information. There exists an injective function $\Theta : \mathcal{S} \rightarrow \mathcal{D}$, where $D \in \mathcal{D}$ is a directed network representation of some $S \in \mathcal{S}$.

Proof. If $\Theta : \mathcal{S} \rightarrow \mathcal{D}$ denotes an injective function that maps a semantic network to a directed network, then

$$\Theta(S) = \bigcup_{\langle i,\omega,j \rangle \in S} (i, b_1) \cup (b_1, i) \cup \gamma(\lambda(\omega)) \cup (b_n, j) \cup (j, b_n),$$

where any b is a vertex in $\gamma(\lambda(\omega))$ and $n > 1$. With respects to the previous example figures, the $\Theta(S)$ mapping is diagrammed in Fig. 4.

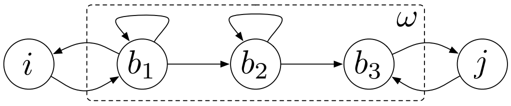

Fig. 4. A D-encoding of S.

Let $D \subseteq (V^d \times V^d)$ denote the directed network $\Theta(S)$. In V^d, every vertex that does not self-loop and has an even degree was originally a vertex in V^s. All other vertices in V^d are used to denote the edge labels of Ω. The growth of the number of vertices in D with respects to $\Theta(S)$ is $\mathcal{O}(|V^s| + |S|\lceil \log_2(|\Omega|) \rceil)$. The growth of the number of edges in D with respects to $\Theta(S)$ is $\mathcal{O}(|S|[2\lceil \log_2(|\Omega|) \rceil + 3])$.

In order to demonstrate the information-preserving quality of Θ, the inverse function Θ^{-1} also exists. Let $\Gamma : V^d \rightarrow \mathbb{N}$ denote the degree of a vertex and let $Q_{i \rightarrow j}$ be the set of paths from vertex i to vertex j in D such that

$$Q_{i \rightarrow j} = \bigcup (i, b_1, \ldots, b_n, j),$$

where $\frac{|\Gamma(i)|}{2}, \frac{|\Gamma(j)|}{2} \in \mathbb{N}$ (i.e. i and j's degree is even), $(i,i),(j,j) \notin D$ (i.e. no self-loops), $(i,b_1),(b_1,i),(b_1,\ldots),(\ldots,b_n),(b_n,j),(j,b_n) \in D$, $i \neq b_1 \neq \ldots \neq b_n$, $j \neq b_1 \neq \ldots \neq b_n$ (i.e. only i and j can be the same vertex), and no b is in a cycle with another b in the sequence. If

$$Q = \bigcup_{i,j \in V^d} Q_{i \rightarrow j},$$

then

$$\Theta^{-1}(D) = \bigcup_{q \in Q} \langle q_1, \lambda^{-1}(\gamma^{-1}(q_2, \ldots, q_{n-1})), q_n \rangle,$$

where $q_1 = i$ and $q_n = j$ and thus, the original vertices in V^s.

Given Θ and Θ^{-1}, a unique, one-to-one mapping between a semantic network and a directed network exists such that a semantic network can be modeled as a directed network without loss of information. □

There exists another proof of this concept. As demonstrated earlier, a binary string of arbitrary length can be represented as a single chain (i.e. sequence, path) of vertices, where each vertex represents a bit. In this representation, a self-loop represents a bit with value 1 and no self-loop represents a bit with value 0. Because any representation of a semantic network, at the lowest level of computing, is ultimately represented as a sequence of bits, a directed network can be used to model that sequence.

III. Mapping a Directed Network to an Undirected Network

This section presents the injective, information-preserving function $\hat{\Theta} : \mathcal{D} \rightarrow \mathcal{U}$ that maps a directed network to an undirected network. A directed network is identified by a set of ordered vertex pairs. For instance, when $D \subseteq (V^d \times V^d)$, $(i, j) \in D$ denotes a directed edge going from i (the source) to j (the sink). A directed edge between i and j is diagrammed in Fig. 5.

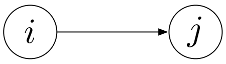

Fig. 5. An edge in a directed network.

An undirected network denoted $U \subseteq \{V^u \times V^u\}$ does not represent edge directionality as elements of U are unordered thus, $\{i, j\}$ states that i and j are connected, but that no particular direction exists. If a directed network is to be represented as an undirected network, then a topological feature in the undirected form must be used to represent edge directionality.

Proposition 2 (Directed-to-Undirected Injection): A directed network can be modeled as an undirected network without loss of information. There exists an injective function $\hat{\Theta} : \mathcal{D} \rightarrow \mathcal{U}$, where $U \in \mathcal{U}$ is an undirected network representation of some $D \in \mathcal{D}$.

Proof. The function $\hat{\Theta}$ maps each ordered vertex pair in D to a set of unique unordered vertex pairs in U. If $R_{i \rightarrow j} = \{i, x\} \cup \{x, y\} \cup \{x, z\} \cup \{y, j\} \cup \{z, j\}$, then

$$\hat{\Theta}(D) = \bigcup_{(i,j) \in D} \{i, i\} \cup R_{i \rightarrow j} \cup \{j, j\},$$

where the vertices x, y, and z are unique for each $(i, j) \in D$. Any vertex with an undirected self-loop in V^u is an original vertex from V^d. The vertices $x, y, z \in V^u$ and their respective edges represent the direction of the edge. The vertex i has one edge which denotes the tail of the original directed edge. The vertex j has two edges which denotes the head of the original directed edge. $\hat{\Theta}$ incurs a vertex growth of $\mathcal{O}(|V^d| + 3|D|)$ and an edge growth of $\mathcal{O}(|V^d| + 5|D|)$. The $\hat{\Theta}$ mapping of the directed edge represented in Fig. 5 is diagrammed in Fig. 6.

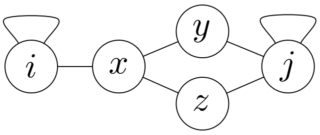

Fig. 6. An undirected network representation of a directed edge.

The function $\hat{\Theta}$ is information preserving because there exists the inverse function $\hat{\Theta}^{-1}$ such that if $q^{+} : (V \times V) \rightarrow \{0, 1\}$ is defined as

$$q^{+}(i, j) = \begin{cases} 1 & \text{if } \{i, x\}, \{x, y\}, \{x, z\}, \{y, j\}, \{z, j\} \in U \\ 0 & \text{otherwise,} \end{cases}$$

then

$$\hat{\Theta}^{-1}(U) = \bigcup_{i,j \in V^u} (i, j) \;:\; \{i, i\}, \{j, j\} \in U \;\wedge\; q^{+}(i, j) = 1.$$

Thus, a directed network can be modeled as an undirected network. □

IV. Mapping a Semantic Network to an Undirected Network

This section presents the unification of the concepts presented in the two previous sections. In this section, by means of function composition, it is demonstrated that a semantic network can be modeled as an undirected network without loss of information. This means that there exists a one-to-one mapping between a semantic network and some undirected network. In short, given the functions Θ and $\hat{\Theta}$ presented previously, an undirected network has the same representative or modeling power as a semantic network.

Proposition 3 (Semantic-to-Undirected Injection): A semantic network can be modeled as an undirected network without loss of information. There exists an injective function $\hat{\Theta} : \mathcal{S} \rightarrow \mathcal{U}$, where $U \in \mathcal{U}$ is an undirected network representation of some $S \in \mathcal{S}$.

Proof. Recall the injective functions $\Theta : \mathcal{S} \rightarrow \mathcal{D}$ and $\hat{\Theta} : \mathcal{D} \rightarrow \mathcal{U}$. Through function composition, there exists the function $\Upsilon : \mathcal{S} \rightarrow \mathcal{U}$ with the rule

$$\Upsilon(S) = \hat{\Theta}(\Theta(S)).$$

Υ incurs a vertex growth of

$$\mathcal{O}([|V^s| + 7|S|\lceil \log_2(|\Omega|) \rceil + 9|S|)$$

and an edge growth of

$$\mathcal{O}([|V^s| + 11|S|\lceil \log_2(|\Omega|) \rceil + 15|S|).$$

Finally, there also exists the inverse function Υ^{-1}, where

$$\Upsilon^{-1}(U) = \Theta^{-1}(\hat{\Theta}^{-1}(U)).$$

Thus, a semantic network can be modeled as an undirected network. □

Given the example semantic network triple diagrammed in Fig. 1, where $S = \langle i, \omega, j \rangle$ and $\lambda(\omega) = (1, 1, 0)$, the undirected network representation given by $\Upsilon(S)$ is diagrammed in Fig. 7. Note that each x, y, and z is a unique vertex even though they are not notated as such.

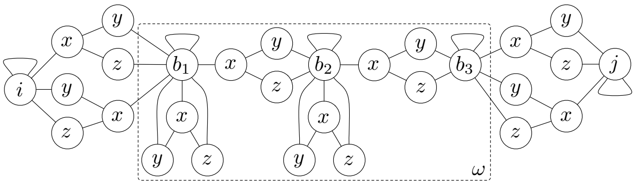

Fig. 7. An undirected network representation of a semantic network triple.

It is interesting to note the various types of self-loops in the undirected network representation in Fig. 7. There are the undirected self-loops as demonstrated by the edges $\{i, i\}$, $\{b_n, b_n\}$, and $\{j, j\}$. Next, there are the directed self-loops as demonstrated by the b_1 and b_2 sub-networks which include their respective x, y, z vertices. Finally, if $i = j$, there also exists the semantic self-loop.

There exists another method to map a semantic network to an undirected network. As discussed previously, a directed network can represent a binary string and any semantic network representation, computationally, is ultimately represented as a series of bits. Therefore, it is possible to represent a semantic network as a directed network binary string. Given $\hat{\Theta}$, it is possible to represent that directed network binary string as an undirected network.

V. Conclusion

This article defined the injective function $\Upsilon : \mathcal{S} \rightarrow \mathcal{U}$. This function demonstrates that a semantic network has a one-to-one mapping with some undirected network. In this model, because an edge in an undirected network is neither labeled nor directed, both the semantic network edge labels and the directionality of edges are represented as topological features of the undirected network. While representing a semantic network as an undirected network is perhaps an inefficient use of resources, it is theoretically possible.

References

[1] Jans Aasman. Allegro graph. Technical Report 1, Franz Incorporated, 2006.

[2] Nicole Alexander and Siva Ravada. RDF object type and reification in the database. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, page 93, Washington, DC, USA, 2006. IEEE Computer Society.

[3] Marcello Bax. RDF applications in digital library. In *International Conference on Electronic Publishing (ICCC2004)*, Brasília, Brazil, June 2004.

[4] Tim Berners-Lee and James Hendler. Publishing on the Semantic Web. *Nature*, 410(6832):1023–1024, April 2001.

[5] Tim Berners-Lee, James A. Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, pages 34–43, May 2001.

[6] Johan Bollen, Marko A. Rodriguez, Herbert Van de Sompel, Luda L. Balakireva, and Aric Hagberg. The largest scholarly semantic network...ever. In *ACM World Wide Web Conference*, Banff, Canada, Banff, Canada 2007. ACM Press.

[7] Ulrick Brandes and Thomas Erlebach, editors. *Network Analysis: Methodolgical Foundations*. Springer, Berling, DE, 2005.

[8] Ryan Lee. Scalability report on triple store applications. Technical report, Massachusetts Institute of Technology, 2004.

[9] Mark Newman, Albert-Laszlo Barabasi, and Duncan J. Watts. *The Structure and Dynamics of Networks*. Princeton University Press, May 2006.

[10] Dennis Quan, Sean Martin, and David Grossman. Applying Semantic Web techniques to bioinformatics. In *2nd International Semantic Web Conference (ISWC2003)*, Sanibel Island, Florida, October 2003.

[11] Marko A. Rodriguez. General-purpose computing on a semantic network substrate. In *Emergent Web Intelligence*, Springer-Verlag, 2009.

[12] Alan Ruttenberg, Tim Clark, William Bug, Matthias Samwald, Olivier Bodenreider, Helen Chen, Donald Doherty, Kerstin Forsberg, Yong Gao, Vipul Kashyap, June Kinoshita, Joanne Luciano, M Scott Marshall, Chimezie Ogbuji, Jonathan Rees, Susie Stephens, Gwendolyn Wong, Elizabeth Wu, Davide Zaccagnini, Tonya Hongsermeier, Eric Neumann, Ivan Herman, and Kei-Hoi Cheung. Advancing translational research with the Semantic Web. *BMC Bioinformatics*, 8(3):S2, 2007.

[13] John F. Sowa, editor. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann, San Mateo, CA, 1991.

Marko A. Rodriguez is currently a researcher at the Center for Non-Linear Studies at the Los Alamos National Laboratory. Marko received his Ph.D. in Computer Science from the University of California at Santa Cruz and his Bachelors of Science in Cognitive Science from the University of California at San Diego. Marko's research interests include semantic network computing, neocortical modeling, and scholarly network analysis.