PNWD-4299

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

A Typed Path Metric in Semantic Graphs

C Joslyn
S al-Saffar
S Purohit

July 2011

Pacific Northwest NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm

This document was printed on recycled paper.
(9/2003)

PNWD-4299

A Typed Path Metric in Semantic Graphs

C Joslyn
S al-Saffar
S Purohit

July 2011

Prepared for the U.S. Department of Energy under
Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

A Typed Path Metric in Semantic Graphs

Cliff Joslyn, Sinan al-Saffar, Sumit Purohit

July 20, 2011

Contents

1 Ontologies and Semantic Graphs

Let $[[n]] = \{1, 2, \ldots, n\}$.

Assume a knowledgebase $\mathcal{K} := \langle \mathcal{O}, G \rangle$, with ontology $\mathcal{O} = \langle \mathcal{V}, \mathcal{S}_\mathcal{V}, \mathcal{E}, \mathcal{S}_\mathcal{V} \rangle$ and semantic graph $G = \langle V, E \rangle$. This is specified as follows:

Node Types (Classes): A set of node types (classes): $\mathcal{V} := \{V^\iota\}_{\iota=1}^N$.

Class Hierarchy: Our classes are arranged in a hierarchy, or partial order: $\mathcal{S}_\mathcal{V} = \langle \mathcal{V}, \leq_\mathcal{V} \rangle$. We say $V \leq_\mathcal{V} V'$ if V is subsumed by V'. If either $V \leq_\mathcal{V} V'$ or $V \geq_\mathcal{V} V'$, then we say that V and V' are comparable (descendant-ancestor), denoted $V \sim V'$; if neither $V \leq_\mathcal{V} V'$ nor $V \geq_\mathcal{V} V'$, then we say that V and V' are non-comparable (siblings or cousins), denoted $V | V'$.

Nodes: A set of nodes: $V = \{v_i\}_{i=1}^n$.

Node Type Function: Each node has a unique type: $t_\mathcal{V} \colon V \to \mathcal{V}, t_\mathcal{V}(v_i) = V^\iota$.

We can now abuse some notation for convenience as follows. First we can also denote a node as v_i^ι, indicating its class $t_\mathcal{V}(v_i) = V^\iota$. Furthermore, since $t_\mathcal{V}$ is a function, it induces a partition of V via the equivalence classes of the nodes of a certain type. Thus for a fixed $t_\mathcal{V}$, it's convenient to reuse the notation $V^\iota = \{v_i \in V : t_\mathcal{V}(v_i) = V^\iota\} \subseteq V$ extensionally, whereas before it was an intensional type. Also denote $n^\iota = |V^\iota|$, and $V = \bigcup_{\iota=1}^N V^\iota$, so that $n = \sum_{\iota=1}^N n^\iota$.

Edge Types: A set of edge types between classes: $\mathcal{E} := \{E^\kappa\}_{\kappa=1}^M \subseteq \mathcal{V}^2$, so that $\forall E^\kappa \in \mathcal{E}, E^\kappa = \left\langle V^\iota, V^{\iota'} \right\rangle$. Establish **domain** and **range** functions $\mathrm{dom}, \mathrm{ran} \colon \mathcal{E} \to \mathcal{V}, \mathrm{dom}(E^\kappa) = V^\iota, \mathrm{ran}(E^\kappa) =$

$V^{\iota'}$. For convenience, we can also denote

$$E^\kappa = \langle \mathrm{dom}(E^\kappa), \mathrm{ran}(E^\kappa) \rangle = \left\langle V^\iota, V^{\iota'} \right\rangle.$$

Edge Type Hierarchy: Our edge types are also arranged in their own partial order $\mathcal{S}_\mathcal{E} = \langle \mathcal{E}, \leq_\mathcal{E} \rangle \subseteq \mathcal{S}_\mathcal{V}^2$. Note that $\mathcal{S}_\mathcal{E}$ is a sub-order of the product order of the class hierarchy. Thus it has the constraint that the domains and ranges of subsuming edge types are subsumed as classes:

$$E^\kappa \leq_\mathcal{E} E^{\kappa'} \quad \rightarrow \quad \mathrm{dom}\,(E^\kappa) \leq_\mathcal{V} \mathrm{dom}\left(E^{\kappa'}\right), \quad \mathrm{ran}(E^\kappa) \leq_\mathcal{V} \mathrm{ran}\left(E^{\kappa'}\right).$$

Edges: A set of edges $E = \{e_k\}_{k=1}^m$, where each edge is an ordered pair $e_k = \langle v_i, v_{i'} \rangle$.

Edge Type Function: Each edge has a unique type: $t_\mathcal{E} \colon E \rightarrow \mathcal{E}, t_\mathcal{E}\,(e_k) = E^\kappa$.

As for classes, we can abuse more notation for edges. First, also denote an edge as $e_k^{\kappa,\iota,\iota'}$, where κ indicates the node type $t_\mathcal{E}(e_k) = E^\kappa$, ι indicates the domain $V^\iota = \mathrm{dom}(E^\kappa)$, and ι' indicates the range $V^{\iota'} = \mathrm{ran}(E^\kappa)$. Then, reuse E^κ extentionally for the set of edges, $E^\kappa = \{e_k \in E : t_\mathcal{E}(e_k) = E^\kappa$. Also denote $m^\kappa = |E^\kappa|$, and $E = \bigcup_{\kappa=1}^M E^\kappa$, so that $m = \sum_{\kappa=1}^M m^\kappa$.

Consider the small example knowledgebase $\mathcal{K} = \langle \mathcal{O}, G \rangle = \langle \langle \mathcal{V}, \mathcal{S}_\mathcal{V}, \mathcal{E}, \mathcal{S}_\mathcal{E} \rangle, \langle V, E \rangle \rangle$ shown in Figs. 1–3. For classes, we have

$$\mathcal{V} = \{\text{Place}, \text{Person}, \text{Phone}, \text{Object}, \text{Entity}\}, N = 5$$

$$\mathcal{E} = \{\text{Owns-Place}, \text{Owns-Phone}, \text{Lives-At}, \text{Located-At}, \text{Owns}, \text{Relation}\}, M = 6$$

with the hierarchies shown in Fig. 1. The domains and ranges of the edge types are shown in Table 1.

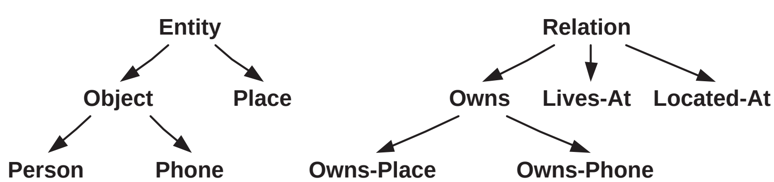

Figure 1: (Left) Node type hierarchy $\mathcal{S}_\mathcal{V}$. (Right) Edge type hierarchy $\mathcal{S}_\mathcal{E}$.

For nodes and node types, we have

$$V = \{s, u, f, m, g, p\}, n = 6$$

$$V^1 = \{s, u, f\}, n^1 = 3; \quad V^2 = \{m, g\}, n^2 = 2; \quad V^3 = \{p\}, n^3 = 1; \quad V^4 = V^5 = \emptyset, \quad n^5 = n^6 = 0$$

where the letters indicate the entities shown in the semantic graph in Fig. 2. Intensionally we have e.g. $t_\mathcal{V}(m) = V^2$, and extensionally $v_1^2 = m$, both indicating that Mario Rossi is (the first node) of type two, and is thus a Person. For edges, we have:

$$E^1 = \{\langle m, u \rangle\}, m^1 = 1; \quad E^2 = \{\langle m, p \rangle\}, m^2 = 1; \quad E^3 = \{\langle m, f \rangle, \langle m, u \rangle, \langle g, u \rangle\}, m^3 = 3;$$

$$E^4 = \{\langle p, f \rangle\}, m^4 = 1; \quad E^5 = \{\langle m, s \rangle\}, m^5 = 1; \quad E^6 = \emptyset;$$

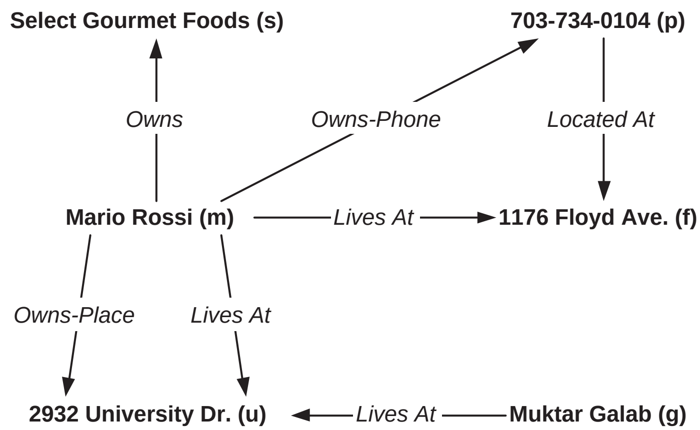

Figure 2: An example semantic graph $\langle V, E \rangle$.

κ	$\hat{\kappa}$	E^{κ}	Symbol	dom(E^{κ})	ran(E^{κ})	$\hat{\kappa}$	$E^{\kappa T}$
1	1	Owns-Place	PL	Person	Place	7	Property-Of
2	2	Owns-Phone	PH	Person	Phone	8	Phone-Of
3	3	Lives-At	LI	Person	Place	9	Abode-Of
4	4	Located-At	LO	Phone	Place	10	Location-Of
5	5	Owns	O	Person	Entity	11	Owned-By
6	6	Relation	R	Entity	Entity	12	Relation^T

Table 1: Edge types.

$$E = \{\langle m, s\rangle, \langle m, u\rangle, \langle m, p\rangle, \langle m, f\rangle, \langle m, u\rangle, \langle g, u\rangle, \langle p, f\rangle\}, m = 7.$$

Intensionally we have e.g. $t_{\mathcal{E}}(\langle g, u\rangle) = E^3$, and extensionally $e_3^3 = \langle g, u\rangle$, both indicating that Mario Rossi is related to University Drive as the (third instance of the) third edge type, and thus "Lives In".

Table 1 shows the complete intensional description of all the edge types, including their domains, ranges, and inverses (see Sec. 2 below). While a single graphical representation of the complex structure of $\mathcal{O}$ is daunting, the left side of Fig. 3 illustrates in a simplified form the "base" relations between the "leaf" edge types operating on the "leaf" node types, while on the right, all of the edge types are shown in the context of the complete inheritance hierarchy $\mathcal{V}$ amongst the node types.

2 Path Types, Path Type Expressions, and Paths

Consider each edge type as a binary relation $E^{\kappa} \subseteq V^{\iota} \times V^{\iota'}$ in the cross product of its domain and range classes, consisting of edges as ordered pairs $e = \left\langle v^{\iota}, v^{\iota'} \right\rangle$. Casting E^{κ} as an $n^{\iota} \times n^{\iota'}$ matrix $E^{\kappa}_{n^{\iota} \times n^{\iota'}}$, then $E^{\kappa T}_{n^{\iota'} \times n^{\iota}}$ is its transpose, an $n^{\iota'} \times n^{\iota}$ matrix consisting of the inverse edges $e^T = \left\langle v^{\iota'}, v^{\iota} \right\rangle$. Let

$$\hat{\mathcal{E}} := \{E^{\kappa}\}_{\kappa=1}^{M} \cup \left\{E^{\kappa T}\right\}_{\kappa=1}^{M}$$

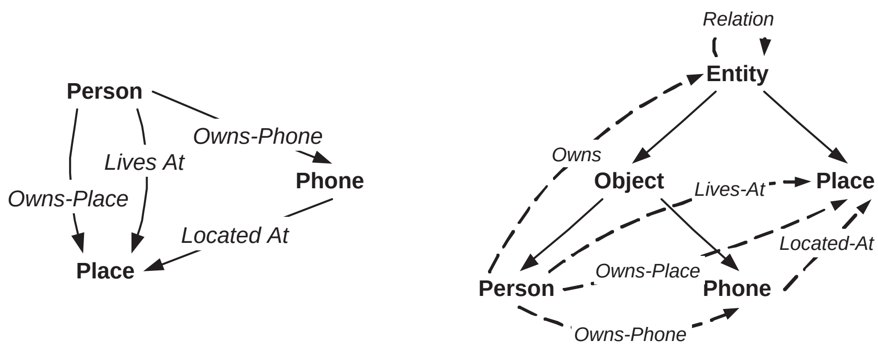

Figure 3: (Left) Example ontology cast as a graph $\mathcal{O} = \langle \mathcal{V}, \mathcal{E} \rangle$. (Right) Convolution of edge types in the context of the node hierarchy $\mathcal{V}$.

be the complete set of all such "forward" and "inverse" edge types, so that $|\hat{\mathcal{E}}| = 2M$, and refer to an **extended edge type** $\hat{E}^{\hat{\kappa}} \in \hat{\mathcal{E}}$, where now $1 \leq \hat{\kappa} \leq 2M$. So an extended edge type $\hat{E}^{\hat{\kappa}}$ is either a forward E^{κ} or inverse edge type $E^{\kappa T}$. The edge type hierarchy naturally extends to extended edge types, where $\hat{E}^{\hat{\kappa}} \leq_{\hat{\mathcal{E}}} \hat{E}^{\hat{\kappa}'}$ iff

1. $\hat{E}^{\hat{\kappa}}$ and $\hat{E}^{\hat{\kappa}'}$ are both forward edge types $E^{\kappa}, E^{\kappa'}$, and $E^{\kappa} \leq_{\mathcal{E}} E^{\kappa'}$; or
2. $\hat{E}^{\hat{\kappa}}$ and $\hat{E}^{\hat{\kappa}'}$ are both reverse edge types $E^{\kappa T}, E^{\kappa' T}$, and $E^{\kappa T} \leq_{\mathcal{E}} E^{\kappa' T}$.

Note that if $\hat{E}^{\hat{\kappa}}, \hat{E}^{\hat{\kappa}'}$ are forward and reverse (or reverse and forward), then $\hat{E}^{\hat{\kappa}} | E^{\hat{\kappa}'}$.

We can similarly extend to individual edges, letting $\hat{E} = \bigcup_{\kappa=1}^{M} (E^{\kappa} \cup E^{\kappa T})$ be the set of extended edges, so that an individual extended edge is $\hat{e} \in \hat{E}$. We also have an extended edge type function $\hat{t}_{\mathcal{E}} \colon \hat{E} \to \hat{\mathcal{E}}$, so that e.g. if $E^{\hat{\kappa}} = E^{\kappa T}$ is the inverse of a edge type $E^{\kappa} \subseteq V^{\iota} \times V^{\iota'}$, then $\hat{t}_{\mathcal{E}}\left(\hat{e}_k^{\hat{\kappa},\iota,\iota'}\right) = \left\langle V^{\iota'}, V^{\iota} \right\rangle$.

A **path type** P **of length** η is an η-long string in $\hat{\mathcal{E}}^*$ (the free monoid over $\hat{\mathcal{E}}$, now seen as an alphabet of both the forward and inverse edge types), denoted $P = \left\langle \hat{E}_j \right\rangle_{j=1}^{\eta}$ for $\hat{E}_j \in \hat{\mathcal{E}}$. Each element $\hat{E}_j \in P$ of the path type could be a forward or reverse edge type, so that for each $\hat{E}_j \in P$, there is a base edge type $E^{\kappa} \in \mathcal{E}$ so that $\hat{E}_j = E^{\kappa}$ or $\hat{E}_j = E^{\kappa T}$.

Let a **path type expression** (pathexp) π be a (limited) regular expression in $\hat{\mathcal{E}}$. Supported operators include

```
|, ( ), ^, $, {m,n}, *, ?, +
```

A path type P is **accepted** by the pathexp π if P is a word of π.

Consider two comparable (subsumptive) edge types $E'^{\kappa'} \leq_{\mathcal{E}} E^{\kappa}$. Then let $l\left(E'^{\kappa'}, E^{\kappa}\right)$ be the maximum chain length in the relation hierarchy $\mathcal{S}_{\mathcal{V}}$ connecting $E'^{\kappa'}$ upward to E^{κ}. We can then represent the **degree of subsumption** or **subsumptiveness** between them as

$$\sigma\left(E'^{\kappa'}, E^{\kappa}\right) = \alpha^{l\left(E'^{\kappa'}, E^{\kappa}\right)}, \tag{1}$$

where $\alpha \geq 1$ is a free parameter to be tuned. α specifies the extent to which subsumption is to be taken into account. If $\alpha = 1$, then subsumption is not considered, whereas the more that $\alpha > 1$, the more subsumption is taken into account. By default, if not otherwise specified, set $\alpha = 2$.

Note that thereby $\sigma(E'^{\kappa'}, E^{\kappa}) \geq 1$, with $\sigma(E'^{\kappa'}, E^{\kappa}) = 1$ iff $E'^{\kappa'} = E^{\kappa}$ or $\alpha = 1$, and that $\sigma(E'^{\kappa'}, E^{\kappa})$ increases as the max chain length between $E'^{\kappa'}$ and E^{κ} in the subsumption hierarchy grows, up to $\alpha^{\mathcal{H}}$ where $\mathcal{H}$ is the height of the relation hierarchy. Also note that if $E'^{\kappa'} | E^{\kappa}$, so that they are non-comparable (siblings or cousins), then

$$\sigma(E'^{\kappa'}, E^{\kappa}) = \infty \tag{2}$$

by convention.

Consider two path types P', P of the same length η. Then we say P' **is subsumed by** P when each edge type in P' inherits from each corresponding edge type in P:

$$P' \leq P \quad := \quad \forall 1 \leq j \leq \eta, \hat{E}'_j \leq_{\mathcal{E}} \hat{E}_j.$$

If all these corresponding types are equal, then we say that the path type P is **type-equal** to path type P', and denote that as

$$P' \simeq P \quad := \quad \forall 1 \leq j \leq \eta, \quad \hat{E}'_j = \hat{E}_j.$$

If one path type P' is subsumed by another P, then we can measure it's **ontologically-adjusted length** $L(P'|P)$ relative to P as the total subsumptiveness of the corresponding edge types:

$$L(P'|P) = \sum_{j=1}^{\eta} \sigma(\hat{E}'_j, \hat{E}_j). \tag{3}$$

Note that $L(P'|P) \geq \eta$, with $L(P'|P) = \eta \leftrightarrow P' \simeq P$.

For a pathexp π, use $\hat{E}_j$ to denote the "atomic" extended edge type of its j'th element. Then we can say that a path type P' is subsumed by a pathexp π when each edge type in P' inherits from the atomic extended edge type it matches in the expression, or

$$P' \leq \pi \quad := \quad \forall 1 \leq j \leq \eta, \quad \hat{E}'_j \leq_{\mathcal{E}} \hat{E}_j.$$

If all these corresponding types are equal, then we say that the path type P' is type-equal to the pathexp π, and denote that as

$$P' \simeq \pi \quad := \quad \forall 1 \leq j \leq \eta, \quad \hat{E}'_j = \hat{E}_j.$$

The ontologically-adjusted length $L(P'|\pi)$ of a path type P' relative to a pathexp π can then be defined exactly as in (3), using our abusive notation for $\hat{E}_j \in \pi$, so that now $L(P'|\pi) = \eta \leftrightarrow P' \simeq \pi$.

Finally, define a **path** p **of type** P as a vector $p := \langle A_j \rangle_{j=1}^{2\eta+1}$ of length $2\eta + 1$, where η is the length of P. For j odd, A_j is a node v^{ι}, and for j even, A_j is an extended edge $\hat{e}^{\hat{\kappa},\iota,\iota'}$. For a path p, define its type as $t_P(p) = P$, and also denote this as p^P. A path p^P must be **consistent** with its path type P and the ontology $\mathcal{O}$, in that $\forall 1 \leq j \leq \eta$, we have

$$\hat{t}_{\hat{\mathcal{E}}}(A_{2j}) = \hat{E}_j = \langle t_{\mathcal{V}}(A_{2j-1}), t_{\mathcal{V}}(A_{2j+1}) \rangle ,$$

or in English, the type of the extended edge at position $2j$ in the path must be the same as the type of the extended edge at position j in the path type, and the node types of the nodes at positions $2j - 1$ and $2j + 1$ in the path must be the appropriate node types of the edge at position $2j$.

3 Typed Path Queries

Fix two nodes v, v'. Then we can define a **subsumptive typed path query**, or just "path query", as a structure $\mathcal{Q} := \langle \pi, v, v' \rangle$ where π is a query pathexp. A path query $\mathcal{Q}$ requests paths p^P of type $P = t_P(p^P)$ for which:

- $A_1 = v$;
- $A_{2\eta+1} = v'$;
- There exists a path type P' such that,
  - P' is accepted by the query pathexp π; and
  - P' subsumes P, $P \leq P'$.

For a path query $\mathcal{Q}$, let $\mathcal{P} = \{p_l\}_{l=1}^{\mu}$ be the set of μ paths returned.

Continuing our example, we begin with the extended edge types:

$$\hat{\mathcal{E}} = \{\text{Owns-Place, Property-Of, Owns-Phone, Phone-of, Lives-At, Above-Of, Located-At, Location-Of,} \\ \text{Owns, Owned-By, Relation, Relation}^T\}$$

There are many syntactic and notational options for pathexps. Referring to Table 1, we will use the edge type symbol for the "forward" direction, and the transpose superscript $\cdot^T$ for the "inverse" direction.

Consider the pathexp $\pi^1 = \texttt{O}^\texttt{T}\ \texttt{O}\ \texttt{LI}^\texttt{T}$, and path query $\mathcal{Q}^1 = \left\langle \pi^1, s, g \right\rangle$. Referring to Fig. 2, in English we're asking for all the typed paths from Select Gourmet Foods to Muktar Galab which match to Owned-By, then Owns, and finally Abode-Of. $\mathcal{Q}_1$ returns $\mu = 1$ path

$$\mathcal{P} = \{p_1 = \left\langle s, O^T, m, PL, u, LI^T, g \right\rangle\}$$

with $\eta = 3$ and path type $P^1 := t_P(p_1) = \left\langle O^T, PL, LI^T \right\rangle$. p_1 matches our path query $\mathcal{Q}^1$, because $A_1^1 = s, A_{2\eta+1}^1 = A_7^1 = g$, the path type $P^1 = \left\langle O^T, PL, LI^T \right\rangle \leq \left\langle O^T, O, LI^T \right\rangle = P'^1$ (because $PL \leq_{\mathcal{E}} O$), and P'^1 is accepted by our pathexp π^1.

Now relax the search criteria slightly to $\mathcal{Q}^2 = \left\langle \texttt{O}^\texttt{T}\ \texttt{R\{1,3\}}\ \texttt{LI}^\texttt{T}, s, g \right\rangle$, or in English, all the paths from Select Gourmet Foods to Muktar Galab whose first edge is still Owned-By, and whose last edge is still Abode-Of, but with at least one, and at most three, intervening edges of any types. Then our return set has $\mu = 2$ paths

$$\mathcal{P} = \{p_1 = \left\langle s, O^T, m, LI, u, LI^T, g \right\rangle, p_2 = \left\langle s, O^T, m, PL, u, LI^T, g \right\rangle\}.$$

Finally consider the path query $\mathcal{Q}^3 = \left\langle (\texttt{LI}|\texttt{LI}^\texttt{T})+\ (\texttt{LI}|\texttt{LI}^\texttt{T})+, f, g \right\rangle$, or in English, all the paths from Floyd Ave. to Muktar Gulab involving chains of at least two Lives-At or Abode-Of edges. There's again $\mu = 1$ match $\mathcal{P} = \{p_1 = \left\langle f, LI^T, m, LI, u, LI^T, g \right\rangle\}$.

4 Typed Path Distance

We seek a semi-metric function $d\colon V^2 \to [0, \infty]$, so that $d(v, v') = d(v', v)$ and $d(v, v') = 0 \leftrightarrow v = v'$.

Assume a **library** of pathexps $\Pi = \{\pi\}$. Fix two nodes $v^\iota, v'^{\iota'} \in V$, with types ι, ι' respectively. For each pathexp $\pi \in \Pi$ in our library, let $\mathcal{P}(\pi, v, v')$ be the μ paths p returned from the path query $\mathcal{Q} = \langle \pi, v, v' \rangle$.

Definition 4 (Path Expression Adjusted Minpath) For a path query $\mathcal{Q} = \langle \pi, v, v' \rangle$ with return set $\mathcal{P}(\pi, v, v')$, let

$$M_\pi(v, v') := \begin{cases} 0, & v = v' \\ \mathrm{Min}_{p \in \mathcal{P}(\pi,v,v')} L(t_P(p)|\pi), & \mu > 0, v \neq v' \\ \infty, & \mu = 0, v \neq v' \end{cases}$$

be the minimum length of its returned paths adjusted to the path query $\mathcal{Q}$.

In English, $M_\pi(v, v')$ is the minimum of the ontologically-adjusted path lengths of the μ paths p conformant with the query path π; zero if the nodes are identical; or infinite if there are no return paths between distinct nodes.

Theorem 5 $M_\pi(v, v') = 0 \leftrightarrow v = v'$.

Proof: It is sufficient to show for $\mu > 0, v \neq v'$, that $\forall p \in \mathcal{P}(\pi, v, v'), L(t_P(p)|\pi) > 0$. We know that $L(t_P(p)|\pi)$ is minimal at η iff $t_P(p) \simeq \pi$, so that if all returned paths are type-equal to π, and so $M_\pi(v, v')$ is just the minimum returned path type length. In turn, η is minimal at 1 iff there is at least one returned path which is a single link. Thus only under both conditions is $M_\pi(v, v')$ minimal at 1. ■

For a node pair v, v', we are interested in identifying those pathexps $\pi \in \mathcal{P}$ in the library which have some return paths, and those which do not. So define

$$\Pi_{v,v'} := \{\pi \in \Pi : \ M_\pi(v, v') < \infty\} \subseteq \Pi.$$

Definition 6 (Typed Path Directed Distance) Let $\hat{d}_\Pi \colon V^2 \to [0, \infty]$ be defined as

$$\hat{d}_\Pi(v, v') := \begin{cases} \frac{\sum_{\pi \in \Pi_{v,v'}} M_\pi(v,v')}{|\Pi_{v,v'}|}, & \Pi_{v,v'} \neq \emptyset \\ \infty, & \Pi_{v,v'} = \emptyset \end{cases}.$$

In English, $\hat{d}_\Pi(v, v')$ is the average, over those pathexps in the library which return paths, of the adjusted minpath between v and v', and infinite if there are no such pathexps.

Definition 7 (Typed Path Distance) Let $d_\Pi \colon V^2 \to [0, \infty]$ be defined as

$$d_\Pi(v, v') := \begin{cases} \mathrm{Min}\left(\hat{d}_\Pi(v, v'), \hat{d}_\Pi(v', v)\right), & \hat{d}_\Pi(v, v') \times \hat{d}_\Pi(v', v) = \infty \\ \frac{\hat{d}_\Pi(v,v') + \hat{d}_\Pi(v',v)}{2}, & \hat{d}_\Pi(v, v') \times \hat{d}_\Pi(v', v) < \infty \end{cases}.$$

In English, the typed path distance is the typed path directed distance in the one direction in which the nodes are connected, if they are connected in only one direction; and the average of those in both directions if they are connected in both directions.

Theorem 8 $d_\Pi(v, v')$ is a semi-metric. Moreover, $\hat{d}_\Pi(v, v')$ is infinite only if v and v' are not connected by any pathexps $\pi \in \Pi$ in either direction.

Proof: By inspection, d is symmetric. Positive definiteness follows from (5). The final clause follows because $M_p(v, v') < \infty$ if there is some return path for either the query $\langle p, v, v' \rangle$ of the query $\langle p, v', v \rangle$ for $v \neq v'$. ∎

Continuing our example, we have our library:

$$\Pi = \{\pi^1, \pi^2, \pi^3\} = \{\texttt{O}^\texttt{T}\ \texttt{O}\ \texttt{LI}^\texttt{T}, \texttt{O}^\texttt{T}\ \texttt{R}\{1,3\}\ \texttt{LI}^\texttt{T}, (\texttt{LI}|\texttt{LI}^\texttt{T}) +\ (\texttt{LI}|\texttt{LI}^\texttt{T})+\}.$$

We will illustrate two queries in the library:

1. $v = s, v' = g$: Consider the first query path $\pi^1 = \texttt{O}^\texttt{T}\ \texttt{O}\ \texttt{LI}^\texttt{T}$ with its return set

   $$\mathcal{P}^1 = \left\{ p_1^1 = \left\langle s, O^T, m, PL, u, LI^T, g \right\rangle \right\}$$

   and $\mu = 1$. The path type of the single returned path is $t_P(p_1^1) = P_1^1 = \left\langle O^T, PL, LI^T \right\rangle$ with $\eta = 3$. Since $PL < O$, we have $P_1^1 \leq \pi^1$. With $\alpha = 2$, and PL immediately below O in the relation hierarchy, we have $\sigma(PL, O) = 2$ for the second link in the path type, and thus the adjusted path length is $L(P_1^1|\pi^1) = 1 + 2 + 1 = 4$, even though $\eta = 3$. Since this is the only returned path, and $s \neq g$, we also have the adjusted minpath between s and g using π^1 as $M_{\pi^1}(s, g) = 4$.

   For the next query path $\pi^2 = \texttt{O}^\texttt{T}\ \texttt{R}\{1,3\}\ \texttt{LI}^\texttt{T}$, we have $\mu = 2$ returned paths with types $P_1^2 = \left\langle O^T, LI, LI^T \right\rangle$ and $P_2^2 = \left\langle O^T, PL, LI^T \right\rangle$, yielding $L(P_1^2|\pi^2) = 1 + 2 + 1 = 4$ and $L(P_2^2|\pi^2) = 1 + 4 + 1 = 6$, so that $M_{\pi^2}(s, g) = \min(4, 6) = 4$.

   Finally, considering $\pi^3 = (\texttt{LI}|\texttt{LI}^\texttt{T}) +\ (\texttt{LI}|\texttt{LI}^\texttt{T})+$, there are no return paths, so $M_{\pi^3}(s, g) = \infty$.

   Now referring to (6), we have $\Pi_{s,g'} = \{\pi^1, \pi^2\} \subseteq \Pi$, so that $\hat{d}_\Pi(s, g) = \frac{4+4}{2} = 4$.

   Looking next at $\hat{d}$ going the other direction, none of the queries $\left\langle \pi^l, g, s \right\rangle$ have any return paths for $l \in [[3]]$, so $\hat{d}_\Pi(g, s) = \infty$, and from (7) we have $d_\Pi(s, g) = \min(\hat{d}_\Pi(s, g), \hat{d}_\Pi(g, s)) = \min(4, \infty) = 4$.

2. $v = f, v' = g$: Now $\mathcal{P}(\pi^1, f, g) = \mathcal{P}(\pi^2, f, g) = \emptyset$, so that $M_{\pi^1}(f, g) = M_{\pi^2}(f, g) = \infty$. But for $\pi^3 = (\texttt{LI}|\texttt{LI}^\texttt{T}) +\ (\texttt{LI}|\texttt{LI}^\texttt{T})+$, we have a single path match in each direction, so that $\Pi_{f,g} = \Pi_{g,f} = \{\pi^3\}$. We have $M_{\pi^3}(f, g) = M_{\pi^3}(g, f) = 3$, since the match is precise over path type length $\eta = 3$, yielding $\hat{d}_\Pi(f, g) = \hat{d}_\Pi(g, f) = 3$, and finally $d_\Pi(f, g) = 3$.

5 Algorithm

Algorithm 1 Typed path directed semi-metric $\hat{d}_\Pi(v, v')$.

INPUT: Library of path type expressions Π, e.g.:

$$\Pi = \{\pi^1, \pi^2, \pi^3\} = \{\mathtt{O}^\mathtt{T}\ \mathtt{O}\ \mathtt{LI}^\mathtt{T}, \mathtt{O}^\mathtt{T}\ \mathtt{R}\{\mathtt{1},\mathtt{3}\}\ \mathtt{LI}^\mathtt{T}, (\mathtt{LI}|\mathtt{LI}^\mathtt{T})+\ (\mathtt{LI}|\mathtt{LI}^\mathtt{T})+\}.$$

INPUT: A free parameter α for the amount of subsumptive "stretching", by default $\alpha = 2$.

INPUT: Two nodes of interest $v, v' \in V$, e.g. $\langle s, g\rangle$.

OUTPUT: Directed semi-metric $d_\Pi(v, v') \in [0, \infty]$.

```
if v = v' then
    return 0
end if
Let Π_{v,v'} ← ∅
Let d̂_Π(v, v') ← 0
for all path type expressions π ∈ Π in the library do
    Let 𝒫 be the set of subsumptively matching paths, e.g. for π^1 we have
        𝒫^1 = { p_1^1 = ⟨s, O^T, m, PL, u, LI^T, g⟩ } .
    Let μ be the number of subsumptively matching paths, e.g. for π^1 we have μ = 1
    if μ > 0 then
        Let M_π(v, v') ← ∞
        Let Π_{v,v'} ← Π_{v,v'} ∪ {π}
        for each of the paths returned, e.g. p_1^1 do
            Let P' be the path type of the returned path, e.g. P'^1 = ⟨O^T, PL, LI^T⟩.
            Let η be the length of P'.
            for L(P'|π) ← 0, j ← 1; j++; j ≤ η do
                Let Ê'_j be the edge type of the j'th edge in P'
                Let Ê_j be the edge type of the j'th edge in π
                Let l(Ê', Ê) be the length of the longest chain in the relation hierarchy from
                Ê' up to Ê
                Let L(P'|π) += α^{l(Ê', Ê)}
            end for
            Let M_π(v, v') ← min(M_π(v, v'), L(π'|π))
        end for
        Let d̂_Π(v, v') += M_π(v, v')
    end if
end for
if |Π_{v,v'}| = 0 then
    return ∞
else
    return d̂_Π(v, v') / |Π_{v,v'}|
end if
```

Algorithm 2 New typed path semi-metric $d_\Pi(v, v')$.

INPUT: Path type expression library Π.

INPUT: A free parameter α for the amount of subsumptive "stretching", by default $\alpha = 2$.

INPUT: Two nodes of interest $v, v' \in V$, e.g. $\langle s, g \rangle$.

OUTPUT: Semi-metric $d_\Pi(v, v') \in [0, \infty]$.

Let $d_1 \leftarrow \hat{d}_\Pi(v, v')$
Let $d_2 \leftarrow \hat{d}_\Pi(v', v)$
if $d_1 = \infty$ or $d_2 = \infty$ **then**
  return $\mathrm{Min}\,(d_1, d_2)$
else
  return $\frac{d_1 + d_2}{2}$
end if