Distributed Representations of Bio-Ontologies for Semantic Web Services

CA Joslyn*[1], DDG Gessler[2], SE Schmidt[3] and KM Verspoor[1]

[1]Computer Science Division, Los Alamos National Laboratory, PO Box 1663, Los Alamos, NM, 87545, USA
[2]National Center for Genome Resources, Santa Fe, NM 87505 USA
[3]Technische Universität Dresden, Germany

Email: CA Joslyn - joslyn@lanl.gov; DDG Gessler - ddg@ncgr.org; SE Schmidt - midt1@msn.com; KM Verspoor - verspoor@lanl.gov;

*Corresponding author

Abstract

We introduce the Semantic Moby/VPIN distributed refactoring of the Gene Ontology optimized for semantic web architectures. We show through an order-theoretical analysis that this alternative representation preserves the original topology, and through Formal Concept Analysis that additional information about ambiguous inheritance may be revealed by the reconstruction.

1 Introduction

Bio-ontologies such as the Gene Ontology (GO) [5] are increasingly important as researchers seek to standardize vocabulary and integrate results into a common framework. They are also being used as the basis for critical bioinformatics tasks, including function prediction [9] and protein family classification [6]. These ontologies are often large (the GO currently contains on the order of 20,000 concepts), making them unamenable for semantic web architectures that need rapid access to only a few terms across many ontologies. Additionally, the common practice of building ontologies with deep, fixed subsumption assertions (e.g., successive, nested `owl:subClassOf` assertions) means that creating these ontologies is a low-throughput, labor intensive task, yielding third-party extensions vulnerable to fragility and rigidity.

We seek an alternative approach that retains the high value present in extant static ontologies, while refactoring them into distributed representations more appropriate for a semantic web services infrastructure. The advantages include supporting dynamic integration, querying of ontological terms from distinct ontologies, and avoiding massive redundancy in distributed ontological representation. In this paper we first describe a distributed representation of GO within the framework of the Virtual Plant Information Network (VPIN)[1] and the SemanticMoby effort [7][2]. This mapping produces an "individual/property-centric" rather than a "subsumption-centric" model, allowing inference of subsumption assertions on demand. This can be used to generate or expand ontologies *de novo*.

But such distributed representations also come with risks and costs. In particular, it's crucial that the original ontology be easily reconstructible from the decomposed pieces in a lossless manner. We can use order theory [8] to demonstrate that this distributed structure preserves the original GO in a lossless manner. Moreover, we can use Formal Concept Analysis (FCA) [4] to produce a reconstruction which may reveal additional information not present in the original GO, in particular allowing for the unique resolution of potential ambiguities of inheritance.

[1]http://vpin.ncgr.org
[2]http://www.semanticmoby.org

2 The Semantic Moby/VPIN GO Representation

Viewed ontologically, the GO has very few properties (or "predicates" in RDF terminology) from which a reasoner could *infer* subsumption amongst classes. Rather, subsumption is *asserted* explicitly with `is-a` relationships, such as "cellular process `is-a` biological process". In order to produce a knowledge structure more amenable to semantic web applications, we seek a distributed refactoring of the GO which allows all GO subsumption relations to be determined, yet minimizes explicit static subsumption statements.

We begin by replacing the subsumption-centric representation of GO with an individual/property-centric representation. Let P be the set of all GO nodes, and for nodes $A, B, C \in P$, consider that we have C `is-a` B and B `is-a` A, and that A is the root of the hierarchy. We could represent this ontologically as the subsumption statements "C is a sub-class of B" and "B is a sub-class of A", or mathematically as $C \subseteq B \subseteq A$. In the spirit of REpresentational State Transfer architecture [3], we replace the representation of A, B, C as classes with their representation as instantiated individuals, members of a single class P, resources which reside at URLs. We then introduce a generic property `superProperty` whose domain and range are $P \cup \{\texttt{null}\}$, where `null` may optionally be used to signify no individual. The semantics of `superProperty` are that its object (right-hand side) is an individual which is a member of a super-class of the class of the subject (left-hand side). We thus assert `C superProperty B`, `B superProperty A`, `A superProperty null`, where the triple is read as subject, predicate, object.

We extend the same convention exactly one level down the hierarchy by introducing the property `subProperty`, and define it dually. We only include the immediate children, since listing all successors for the root would reproduce the entire ontology in the root's definition. It is often desirable to know an individual's root class directly (e.g., Biological Process), instead of following a long chain of `superProperty` statements. We therefore finally introduce a property `rootProperty` for all individuals that points to the root of the DAG, e.g. `A rootProperty A`, `B rootProperty A`, etc.

Note that we specifically wish to avoid static subsumption statements as in the use of `owl:subClassOf`, or by naming our property something like `hasParent`. Rather, we wish to encode the necessary information to derive subsumption dynamically. Also, the statement `C superProperty A` could be inferred from `C superProperty B` and the definition of B, but making all `superProperty` statements explicit in the definition of C allows one to build complete definitions within a single file. This has significant advantages in semantic web services architectures. Finally, solely reading the definition of C, one cannot determine if $B \subseteq A$ or $A \subseteq B$. This is an important encapsulation, because the only explicit statements about C that should appear in its definition are those where C is the subject.

Our final definitions are therefore:

```
A: A rootProperty A      B: B rootProperty A      C: C rootProperty A
   A subProperty B          B superProperty A        C superProperty B
                            B subProperty C          C superProperty A
```

The Semantic Moby/VPIN (SMV) refactoring of the GO is available at `http://ontologies.ncgr.org`, an implementation of the abstract Open Biomedical Ontology (OBO) standard. Thus we define the abstract classes for OBO at `http://ontologies.ncgr.org/OpenBiomedicalOntologies` with the extension that includes the properties `superProperty`, `rootProperty`, and `subProperty`. We then map these into specific terms for the Gene Ontology at `http://ontologies.ncgr.org/GeneOntology`. Finally, we map each GO concept into an individual, whose definition files are available at

`http://ontologies.ncgr.org/GeneOntology/<ontology>/<GO_id>`,

where `<ontology>` is one of BiologicalProcess, CellularComponent, or MolecularFunction, and `<GO_id>` is the GO ID of the term, e.g., `GO_0000001`. In this proof-of-concept implementation, only subsumptive `is-a` relations are included, but `has-part` is also transitive, and thus amenable to the same analysis.

3 Order Theoretical Representation and Reconstruction

The SMV distributed GO implementation has a natural mathematical representation in order theory, the theory of ordered sets and lattices [8]. In this section, we outline this representation, illustrate some of its features, and show how it demonstrates that the original GO can be reconstructed exactly from its SMV factors. Moreover, in the event that there is a portion of the GO with a branching structure complex enough to not allow identification of a least common subsumer, the methodology of FCA [4] will disentangle and disambiguate those connections and allow such identification.

3.1 Factor Reconstruction

Fig. 1 shows a collection of models of a hypothetical portion of the GO, all of which are Directed Acyclic Graphs (DAGs). The left side shows a set of GO nodes P as we typically encounter them in a graphical viewer. Nodes in P are GO categories, connected by arrows indicating `is-a` relations. This is a structure called the Hasse diagram of a partially ordered set (poset) $\mathcal{P}$ on the set of nodes P. Technically, we deal only with bounded, finite posets, and insert an inconsequential virtual bottom (here D) if lacking.

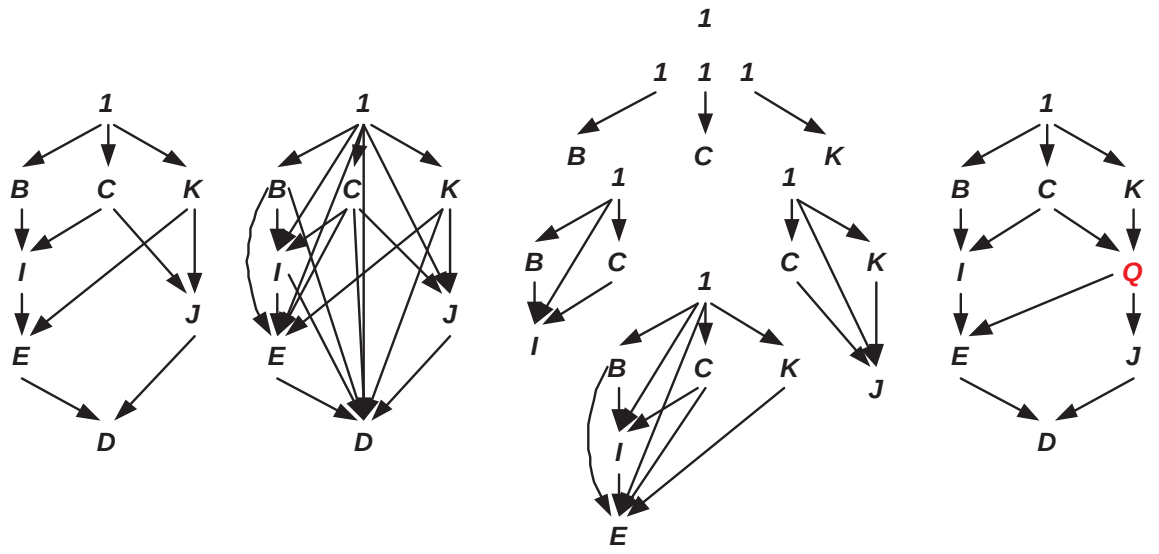

Figure 1: (Left) The Hasse diagram of a model of a GO portion. (Center Left) Transitive closure. (Center Right) Principle filters. (Right) Dedekind-MacNeille completion.

In the center left is the transitive closure of the Hasse diagram, which is a complete relational representation of the ordered set $\mathcal{P}$. This structure represents the inclusion of all transitive links, effectively what happens when the transitivity of the GO's "true path rule"[3] is followed through to completion, as in the SMV representation. In order theory, a "principle filter" of a node $X \in P$ in a poset $\mathcal{P}$ is a structure $\uparrow X$ consisting of the node X and all of its ancestors, all the way up to the root. In the example, the principle filter of J is the set $\uparrow J = \{C, J, K, 1\}$. The center right of Fig. 1 shows the collection of all the principle filters $\uparrow X, X \in P$ "exploded out", except $\uparrow D$, which is identical to the center left diagram.

In the SMV representation of the GO, the `superProperty` relation results in the storage of the principle filters $\uparrow X$ for each node $X \in P$ in the database. An important result from order theory is that the original structure of a poset $\mathcal{P}$ is completely recoverable from the principle filters of its atoms (here $\uparrow E$ and $\uparrow J$). Effectively, all that is required is unioning together the filters, and then constructing the transitive reduction [1], resulting in the original Hasse diagram. Thus the SMV representation of the GO is lossless.

[3] http://www.geneontology.org/GO.usage.shtml#truePathRule

	B	C	D	E	I	J	K	1
B	√							√
C		√						√
D	√	√	√	√	√	√	√	√
E	√	√		√	√		√	√
I	√	√			√			√
J		√				√	√	√
K							√	√
1								√

Table 1: The formal context of our example.

3.2 Lattice Completion

Given two GO classes A and B, what do they have in common? Technically, this is the question of calculating their least common subsumer (LCS) [2], a critical and foundational ontolgy operation (e.g. "what is the common function of these two genes?"). In our example in Fig. 1, the LCS of I and J is C. But what about E and J? Here there are two possible LCSs, C and K. In general in bounded posets, not all pairs of nodes need have unique LCSs, but if they do, then the poset is called a lattice.

So while we generally prefer ontologies to be lattices, and not proper posets, if a GO portion happens to be one, then the SMV GO representation provides a solution through the use of FCA [4]. In particular, the `superProperty` properties of the SMV entries for each node $X \in P$ determine a row in a matrix in $P \times P$, called a formal context, where a cell in an X row indicates an ancestor of X. Our example formal context is shown in Tab. 1. FCA then provides a canonical method to calculate a lattice which precisely represents the context, even if the original ontology was not a lattice. Technically, this is called calculation of the Dedekind-MacNeille completion (Theorem 4 in [4]), and the example is shown on the far right of Fig. 1. Notice the inclusion of Q as the new unique parent of E and J, our two nodes lacking an LCS. Q thus acts as a placeholder for whatever is held in common between C and K, from which it multiply inherits.

4 Conclusion

The SMV represention of large taxonomic ontologies such as the GO is distributed, flexible, and conducive to semantic web architectures, and together with our order theoretical analysis promises significant advances. For example, the lattice-like properties of the GO remains an uninvestigated empirical question: are LCSs always available in the GO? If not, to what extent is it a proper poset and not a lattice, and can we identify the offending portions? The SMV representation allows this calculation to be performed relatively easily, constructing the formal context directly from the database entries, and subtracting the GO from its FCA reconstruction. Moreover, our approach points the way for the use of distributed semantic web implementations and order theoretical technology in other ontology tasks such as induction.

Partial support for this work was provided by NSF BD&I grant 0516487.

References

1. Aho, AV; Garey, MR; and Ullman, JD: (1972) "The Transitive Reduction of a Directed Graph", *SIAM Journal of Computing*, v. **1**:2, pp. 131-137
2. Baader, Franz; Sertkaya, Baris; and Turham, Anni-Yasmi: (2004) "Computing the Least Common Subsumer w.r.t. a Background Terminology", in: *Proc. JELIA 2004, Lecture Notes in AI*, v. **3229**, pp. 400-412
3. RT Fielding: (2000) *Architectural Styles and the Design of Network-based Software Architectures*, PhD Dissertation, UC Irvine
4. Ganter, Bernhard and Wille, Rudolf: (1999) *Formal Concept Analysis*, Springer-Verlag
5. Gene Ontology Consortium: (2000) "Gene Ontology: Tool For the Unification of Biology", *Nature Genetics*, v. **25**:1, pp. 25-29
6. AG Maguitman, A Rechtsteiner, KM Verspoor, CE Strauss, and LM Rocha: (2006)"Large-Scale Testing Of Bibliome Informatics Using Pfam Protein Families", *Pacific Symposium on Biocomputing*, 11:76-87.
7. G Schiltz, D Gessler, L Stein: (2004) "Semantic MOBY", Position Paper for the W3C Workshop on Semantic Web for Life Sciences, W3C, `http://lists.w3.org/Archives/Public/public-swls-ws/2004Sep/att-0036/smoby-w3c-sw-ls.pdf`
8. Schröder, Bernd SW: (2003) *Ordered Sets*, Birkhauser, Boston
9. Verspoor, KM; Cohn, JD; Mniszewski, SM; and Joslyn, CA: (2006) "Categorization Approach to Automated Ontological Function Annotation", *Protein Science*, 15:6, pp. 1544-1549