



Large-scale, Exhaustive Lattice-based Structural Auditing of SNOMED CT

Journal:	<i>AMIA 2010 Annual Symposium</i>
Manuscript ID:	AMIA-1620-A2009
Manuscript Type:	Paper
Date Submitted by the Author:	15-Mar-2010
Complete List of Authors:	Zhang, GQ; Case Western, Center for Clinical Investigation

Large-scale, Exhaustive Lattice-based Structural Auditing of SNOMED CT

GQ Zhang¹, PhD and Olivier Bodenreider², MD, PhD
¹Case Western Reserve University, Cleveland, OH 44106
²National Library of Medicine, Bethesda, MD 20892

Abstract

One criterion for the well-formedness of ontologies is that their hierarchical structure form a lattice. Formal Concept Analysis (FCA) has been used as a technique for assessing the quality of ontologies, but is not scalable to large ontologies such as SNOMED CT. We developed a methodology called **Lattice-based Structural Auditing (LaSA)**, for auditing biomedical ontologies, implemented through automated SPARQL queries, in order to *exhaustively identify all non-lattice pairs in SNOMED CT*. The percentage of non-lattice pairs ranges from 0 to 1.66 among the 19 SNOMED CT hierarchies. Preliminary manual inspection of a limited portion of the >518K non-lattice pairs, among over 34 million candidate pairs, revealed inconsistent use of precoordination in SNOMED CT, but also a number of false positives. Our results are consistent with those based on FCA, with the advantage that the LaSA computational pipeline is scalable and applicable to ontological systems consisting mostly of taxonomic links.

1 Introduction

Auditing of large ontological systems is an indispensable part of the ontology life-cycle [1]. One criterion for the well-formedness of ontologies is that their hierarchical structure form a lattice [2]. Simply speaking, a lattice is a structure in which two concepts do not share more than one least common ancestor. Lattice fragments are pervasive in SNOMED CT. For example, in Figure 1, the concepts “Transsphenoidal hypophysectomy” and “Partial hypophysectomy” share only one least common ancestor, “Hypophysectomy” and illustrate a lattice-conforming structure, or a *lattice pair*.

There are also many non-lattice fragments in SNOMED CT. In Figure 2, the concept pair (double circled) “Partial excision of pituitary gland by transfrontal approach” and “Partial excision of pituitary gland by transsphenoidal approach” is an example of a *non-lattice pair*, because these two concepts share more than

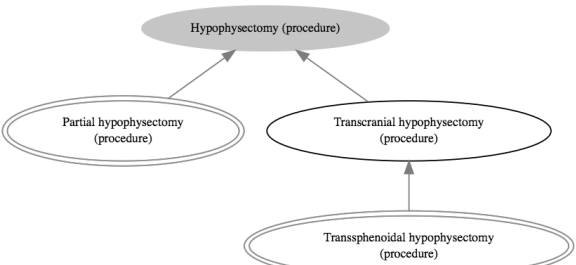


Figure 1: A lattice fragment in SNOMED CT's Hypophysectomy sub-hierarchy. The double-circled nodes share a unique least common ancestor.

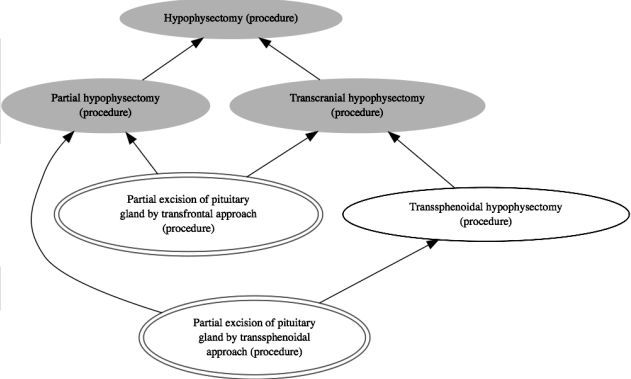


Figure 2: A non-lattice fragment in SNOMED CT's Hypophysectomy sub-hierarchy. The double-circled nodes share more than one least common ancestor.

one least common ancestor (namely “Partial hypophysectomy” and “Transsphenoidal hypophysectomy”).

The present study is motivated in part by the recent work of Jiang and Chute [3], who used Formal Concept Analysis (FCA [4]) as an auditing tool by constructing contexts from normal form presentations in SNOMED CT and analyzed the resulting lattice hierarchy for unlabeled nodes. They showed that this method can automatically identify a candidate pool of missing concepts for further examination by domain experts. However,

constructing lattices from contexts is so computationally expensive [5] that it is not scalable. FCA is therefore not applicable to the entirety of large ontologies such as SNOMED CT.

We introduce **Lattice-based Structural Auditing (LaSA)**, a methodology for auditing large biomedical ontologies. LaSA complements FCA-based and other existing approaches to ontological auditing by taking the lattice-property directly as a structural principle. The main contribution of this study is to demonstrate the applicability of lattice-based methods to an entire, large biomedical terminology: SNOMED CT. Our results are consistent with those based on FCA, with the advantage that the LaSA computational pipeline is scalable and applicable to ontological systems consisting mostly of taxonomic links.

2 Background

Our methodology draws on background knowledge from three areas. One is lattice theory, which serves as the *mathematical* and *algorithmic* foundation for LaSA. The second is the Resource Description Framework (RDF) and its associated SPARQL query language which is used for implementing LaSA and obtaining the experimental results presented in Section 4. The third area is SNOMED CT. We provide a concise overview of these areas to prepare for the development described in the subsequent sections.

2.1 Lattice

We first review some elementary concepts from lattice theory. Our main references will be [6, 7]. A partially ordered set (poset) is a set L with a reflexive, symmetric, transitive relation $\sqsubseteq \subseteq L \times L$. If (L, \sqsubseteq) is a poset, then its dual is the poset (L, \supseteq) . We denote posets by their carrier set as long as the partial order is clear from the context. A poset L is a *complete lattice* if every subset $S \subseteq L$ has a least upper bound $\bigvee S$ (join) and a greatest lower bound (meet) $\bigwedge S$. A poset L is a *lattice* if every two elements of L have a join and a meet. These meets and joins of binary sets will be written in infix notation: $\bigvee\{x, y\} = x \vee y$ and $\bigwedge\{x, y\} = x \wedge y$.

In connection with ontology, one can think of concepts as elements of a poset, and the ordering relation as the subsumption relation [8, 9]. If x, y are concepts, we write $x \sqsubseteq y$ to mean x IS-A y , or y subsumes x . The join $x \vee y$ of two concepts x, y is the lowest common ancestor of x and y , and the meet $x \wedge y$ is their greatest common descendant.

Every finite lattice is a complete lattice. Therefore, every finite lattice has a top (largest) element and a bottom (least) element. One can think of a tree as a lattice

by adjoining a superficial bottom. In this sense, lattices are more general than trees, but posets are more general than lattices. Multiple inheritance is not permitted in trees: each node in a tree can have at most one parent (the node immediately above it). Lattices permit multiple inheritance but insist on the existence of (unique) join and meet for any pair of nodes.

2.2 RDF and SPARQL

The Resource Description Framework (RDF) is a directed, labeled graph data format for representing information in the Web [10]. Based on (subject, predicate, object) triples, RDF is well suited for the representation of graphs in general, including posets and lattices. Because of its origins in the Semantic Web, RDF uses Unified Resource Identifiers (URIs) as names for the nodes and the links in the graph.

SPARQL is a query language for RDF graphs [11]. SPARQL queries are expressed as constraints on graphs, and return RDF graphs or sets as results. For example, SPARQL can be used for retrieving the set of common ancestors of two nodes in a graph.

2.3 SNOMED CT

SNOMED CT is a comprehensive concept system for healthcare developed by the International Health Terminology Standard Development Organization (IHTSDO) [12]. SNOMED CT provides broad coverage of clinical medicine, including findings, diseases, and procedures, and is used in electronic medical records [13]. The development of SNOMED CT is supported by an infrastructure based on description logics. From a structural perspective, SNOMED CT can be seen as a series of large directed acyclic graphs, one for each of its 19 “hierarchies”. No concept is shared across hierarchies. The version of SNOMED CT used in this study is dated July 31, 2009 and comprises 307,754 active concepts.

3 Methods

LaSA exhaustively checks concept pairs for conformation to the requirement of being a part of a lattice. For each pair of concepts a and b , we find all their common ancestors. Among all their common ancestors, the minimal ones must be unique, to conform to the mathematical definition of a lattice, in which every pair of elements must have a (unique) *least common ancestor*.

Our method for identifying non-lattice fragments involves three steps: 1) acquiring SNOMED CT data; 2) selecting probes; 3) testing probes.

3.1 Acquiring SNOMED CT data

From the distribution of SNOMED CT, we extracted all the IS-A relations among active concepts. We created URIs for all SNOMED CT concepts and used the *rdfs:subClassOf* predicate to represent the IS-A relation. Then we computed the transitive closure of the IS-A relation and created a distinct set of triples for it. The two sets of triples were loaded into two separate graphs using the open source Virtuoso triple store [14].

3.2 Selecting probes

Not every pair of SNOMED CT concepts needs to be tested for its lattice properties. Since the 19 hierarchies in SNOMED CT do not share any concepts, only pairs within the same hierarchy require testing. Moreover, pairs in which each concept does not have at least two parent concepts cannot form a non-lattice structure. Finally, any pair in which one concept is an ancestor of the other does not need to be tested because the ancestor concept in such pairs is the unique common ancestor of the two, with reflexivity assumed.

More formally, a pair of concepts a, b is called a *probe* if a and b are not in hierarchical relationship to each other (i.e. $a \not\sqsubseteq b$ and $b \not\sqsubseteq a$), and each node has at least two direct parents.

3.3 Testing probes

The following simple algorithm finds the minimal common ancestors of a, b by counting the instances of two specific cases where a node is situated as a part of the common ancestor subgraph. The key insight, however, is the idea of keeping track of appropriate counts.

Data: Transitivity closed RDF-triple store and probe (a, b)
Output: The minimal common ancestors of a, b

- 1 Set *count* to 0 for each node;
- 2 **if** $?sb$ is a common ancestor of a, b **then**
- 3 | increment *count*($?sb$) by 1
- 4 **end**
- 5 **if** $?sb$ is an ancestor of a common ancestor $?sa$ of a, b **then**
- 6 | increment *count*($?sb$) by 1
- 7 **end**
- 8 Sort $?sb$ in ascending order according to its *count*;
- 9 Mark all $?sb$ with *count*($?sb$) = 1 as minimal common ancestor

Algorithm 1: Finding minimal common ancestors

To implement Algorithm 1, we construct two SPARQL subqueries, corresponding to two RDF-subgraphs. Both subqueries are executed against a

transitively-closed graph of IS-A relations.

The first subquery (Figure 3) finds all common ancestors and tracks the result by having each common ancestor to receive 1 as the *counts*. This is straightforward using the SPARQL query

```
a rdfs:subClassOf ?sb
b rdfs:subClassOf ?sb
```

to get hold of all nodes $?sb$ that are ancestors of a and b .

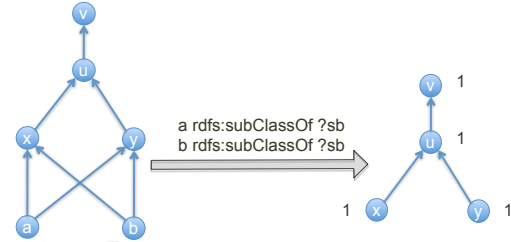


Figure 3: SPARQL query finding all common ancestors of a, b , with each such ancestor receiving count 1.

The second subquery (Figure 4) finds all common ancestors that are *ancestors among the common ancestors* – those nodes $?sb$ that are above a common ancestor $?sa$. This can be achieved by the SPARQL query

```
a rdfs:subClassOf ?sa
b rdfs:subClassOf ?sa
?sa rdfs:subClassOf ?sb
```

The counts for $?sb$ here keep track of the in-degrees of each node in the common-ancestor subgraph.

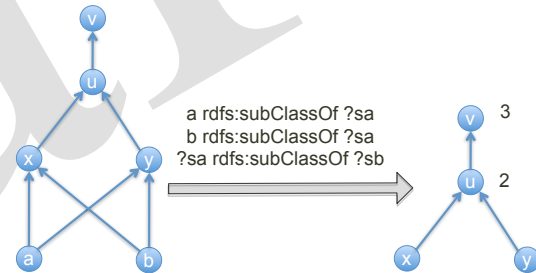


Figure 4: SPARQL query finding all common ancestors of a, b , with each such ancestor receiving count 1.

Putting the two subqueries together, one can see that nodes x and y receive a total count of 1, respectively; they are the minimal common ancestors of a and b .

LaSA separates two kinds of probes: those with a unique minimal common ancestor, and those with more than one minimal common ancestor. The latter is an indication of a non-lattice fragment generated by the current probe (called a “BAD” pair).

3.4 Implementation

The algorithms presented earlier for selecting the probes to be tested, and testing them, were implemented with-

Hierarchy	ID	TP	PP	NL	PP%	NL%	T(ms)
Procedure	71388002	1530566128	65800235	174574	4.3	0.26	31
Clinical Finding	404684003	5103176851	243428748	251662	4.8	0.1	7
Body Structure	123037004	490111086	29934856	91787	6.1	0.31	11
Specimen	123038009	730236	53397	889	7.3	1.66	2

Table 1: Summary of results. ID - SNOMED CT id of the concept;

TP - the total number of pairs obtained by the formula $n \times (n-1)/2$, where n is the number of concept in the hierarchy;

PP - the total number of probe pairs from TP, according to the criteria given in Section 3.2;

NL - the total number of non-lattice pairs from PP;

PP% - the percentage of probe pairs among all pairs, i.e. PP/TP;

NL% - the percentage of non-lattice pairs among all probe pairs, i.e. NL/PP;

T(ms) - the average SPARQL query time of pairs in PP in milliseconds.

out any ad hoc programming. Generic queries were created for each algorithm and subpart thereof and loaded as stored procedure. We used a simple script to compute the cartesian product of all pairs of concepts within a given hierarchy of SNOMED CT. Through this script, each pair was evaluated as a potential probe (by querying a stored procedure instantiated with the pair). If qualified as a probe, the pair was then tested for its lattice properties by querying a second stored procedure. The results were stored in text files for further processing. The open source Virtuoso RDF store version 06.00.3123 was used for this experiment, running on a Dell 2950 server (Dual Xeon processor) with 32GB of memory. A total of 500,000 9kB buffers were allocated to Virtuoso.

4 Results

4.1 Quantitative results

From the 307,754 active concepts in SNOMED CT, we created RDF triples for representing IS-A relations. The graph of direct hierarchical relations contains a total of 439,733 `rdfs:sibClassOf` triples, while the transitively-closed graph contains 1,191,796 triples.

Table 1 summarizes the our SPARQL query results based on the method described in Section 3 for 4 of the larger hierarchies in SNOMED CT: Procedure, Clinical Finding, Body Structure and Specimen. For example, 4.3% of probe pairs were found among all possible pairs in Procedure, and among the probing pairs, 0.26% were found to be non-lattice pairs.

Even though the numbers of non-lattice pairs seem large, they represent a very small percentage. For example, for Procedure, only 0.01% of pairs are non-lattice pairs, among all possible pairs within the hierarchy; for Clinical Finding, it is only 0.005%.

4.2 Evaluation

We ran our LaSA algorithm on the Hypophysectomy sub-hierarchy of SNOMED CT, the running example included in Jiang and Chute [3], using the same version of SNOMED CT as was used in their study, and performed a systematic comparison of our respective results.

Jiang and Chute found five anonymous nodes using FCA; we also found exactly five non-lattice pairs. Three non-lattice pairs correspond exactly to the three anonymous nodes (1, 3 and 4 in Table 3, page 95 of [3]), with a perfect match between our non-lattice pair and the extensions constructed using FCA.

The two remaining non-lattice pairs are part of the extension for Node 5. However, we found no non-lattice counterpart for Node 2. Upon closer inspection, the extension of Node 2 in Jiang and Chute represents a case of anonymous node without any lattice-violation. The creation of the intermediary node “Transfrontal hypophysectomy” would be justified only because several subtypes of excision of the pituitary gland by transfrontal approach are described, not to create a unique lowest common ancestor for these subtypes of hypophysectomy. Our findings are therefore consistent with that of Jiang and Chute, but FCA identified one anonymous node, which LaSA cannot identify.

5 Discussion

There is a close relation between precoordination and the lattice properties of the terminology. Two sibling concepts denoting multiple features (including two common features) will form a lattice only if their lowest common ancestor represents both features (and will not form a lattice if each common feature is represented by a distinct ancestor). As can be seen from Figure 5, the double-circled concepts “Tissue specimen from breast” and “Tissue specimen from heart” share the two features of being a kind of tissue specimen and of origin trunk.

The current representation in SNOMED CT involves two edges shared by the two double-circled concepts to “Tissue specimen” and to “Specimen from trunk”, which is the reason why this fragment is not a lattice. In order to transform this fragment into a lattice, a new concept “Tissue specimen from trunk” would need to be created, which would be the direct descendant of the current lowest common ancestors and would become a new unique lowest common ancestor of “Tissue specimen from breast” and “Tissue specimen from heart”.

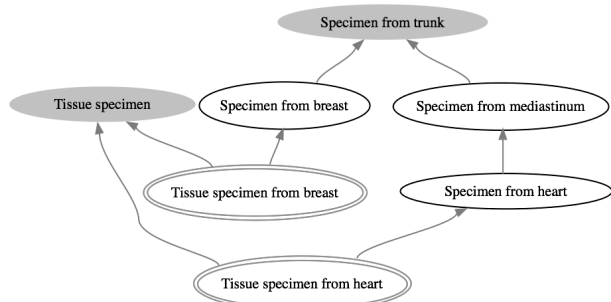


Figure 5: Non-lattice pair from the Specimen hierarchy.

Beyond the technicalities (i.e., the structural properties of the terminology), one question for SNOMED CT is how much precoordination is needed in clinical applications. The concept “Tissue specimen from trunk” is a valid concept, but it is unclear whether such intermediary nodes would be useful to users. (“Tissue specimen from mediastinum” would be a legitimate candidate as well). On the one hand, having many precoordinated terms would reduce the need for having to deal with post-coordination, which is nontrivial for most users. On the other, tens of thousands of such precoordinated nodes are likely to be needed to transform SNOMED CT into a lattice, which comes at a cost in terms of maintenance (for the developers) and in terms of increase volume for the users.

From a quality assurance perspective, what is important is to ensure that precoordination is used consistently in SNOMED CT, so as to facilitate usage. This, for example, would be an argument in favor of the creation of a concept “Transfrontal hypophysectomy”, as suggested by Jiang & Chute, in order to mirror, e.g., “Transsphenoidal hypophysectomy”.

In summary, LaSA produces results consistent with the FCA-based approach of Jiang & Chute, without suffering from their computational scalability problem. Additionally, unlike FCA, LaSA is applicable to ontological systems consisting mostly of taxonomic links. Among the methods developed for quality assurance in SNOMED CT, LaSA has been effective in identi-

fying (potentially) missing precoordinated concepts in SNOMED CT. A model of desirable (vs. excessive) precoordination should be developed, which could be used in conjunction with LaSA in order to determine which lattice violations are indicative of problems and require the attention of the editors of the terminology system. Meanwhile, we have shown that a lattice-based approach is applicable to large-scale terminologies and can be implemented with minimal programming effort.

Acknowledgements

This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM).

References

[1] Zhu X, Wei JW, Baorto D, Weng C, Cimino J. A review of auditing methods applied to the content of controlled biomedical terminologies. *J Biomedical Informatics*, 42:412-25, 2009.

[2] Zweigenbaum P, Bachimont B, Bouaud J, Charlet J, Boisvieux JF. Issues in the structuring and acquisition of an ontology for medical language understanding. *Methods Inf Med*. 1995 Mar;34(1-2):15-24.PMID: 9082125.

[3] Jiang G, Chute CG. Auditing the semantic completeness of SNOMED CT using formal concept analysis. *J Am Med Inform Assoc*. 2009 Jan-Feb;16(1):89-102.

[4] Ganter B, Wille R. *Formal Concept Analysis*. Springer-Verlag (1999).

[5] Kuznetsov SO, Obiedkov SA. Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Intell.* 14(2-3), pages 189-216 (2002).

[6] Gierz G, Hofmann KH, Keimel K, Lawson DJ, Mislove M, Scott DS. *Continuous Lattices and Domains*. In *Series: Encyclopedia of Mathematics and its Applications* (No. 93), Cambridge University Press, 2003.

[7] Zhang GQ. *Logic of Domains*. Birkhäuser, Boston, 1991.

[8] Joslyn C. Poset Ontologies and Concept Lattices as Semantic Hierarchies. *Lecture Notes in Computer Science* 2004;3127:287-302.

[9] Zhang GQ, Shen G, Tian Y, Sun J. Concept analysis as a formal method for menu design. *Lecture Notes in Computer Science* 2006;3941:173-87.

[10] RDF: <http://www.w3.org/RDF/>

[11] SPARQL: <http://www.w3.org/TR/rdf-sparql-query/>

[12] International Health Terminology Standard Development Organization (IHTSDO): <http://www.ihtsdo.org/>

[13] Donnelly K. SNOMED-CT: The advanced terminology and coding system for eHealth. *Stud Health Technol Inform* 2006;121:279-90

[14] Virtuoso: <http://virtuoso.openlinksw.com/>