ARTICLE TEMPLATE

An Algebraic Approach to Ontology Modularization and Knowledge Refinement

Andrew LeClair[a] and Ridha Khedri[b]

[a]A. LeClair is with the AI research lab at Genaiz
[b]R. Khedri is with the Department of Computing and Software, McMaster University, Hamilton, Canada

ARTICLE HISTORY
Compiled November 5, 2025

ABSTRACT
Ontology modularization is the process of extracting a smaller, reusable component of an ontology called a module. It is often used in the application domains of ontology alignment, ontology reasoning, and security and privacy. Ontology modularization, as presented in the literature, is subjected to two problems that cripple further formal exploration of modules. The first problem is that there is no precise definition of a module. The second is the lack of a formal approach to relate the knowledge of a module to that of its source ontology. In this paper, we present three contributions to the field to address these two problems. We present an algebraic approach to the modularization of an ontology. This allows us to precisely define what a module is, and how one can be extracted. The second contribution is an approach to characterize the knowledge that is retained or lost as a result of the modularization process. Finally, we present a formalization to specify modules as a refinement, or coarsening, of another set of modules. With this work, we set the ground for methods that are able to modularize an ontology motivated by goals of specific knowledge preservation or transformation.

KEYWORDS
Ontology, Ontology Modularization, Algebraic Modularization, Knowledge Refinement

1. Introduction

Ontologies provide a formal means to represent domain knowledge, facilitating semantic interoperability and reasoning among diverse system components Bittner, Donnelly, and Winter (2005); Guizzardi and Guarino (2024); Horrocks, Li, Turi, and Bechhofer (2004). By establishing an ontological commitment—an agreement on the concepts and relations within a domain—ontologies create a common language, crucial for communication between independent agents, such as sensors in an Internet of Things (IoT) ecosystem Pliatsios, Kotis, and Goumopoulos (2023) and multi-context reasoning LeClair, Marinache, El Ghalayini, MacCaull, and Khedri (2022).

This work focuses on domains conceptualized from structured or semi-structured datasets, common in fields like IoT and natural sciences Calosi and Graziani (2014).

CONTACT Andrew LeClair Author. Email: andrew.leclair@genaiz.com
CONTACT Ridha Khedri Author. Email: khedri@mcmaster.ca

In such domains, data often exhibits inherent part-whole relationships (mereology). For instance, accelerometer, GPS, and gyroscope data individually act as parts that, when combined, form a more complete whole for road quality analysis Fouad, Mahmood, Mahmoud, Mohamed, and Hassanien (2014). While mereology is foundational in abstract ontologies like the General Formal Ontology Herre et al. (2006), applying it effectively to concrete, data-driven domains presents challenges Barton, Tarbouriech, Vieu, and Ethier (2022).

Traditional ontology languages, such as OWL, excel at capturing conceptual structures Guarino, Oberle, and Staab (2009) but often struggle to directly represent the mereological relationships inherent in Cartesian datasets (i.e., the structured data itself) without complex extensions like Ontology-Based Data Access/Integration systems Pinkel et al. (2017); Poggi et al. (2008); Xiao et al. (2018). Extending OWL with Cartesian types suitable for mereological reasoning is non-trivial due to potential inferencing ambiguities Krieger and Willms (2015). To address this, we utilize the Domain Information System (DIS) Marinache (2016). DIS employs cylindric algebra for dataset representation and Boolean Algebra for conceptual mereological structure, providing a robust framework for representing parthood within structured data without the limitations faced by traditional approaches.

Within mereology, the concept of "coarsening" refers to simplifying or abstracting context, akin to moving from a specific location ("in Hyde Park") to a broader one ("in London") Bittner (2004). While trivial in simple taxonomies, identifying coarser yet consistent representations in rich mereological systems is complex. This notion of changing granularity or coarseness is relevant not only for comparing statements but also for comparing ontologies Kleshchev and Artemjeva (2005) and is particularly pertinent to ontology modularization Del Vescovo et al. (2020).

Ontology modularization aims to extract smaller, self-contained components (modules) from a larger ontology for purposes like specialized reasoning, maintenance, or privacy LeClair et al. (2022). Tasks range from finding one suitable module (*GetOne*) to the more complex task of finding all modules that satisfy certain criteria (*GetAll*) Del Vescovo, Parsia, Sattler, and Schneider (2010). Given the inherent parthood nature of mereology and the capabilities of DIS, we explore how the structure of DIS lends itself to modularization. Specifically, this paper investigates ontology modularization within the DIS framework by leveraging the concept of coarsening. We define and demonstrate a specific form of the *GetAll* task where modules are generated by systematically adjusting the granularity (coarseness) of the ontology's fundamental components (atoms). Crucially, the set of all modules generated by our method forms a lattice structure. Each module within this lattice is not only a proper subontology but, owing to the underlying Boolean Algebra of a DIS, is guaranteed to be a Boolean Subalgebra. This ensures that the fundamental algebraic operations are preserved within each module, maintaining structural consistency. Furthermore, the relationships within this lattice directly correspond to the notion of coarsening. This structure provides significant advantages: it allows for the direct comparison of any two modules to determine which is coarser, enables the identification of modules that precisely coarsen specific information, and potentially facilitates the creation of new modules (via lattice operations like meet and join) that merge or refine the perspectives of existing ones. Thus, our approach finds all modules that preserve the core knowledge, organized by their level of detail, offering a structured way to navigate different granularities of the original ontology.

The structure of the remainder of the paper is as follows. In Section 2, we introduce the background necessary for discussing the techniques introduced in this paper. In

Section 3, we present the technique of this paper for modularizing a DIS-based ontology. In Section 4, we examine the existing literature regarding multi-agent knowledge-based systems. In Section 5, we discuss the results of the modularization technique, and finally, we conclude by pointing to the research and our future work in Section 6.

2. Mathematical Background

In this Section, we present the necessary mathematical background for understanding of the definitions introduced.

2.1. Lattice Theory and Boolean Algebras

A *lattice* is an abstract structure that can be defined as either a relational or algebraic structure Davey and Priestley (2002). In our work, we use both the algebraic and relational definitions of lattices, selecting the one that best conveys our ideas conciseley. Accordingly, we present both definitions and explain the relationship between them.

Let $(L, \leq)$ be a partially ordered set. For an arbitrary subset $S \subseteq L$, an element $u \in L$ is an *upper bound* of S if $s \leq u$ for each $s \in S$. Dually, an element $l \in L$ is a *lower bound* of S if $l \leq s$ for each $s \in S$. An upper bound u is a *least upper bound* (or *join*, denoted $\bigvee S$ or $s_1 \oplus s_2$ for two elements) if $u \leq x$ for every upper bound x of S. Dually, a lower bound l is a *greatest lower bound* (or *meet*, denoted $\bigwedge S$ or $s_1 \otimes s_2$ for two elements) if $x \leq l$ for every lower bound x of S. If every two elements $a, b \in L$ have a join, L is a join-semilattice. If they have a meet, L is a meet-semilattice. A lattice is a partially ordered set that is both a join- and meet-semilattice.

Algebraically, a lattice can be defined as a structure $(L, \oplus, \otimes)$, where L is a set and $\oplus$ (join) and $\otimes$ (meet) are binary operations that are commutative, associative, idempotent (i.e., $a \oplus a = a$, $a \otimes a = a$), and satisfy the absorption laws (i.e., $a \oplus (a \otimes b) = a$ and $a \otimes (a \oplus b) = a$) for all $a, b \in L$.

The relational order $\leq$ and algebraic operations are connected by the equivalences: $a \leq b \iff a \otimes b = a \iff a \oplus b = b$. This allows us to interchange these perspectives as needed.

A *sublattice* is a non-empty subset M of a lattice L such that for all $x, y \in M$, $x \oplus y \in M$ and $x \otimes y \in M$.

A *Boolean lattice* is a *complemented distributive* lattice Sikorski (1969). A distributive lattice is one where $\otimes$ distributes over $\oplus$ and vice-versa. A complemented lattice is bounded, meaning it includes a universal *bottom* element (0 or $e_{\mathcal{C}}$) and a universal *top* element (1 or $\top$), and every element a has a unique *complement* a' such that $a \oplus a' = 1$ and $a \otimes a' = 0$.

The corresponding algebraic structure to a Boolean lattice is a *Boolean algebra* $\mathcal{B} = (B, \otimes, \oplus, ', 0, 1)$, where B is the carrier set. In a finite Boolean algebra, an *atom* is an element $a \in B$, $a \neq 0$, such that for any $x \in B$, if $x \leq a$, then either $x = 0$ or $x = a$ Halmos (2018). In this work, we consider only finite Boolean algebras. A finite Boolean algebra is generated by the power set of its atoms; any element is the join of a unique set of atoms Hirsch and Hodkinson (2002). All Boolean algebras with the same number of atoms are isomorphic. A *Boolean subalgebra* of $\mathcal{B}$ is a sublattice that is also closed under the complement operation and contains 0 and 1.

Two distinguished types of substructures (which are sublattices but not necessarily Boolean subalgebras) are *ideals* and *filters*. For a Boolean algebra $\mathcal{B}$ with carrier set B, $I \subseteq B$ is an ideal if I is non-empty, closed under join ($\forall i, j \in I, i \oplus j \in I$), and downward closed ($\forall i \in I, \forall b \in B, \text{if } b \leq i$ then $b \in I$; equivalently $\forall i \in I, \forall b \in B, i \otimes b \in I$). A filter $F \subseteq B$ is the dual: non-empty, closed under meet ($\forall i, j \in F, i \otimes j \in F$), and upward closed ($\forall i \in F, \forall b \in B, \text{if } i \leq b$ then $b \in F$; equivalently $\forall i \in F, \forall b \in B, i \oplus b \in F$).

An ideal I (or filter F) is *proper* if $I \neq B$ (or $F \neq B$). A *principal ideal* generated by $b \in B$ is $L_{\downarrow b} = \{a \in B \mid a \leq b\}$. A *principal filter* generated by $b \in B$ is $L_{\uparrow b} = \{a \in B \mid b \leq a\}$. An ideal I is *maximal* if $I \neq B$ and the only proper ideal strictly containing I is B itself. The dual of a maximal ideal is an *ultrafilter*. For any maximal ideal I (or ultrafilter F) and any element $x \in B$, exactly one of x or x' is in I (or F).

2.1.1. Partition Lattices and Boolean Subalgebras

The concept of partitioning the finite set of atoms of a Boolean algebra is central to constructing its subalgebras. A *partition* P of a set X is a collection of non-empty, pairwise disjoint subsets of X (called blocks) whose union is X. Formally, $P \subseteq \mathcal{P}(X)$ such that:

(i) For every $A \in P$, $A \neq \emptyset$.
(ii) For every $A, B \in P$, if $A \neq B$, then $A \cap B = \emptyset$.
(iii) $\bigcup_{A \in P} A = X$.

Let P_1 and P_2 be partitions of X. P_1 is a *refinement* of P_2 (or P_2 is coarser than P_1), denoted $P_1 \leq_p P_2$, if every block in P_1 is a subset of some block in P_2:

$$P_1 \leq_p P_2 \iff \forall s \in P_1, \exists t \in P_2 \text{ such that } s \subseteq t. \tag{1}$$

The set of all partitions of X forms a *partition lattice* when ordered by the refinement relation. The top of the partition lattice is the coarsest partition $\{X\}$ (one block containing all elements), and the bottom is the finest partition where each block is a singleton set $\{\{x\} \mid x \in X\}$.

For a finite Boolean algebra $\mathcal{B}$ with set of atoms X_{atoms}, there is a dual isomorphism between its lattice of Boolean subalgebras and the partition lattice of X_{atoms} Birkhoff (1940). Each partition P of X_{atoms} corresponds to a unique Boolean subalgebra of $\mathcal{B}$, whose atoms are formed by taking the join ($\oplus$) of the original atoms within each block of P. The lattice of Boolean subalgebras is ordered by the subalgebra relation ($\subseteq$), which corresponds to the inverse of the refinement relation $\leq_p$ on partitions. The original Boolean algebra $\mathcal{B}$ corresponds to the finest partition (atoms of $\mathcal{B}$ are atoms of the subalgebra), and the trivial two-element Boolean subalgebra $\{0, 1\}$[1] is the coarsest.

2.2. Domain Information System

A Domain Information System (DIS) Marinache (2016) consists of three components: a domain ontology, a domain data view, and a function mapping between them. We focus on the domain ontology component for modularization. A domain ontology models concepts and their relationships. Let C be a set of concepts, and $(C, \oplus, e_C)$ be a commutative idempotent monoid[2], where e_C is the identity for $\oplus$.

Within C, a subset $L \subseteq C$ forms the carrier set of a Boolean lattice $\mathcal{L} = (L, \leq_L, e_C, \top)$, where $\leq_L$ is the lattice order, e_C is its bottom element (analogous to 0), and $\top$

[1] where $0 = 1$ is not standard

[2] A monoid is an algebraic structure with a single associative operator and an identity element.

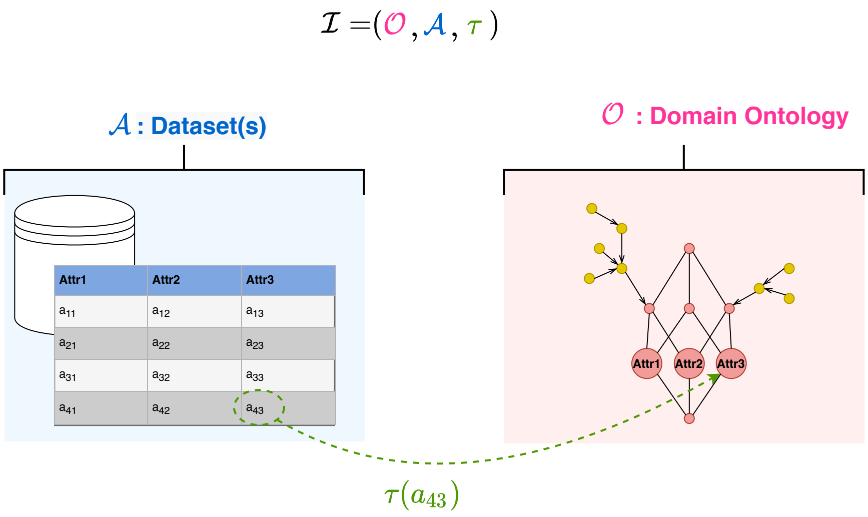
is its top element (analogous to 1). This order $\leq_L$ is referred to as the *partOf* relation, denoted $\sqsubseteq_{\mathcal{C}}$. The Boolean algebra corresponding to $\mathcal{L}$ is $\mathcal{B}_L = (L, \otimes, \oplus, ', e_{\mathcal{C}}, \top)$, where $\otimes$ (meet) and $\oplus$ (join) are related to $\sqsubseteq_{\mathcal{C}}$ (i.e., $\leq_L$) by: $a \sqsubseteq_{\mathcal{C}} b \iff a \leq_L b \iff a \otimes b = a \iff a \oplus b = b$.

A concept $k \in L$ is *atomic* in $\mathcal{B}_L$ if $k \neq e_{\mathcal{C}}$ and for any $k' \in L$, if $k' \sqsubseteq_{\mathcal{C}} k$ (i.e., $k' \leq_L k$), then $k' = k$ or $k' = e_{\mathcal{C}}$.

The domain ontology may also include rooted graphs. Let $C_i \subseteq C, R_i \subseteq C_i \times C_i$, and $t \in C_i$. A rooted graph at t, $G_t = \big(C_i, R_i, t\big)$, is a connected directed graph of concepts with t as its unique sink (root): $t \in C_i$ is *root* of $G_t \iff \forall (k \mid k \in C_i : k = t \vee (k, t) \in {R_i}^+)$.

Definition 2.1 (Domain Ontology)**.** Let $\mathcal{C} = \big(C, \oplus, e_{\mathcal{C}}\big)$ be a commutative idempotent monoid. Let $\mathcal{L} = (L, \sqsubseteq_{\mathcal{C}})$ (equivalently $(L, \leq_L)$) be a Boolean lattice, with $L \subseteq C$, such that $e_{\mathcal{C}} \in L$. Let $\mathcal{G} = \{G_t \mid G_t \textit{ is a rooted graph of } \; t \wedge t \in L\}$. A *domain ontology* is the mathematical structure $\mathcal{O} \overset{\text{def}}{=} \big(\mathcal{C}, \mathcal{L}, \mathcal{G}\big)$.

Figure 1 shows a high-level representation of a DIS, separating the domain ontology from the data view (linked by a τ operator, details in Marinache (2016)). We focus on modularizing $\mathcal{O}$.



$\mathcal{I} = (\mathcal{O}, \mathcal{A}, \tau)$
$\mathcal{A}$: Dataset(s)
$\mathcal{O}$: Domain Ontology
Attr1
Attr2
Attr3
a11
a12
a13
a21
a22
a23
a31
a32
a33
a41
a42
a43
Attr1
Attr2
Attr3
$\tau(a_{43})$

Figure 1. High-level representation of a Domain Information System

Figure 2 shows the Boolean lattice $\mathcal{L}$ of a DIS representation for Park and Recreation Properties (data from *Recreation and Parks Properties* (n.d.), simplified in Table 1). Each attribute (column like *Longitude*, *Latitude*, etc.) from the dataset forms an atom of $\mathcal{L}$. The concept *Park Property* is $\top$. Concepts like *Location* (*Longitude* $\oplus$ *Latitude*) are joins of atoms. The figure also shows rooted graphs, e.g., associated with the concept *Park* (*Long.* $\oplus$ *Lat.* $\oplus$ *Acres* $\oplus$ *Prop. Type*).

3. Ontology Modularization

This work presents a unified notion of modularization for a domain ontology within the Domain Information System (DIS) framework Marinache (2016). As highlighted in the introduction, previous approaches often lack a precise, universally applicable

Table 1. Park Property Dataset

Longitude	Latitude	Acres	Property Type	Realtor Neighborhood	City
-122.39982015	37.7955308	2.012	Civic Plaza	Financial Distric	San Francisco
-122.49239745	37.7219234	608.486	Regional Park	Lake Shore	San Francisco
-122.41742016	37.79642235	0.1489	Mini Park	Nob Hill	San Francisco
-122.45465821	37.73906281	40.7118	Regional Park	Miraloma Park	San Francisco
-122.40750049	37.78793123	2.5997	Civic Plaza	Downtown	San Francisco
-122.41204637	37.74706105	2.2460	Neighborhood Park	Bernal Heights	San Francisco
...	...	...	...	...	...

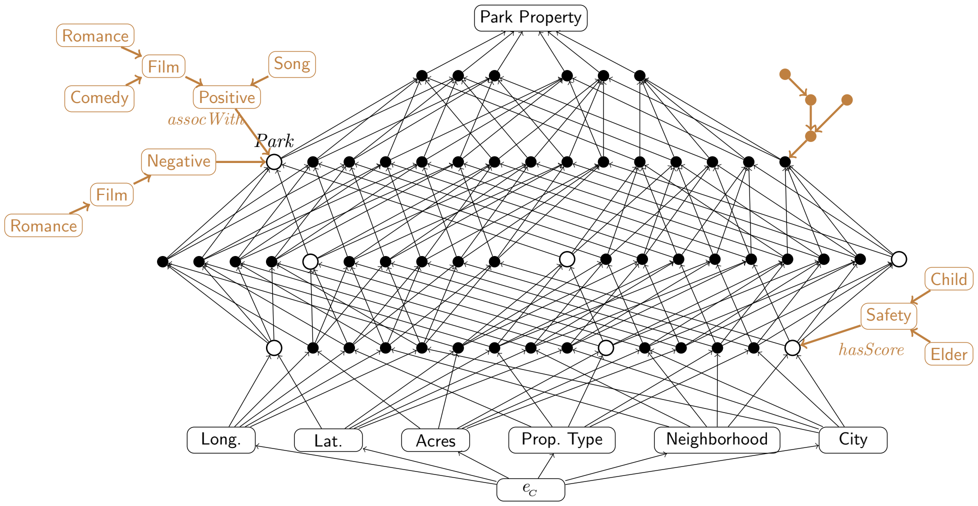


Figure 2. The Boolean lattice for the Park Properties

definition of a module or a formal way to relate the knowledge content of a module back to its source. This section directly addresses these issues by leveraging the algebraic foundations of DIS. We introduce a formal definition for modules based on Boolean subalgebras, allowing for rigorous analysis of their properties and knowledge content relative to the original ontology and each other.

We begin with the general definition of a module, adapted from LeClair., Khedri., and Marinache (2019), which establishes the basic structural requirements for a sub-ontology within DIS.

Definition 3.1 (Module LeClair. et al. (2019))**.** Given a domain ontology $\mathcal{O} = (\mathcal{C}, \mathcal{L}, \mathcal{G})$, a *module* $\mathcal{M}$ of $\mathcal{O}$ is defined as a domain ontology $\mathcal{M} = (\mathcal{C}_M, \mathcal{L}_M, \mathcal{G}_M)$ that satisfies the following conditions:

- $C_M \subseteq C$
- $\mathcal{L}_M = (L_M, \sqsubseteq_C)$ such that $L_M \subseteq C_M$, $\mathcal{L}_M$ is a Boolean sublattice of $\mathcal{L}$, and $e_c \in L_M$
- $\mathcal{G}_M = \{G_n \mid G_n = (C_n, R_n, t_n) \wedge G_n \in \mathcal{G} \wedge t_n \in L_M\}$

where C_M and C are the carrier sets of $\mathcal{C}_M$ and $\mathcal{C}$, respectively.

With this definition, the process of modularization can be understood as a function that produces a sub-ontology conforming to these criteria.

To evaluate the knowledge preservation capabilities of a module $\mathcal{M}$ with respect to its source ontology $\mathcal{O}$, we adapt the established notion of a *conservative extension* from Description Logics Lutz, Walther, and Wolter (2007). In DL, an ontology $\mathcal{T}'$ is a Σ-conservative extension of $\mathcal{T}$ if, for any statement α using only symbols from a signature $\Sigma \subseteq \text{sig}(\mathcal{T})$, $\mathcal{T} \models \alpha$ if and only if $\mathcal{T}' \models \alpha$. This means the extended ontology $\mathcal{T}'$ neither loses nor adds knowledge expressible in the signature Σ.

In the context of a DIS ontology $\mathcal{O} = (\mathcal{C}, \mathcal{L}, \mathcal{G})$ and a module $\mathcal{M} = (\mathcal{C}_M, \mathcal{L}_M, \mathcal{G}_M)$, the signature of the module can be considered as the union of the concept language, lattice language, and the relational language coming from the graph structure of the rooted graphs attached to it. The signature of the lattice part $\mathcal{L}_M = (L_M, \oplus, e_c, \top_{\mathcal{L}})$, where $L_M \subseteq C_{At_c}$ that includes the atomic concepts. Also, we have the signature of the relational part of the ontology as given by the rooted graphs. This signature is $\mathcal{M}_{\mathcal{G}} = (C, R_{t_1}, \cdots, R_{t_n})$ for R_{t_i}, $1 \leq i \leq n$, is a relation of the rooted graph G_{t_i}.

We define a DIS module $\mathcal{M}$ to be *conservatively extended* by an $\mathcal{O}$ (i.e., $\mathcal{O}$ is a *conservative extension of* $\mathcal{M}$) with respect to its signature if it satisfies two conditions:

(1) **Algebraic Conservativity:** For any algebraic statement ϕ (e.g., $a \sqsubseteq_C b$, $a \oplus b = c$, $a' = d$) formulated using only concepts from L_M:

$$\mathcal{M} \models_{alg} \phi \iff \mathcal{O} \models_{alg} \phi$$

where $\models_{alg}$ denotes entailment within the respective Boolean algebras. This means $\mathcal{M}$'s Boolean lattice $\mathcal{L}_M$ must behave identically to $\mathcal{L}$ for all statements expressible purely within L_M.

(2) **Structural Conservativity:** For any structural statement ψ concerning paths or reachability between concepts in $\mathcal{C}_M$ via relations present in $\mathcal{G}_M$ (i.e., statements about entailments derived from the graph structures in $\mathcal{G}_M$):

$$\mathcal{M} \models_{struct} \psi \iff \mathcal{O} \models_{struct} \psi$$

where $\models_{struct}$ denotes entailment derived from the graph structures. This primarily implies that if a graph $G_t \in \mathcal{G}_M$ (with $t \in L_M$), its structure and the relations it contains pertinent to $\mathcal{C}_M$ are preserved from $\mathcal{O}$.

A module $\mathcal{M}$ for which $\mathcal{O}$ is a conservative extension is considered *locally correct* (if $\mathcal{M} \models \alpha \implies \mathcal{O} \models \alpha$) and *locally complete* (if $\mathcal{O} \models \alpha \implies \mathcal{M} \models \alpha$ for α in sig($\mathcal{M}$)).

The general module definition (Definition 3.1) ensures that $\mathcal{L}_M$ is a Boolean sublattice and $\mathcal{G}_M$ contains only relevant subgraphs. This provides a basis for structural conservativity for the included graphs. However, merely being a sublattice does not guarantee algebraic conservativity, particularly for the complement operator ($'$), as seen with view traversal modules. This limitation motivates our focus on modules where $\mathcal{L}_M$ is a full Boolean subalgebra of $\mathcal{L}$, as this directly ensures algebraic conservativity.

Such algebraically sound modules can be systematically constructed by leveraging the relationship between Boolean subalgebras and partitions of the atoms of the original Boolean algebra $\mathcal{B}$ (associated with $\mathcal{L}$), as introduced in Section 2.1.1. Let X be the set of atoms of $\mathcal{B}$. Any partition P of X induces a unique Boolean subalgebra of $\mathcal{B}$. The atoms A_P of this subalgebra are formed by taking the join ($\oplus$) of the original atoms within each block of the partition P:

For a given $B \in P$, we denote by S_B the quantification $\oplus(x \mid x \in B : x)$ on the elements of B. Then, A_P is defined as follows:

$$A_P = \{S_B \ \mid \ B \in P\} \tag{2}$$

Based on this construction, we define an algebraic module as follows:

Definition 3.2 (Algebraic Module)**.** Let $\mathcal{O} = (\mathcal{C}, \mathcal{L}, \mathcal{G})$ be a domain ontology with associated Boolean algebra $\mathcal{B}$ over the carrier set L, and let X be the set of atoms of $\mathcal{B}$. Let P be a partition of X. An *algebraic module* $\mathcal{M}_P$ corresponding to partition P is a domain ontology $\mathcal{M}_P = (\mathcal{C}_P, \mathcal{L}_P, \mathcal{G}_P)$ where:

- A_P is the set of atoms generated from P using Equation 2.
- $\mathcal{L}_P = (L_P, \sqsubseteq_C)$ is the Boolean lattice generated by the power set of A_P, such that $L_P \subseteq L$. $\mathcal{L}_P$ is a Boolean subalgebra of $\mathcal{L}$.
- $\mathcal{C}_P = \{c \mid c \in L_P \vee \exists(G_n \mid G_n \in \mathcal{G}_P : c \in C_n)\}$ is the carrier set for the monoid $\mathcal{C}_P$.
- $\mathcal{G}_P = \{G_n \mid G_n = (C_n, R_n, t_n) \wedge G_n \in \mathcal{G} \wedge t_n \in L_P\}$

By its construction, where $\mathcal{L}_P$ is explicitly a Boolean subalgebra of $\mathcal{L}$ (and thus also a Boolean sublattice containing e_c), an algebraic module as defined in Definition 3.2 inherently satisfies the conditions specified in the general Definition 3.1. It provides a concrete, algebraically grounded method for creating modules with desirable properties.

Having defined the general algebraic module based on atom partitions, we now examine its relationship to the *view traversal* technique introduced in LeClair. et al. (2019).

3.1. View Traversal Modules and Their Limitations

The view traversal technique produces a module whose Boolean lattice L_M consists of the elements in a *principal ideal* $L_{\downarrow c}$ generated by a chosen concept c from the original ontology $\mathcal{O}$. The extraction of a view traversal module that satisfies the conditions of Definition 3.1 can be seen in Figures 3 and 4. Figure 3 identifies the elements of the principal ideal generated by the concept *Neighborhood Feature* and Figure 4 shows the view traversal module with *Neighborhood Feature* as its top concept and the new atoms of *Property Type*, *Neighborhood*, and *City*. However, as noted earlier, the resulting Boolean lattice $L_{\downarrow c}$ is generally *not* a Boolean subalgebra of the original lattice $\mathcal{L}$. For instance, the complement of $e_{_C}$ within the view traversal module (Figure 4) is *Neighborhood Feature*, whereas in the full ontology (Figure 3), $e'_{_C} = \top$. This lack of algebraic closure means the complement operator is not preserved, leading to potential inconsistencies and limiting the scope of algebraic analysis.

3.2. The Principal Ideal Subalgebra Module (PISM)

To address the algebraic limitations of view traversal while retaining the intuition of focusing on a specific concept c and its sub-concepts (the principal ideal $L_{\downarrow c}$), we can minimally extend the view traversal module to form a proper Boolean subalgebra. This is achieved by including not only the principal ideal $L_{\downarrow c}$ but also the corresponding *principal filter* $L_{\uparrow c'}$ generated by the complement c' of the starting concept c Harding, Heunen, Lindenhovius, and Navara (2017). The union of the principal ideal and its corresponding principal filter forms the carrier set of a Boolean subalgebra. This leads to the definition of the Principal Ideal Subalgebra Module (PISM).

Let $\mathcal{L}$ be the Boolean lattice of the ontology, $\mathcal{B}$ its associated Boolean algebra, $c \in L$ a concept, and c' its complement. Let $L_{\downarrow c}$ be the principal ideal generated by c, and

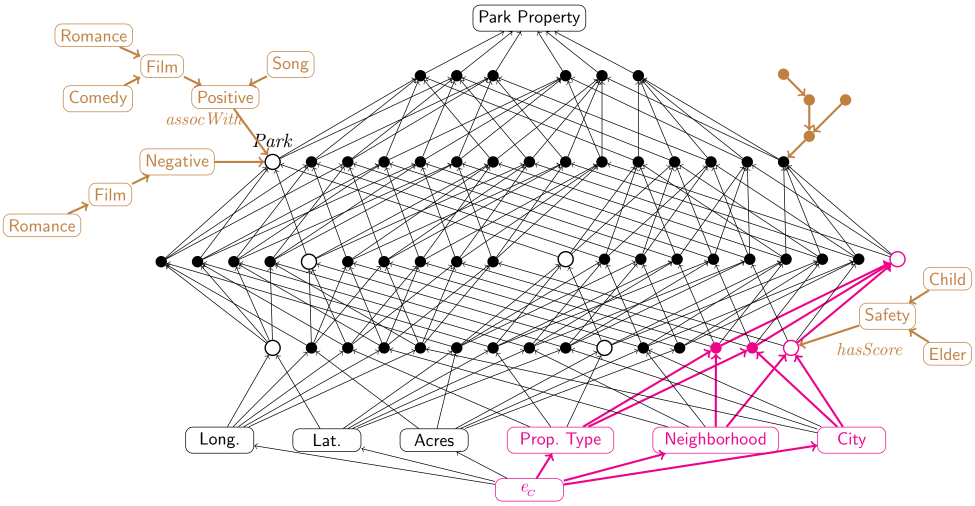


Figure 3. The Boolean lattice for the Park Ontology with the view traversal module highlighted for $c =$ *Neighborhood Feature*

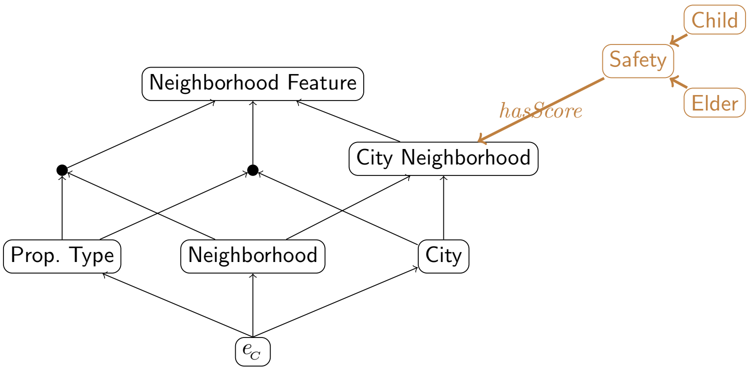


Figure 4. The view traversal module using $c =$ *Neighborhood Feature*

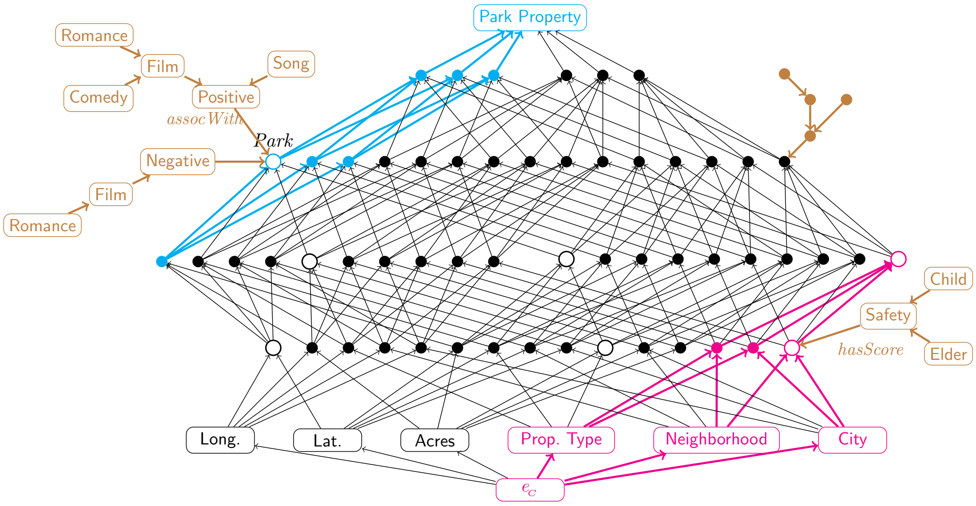


Figure 5. The Boolean lattice for the Park Ontology with the principal ideal subalgebra module highlighted for $c = \mathit{Neighborhood\ Feature}$

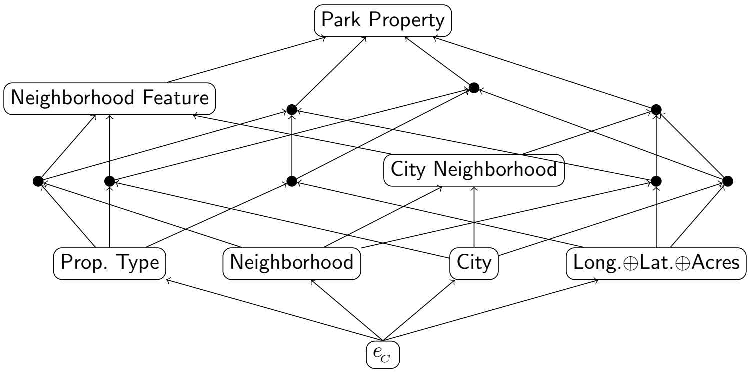


Figure 6. The Boolean lattice of the principal ideal subalgebra module extracted from the Park Ontology with $c = \mathit{Neighborhood\ Feature}$

$L_{\uparrow c'}$ be the principal filter generated by c'. Let $L_{\updownarrow c} \overset{\text{def}}{=} L_{\downarrow c} \cup L_{\uparrow c'}$ be the carrier set for the resulting Boolean subalgebra $\mathcal{B}_{\updownarrow c}$.

Definition 3.3 (Principal Ideal Subalgebra Module)**.** Let $\mathcal{O}(\mathcal{C}, \mathcal{L}, \mathcal{G})$ be a given domain ontology. Let c be a concept in the carrier set L of $\mathcal{L}$. Let $L_{\updownarrow c} \overset{\text{def}}{=} L_{\downarrow c} \cup L_{\uparrow c'}$. The *principal ideal subalgebra module* $\mathcal{M}_{\updownarrow c} = (\mathcal{C}_{\updownarrow c}, \mathcal{L}_{\updownarrow c}, \mathcal{G}_{\updownarrow c})$ is defined as:

(1) $\mathcal{L}_{\updownarrow c} \overset{\text{def}}{=} (L_{\updownarrow c}, \sqsubseteq_V)$, where $\mathcal{L}_{\updownarrow c}$ is the Boolean subalgebra generated by $L_{\updownarrow c}$ and $\sqsubseteq_V$ is the restriction relation of $\sqsubseteq_C$ to $L_{\updownarrow c}$.

(2) $\mathcal{G}_{\updownarrow c} \overset{\text{def}}{=} \{G_i \mid G_i \in \mathcal{G} \ \wedge \ G_i = (C_i, R_i, t_i) \wedge t_i \in L_{\updownarrow c}\}$

(3) $C_{\updownarrow c} \overset{\text{def}}{=} \{k \mid k \in L_{\updownarrow c} \vee \exists (G_i \mid G_i \in \mathcal{G}_{\updownarrow c} : k \in C_i)\}$ is the carrier set for $\mathcal{C}_{\updownarrow c}$.

Figure 5 illustrates the construction of a PISM for the concept defined as $c = \mathit{Prop.Type} \oplus \mathit{Neighborhood} \oplus \mathit{City}$. The magenta sublattice is the principal ideal $L_{\downarrow c}$, and the cyan sublattice is the principal filter $L_{\uparrow c'}$ (defined with the remaining atoms, $c' = \mathit{Long.} \oplus \mathit{Lat.} \oplus \mathit{Acres}$). The union of these disjoint sets (except for e_c and $\top$ potentially being boundary cases depending on c) forms the carrier set $L_{\updownarrow c}$. The resulting PISM's Boolean lattice is shown in Figure 6.

This PISM construction is, by definition, an algebraic module (satisfying Definition 3.2) and thus also a module (satisfying Definition 3.1). It resolves the complement issue of the view traversal module.

Claim 3.1. Let $\mathcal{M}_c$ be the module obtained by view traversal using concept c. Let c' be the complement of c. The module $\mathcal{M}_{\updownarrow c}$ (the PISM) that is an extension of $\mathcal{M}_c$ with the principal filter generated by c' is locally correct and locally complete.

Proof. The claim holds by construction. The PISM $\mathcal{M}_{\updownarrow c}$ is formed using a Boolean subalgebra $\mathcal{B}_{\updownarrow c}$ which is a subalgebra of the ontology's Boolean algebra $\mathcal{B}$. Therefore, any formula satisfiable in $\mathcal{B}_{\updownarrow c}$ is satisfiable in $\mathcal{B}$, and any formula written only using elements of $L_{\updownarrow c}$ that is satisfiable in $\mathcal{B}$ is also satisfiable in $\mathcal{B}_{\updownarrow c}$ (conservative extension property). For instance, as $e_c \in L_{\updownarrow c}$, the formula $e'_c = \top$ is true in both the PISM and the original ontology (Figure 6 vs Figure 2). However, the formula $\mathit{Acres} \sqsubseteq_c \mathit{Long.} \oplus \mathit{Lat.} \oplus \mathit{Acres}$, while true in the original ontology, is not expressible in the PISM of Figure 6 because *Acres* is not an element (concept) of $L_{\updownarrow c}$, while $\mathit{Long.} \oplus \mathit{Lat.} \oplus \mathit{Acres}$ *is* (and is, in fact, an atom of this PISM). □

The PISM construction introduces a *coarsening* effect. The atoms of the PISM (visible in Figure 6) are the atoms of the principal ideal $L_{\downarrow c}$ plus the single element c' (which is the bottom element of the principal filter $L_{\uparrow c'}$). Notice that $c' = \mathit{Long.} \oplus \mathit{Lat.} \oplus \mathit{Acres}$ is an atom in the PISM, but it is decomposable into three distinct atoms (*Long.*, *Lat.*, *Acres*) in the original ontology. By making c' atomic within the PISM, we lose the ability to reason about its constituent parts individually within that module. The knowledge represented is thus coarser than in the original ontology.

While PISMs provide an algebraically sound extension of view traversal, they come with trade-offs regarding size and isomorphism:

Claim 3.2 (Maximal Principal Ideal Subalgebra Module)**.** Let c be a concept in an ontology's Boolean lattice L. If the principal ideal $L_{\downarrow c}$ generated by c is a maximal ideal, then the resulting principal ideal subalgebra module $\mathcal{M}_{\updownarrow c}$ is equal to the original ontology $\mathcal{O}$.

Proof. This follows from the definitions of a maximal ideal and ultrafilter. If $I = L_{\downarrow c}$ is maximal, its corresponding filter $F = L_{\uparrow c'}$ is an ultrafilter. For any $x \in L$, a maximal ideal I contains exactly one of x or x'. Since F is the dual, if $x \in I$, then $x' \in F$. Therefore, every element $x \in L$ belongs to either I or F, meaning $L_{\updownarrow c} = I \cup F = L$. The PISM is the original ontology. □

Claim 3.3. Let c be a concept in the Boolean lattice L. If the principal ideal $L_{\downarrow c}$ generated by c is not a maximal ideal, then the resulting principal ideal subalgebra $\mathcal{B}_{\updownarrow c}$ (and thus the PISM $\mathcal{M}_{\updownarrow c}$) is not isomorphic to the original Boolean algebra $\mathcal{B}$.

Proof. We prove this by contradiction. Assume $I = L_{\downarrow c}$ is not maximal, but assume $L_{\updownarrow c} = I \cup F = L$. This implies every element $x \in L$ belongs to either I or F. Consequently, every atom a of $\mathcal{B}$ must belong to either I or F. If F is a filter formed from a composite concept c (a concept formed from the composition of atoms), then I is an ideal that covers all atoms that are not the atoms of c. Therefore, the atoms of c are not included in $L_{\updownarrow c}$. The only way to include the atoms of c is for if the filter F is formed over one of the atoms. However, a filter formed over an atom is a maximal filter, and therefore I would be the corresponding maximal ideal. Therefore, we have a contradiction. The only way to include all concepts is for F to be formed over an atom and be a maximal filter. □

Claim 3.3 guarantees that if the generating concept c is not ”almost maximal” (specifically, if c is not the join of all atoms but one), the resulting PISM is strictly smaller in terms of the number of concepts in its Boolean lattice than the original ontology. This addresses the motivation of extracting smaller components. However, the PISM is necessarily larger than the corresponding view traversal module M_c due to the inclusion of $L_{\uparrow c'}$. There is a trade-off: the view traversal module is smaller but lacks algebraic closure; the PISM provides algebraic closure but is larger.

3.2.1. Limitations of the PISM Approach

A crucial point is that not all algebraic modules can be constructed as PISMs. Consider the function f mapping a principal ideal $p = L_{\downarrow c}$ to its corresponding PISM $\mathcal{M}_{\updownarrow c}$ (whose algebra is $\mathcal{B}_{\updownarrow c}$).

Claim 3.4. Let P_{ideal} be the set of all principal ideals of $\mathcal{B}$ and let B_{sub} be the set of all Boolean subalgebras of $\mathcal{B}$. Let $f : P_{ideal} \longrightarrow B_{sub}$ be the function where $f(L_{\downarrow c}) = \mathcal{B}_{\updownarrow c}$ (the Boolean algebra of the PISM generated from c). The function f is total, injective, and non-surjective.

Proof. Totality: Every principal ideal $L_{\downarrow c}$ has a corresponding principal filter $L_{\uparrow c'}$, and their union generates a unique Boolean subalgebra $\mathcal{B}_{\updownarrow c}$. Thus, f is total. Injectivity: A principal ideal $L_{\downarrow c}$ uniquely determines c. The complement c' is also unique, determining the unique principal filter $L_{\uparrow c'}$. The resulting subalgebra $\mathcal{B}_{\updownarrow c}$ depends uniquely on c. Thus, f is injective. Non-surjectivity: Consider the Boolean algebra $\mathcal{B}$ corresponding to Figure 2. Let $\mathcal{B}_1$ be the Boolean subalgebra generated by the partition $P_1 = \{\{\mathit{Long.} \oplus \mathit{Lat.} \oplus \mathit{Acres}\}, \{\mathit{Prop.\ Type} \oplus \mathit{Neighborhood} \oplus \mathit{City}\}\}$. $\mathcal{B}_1$ is a valid Boolean subalgebra of $\mathcal{B}$ (it corresponds to the bottom module in Figure 9, excluding the trivial top/bottom). However, $\mathcal{B}_1$ cannot be generated as a PISM, because neither of its atoms generates a principal ideal whose union with the corresponding filter yields exactly $\mathcal{B}_1$. For instance, the ideal for $c = \mathit{Prop.\ Type} \oplus \ldots$ includes many other elements. Thus, there exists a Boolean subalgebra $\mathcal{B}_1$ that has no pre-image under f. Therefore, f is non-surjective. □

The non-surjectivity demonstrated by Claim 3.4 is significant: it proves that focusing solely on extending view traversal modules (PISMs) is insufficient to capture *all* possible algebraically sound modules (Boolean subalgebras). This motivates the general approach using the lattice of all partitions.

3.3. The Lattice of Algebraic Modules and Knowledge Refinement

As established in Section 2.1.1, there is a lattice structure over all Boolean subalgebras of a given finite Boolean algebra $\mathcal{B}$. This lattice is dually isomorphic to the partition lattice of the atoms X of $\mathcal{B}$. Since each algebraic module (Definition 3.2) corresponds uniquely to a Boolean subalgebra (generated by a partition P of X), the set of all possible algebraic modules for a given ontology $\mathcal{O}$ also forms a lattice, ordered by the module refinement relation.

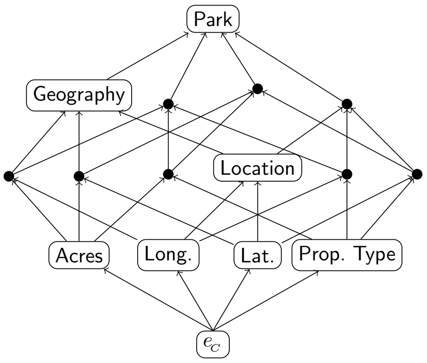


Figure 7. The Boolean lattice of the module extracted from the Park Ontology with $c = \mathit{Park}$

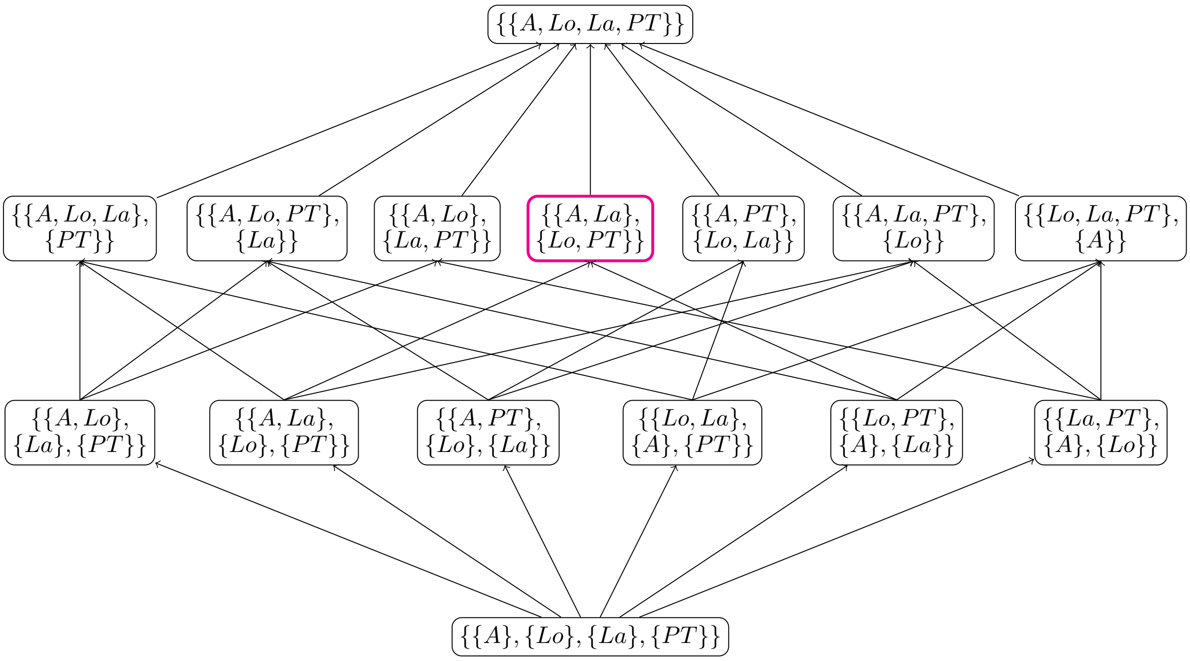
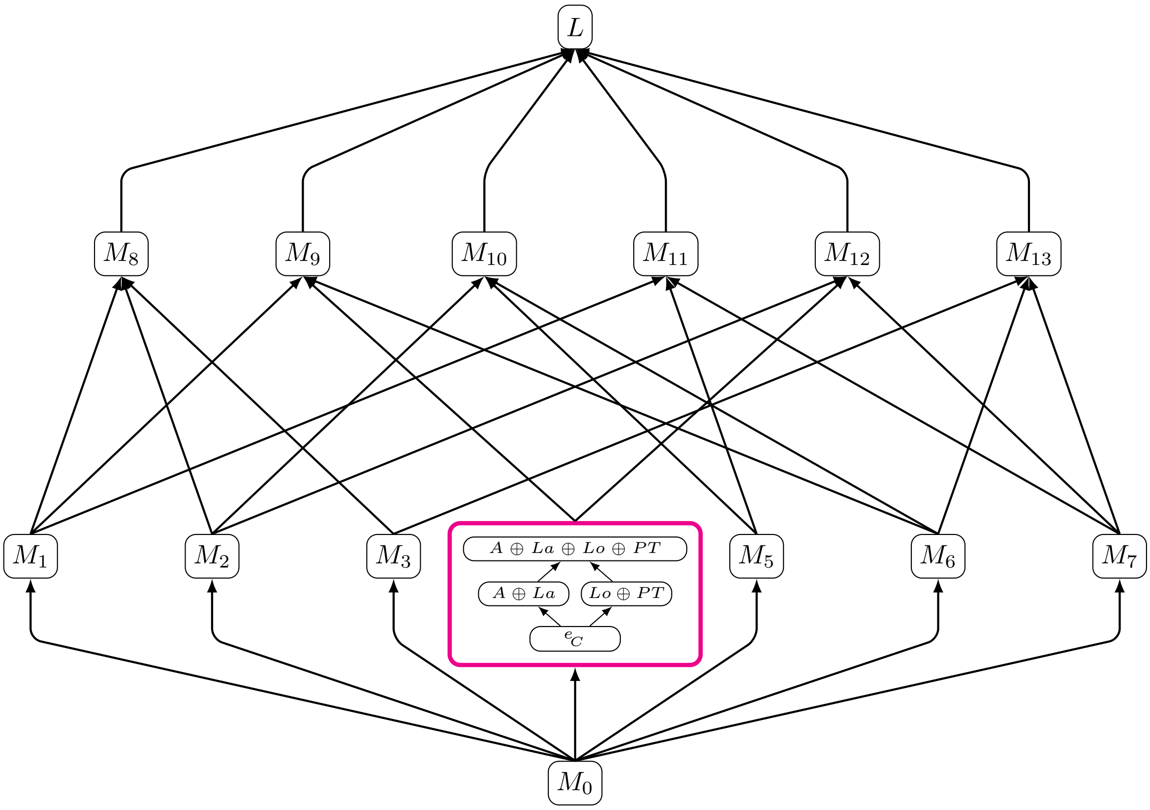

$\{\{A, Lo, La, PT\}\}$
$\{\{A, Lo, La\}, \{PT\}\}$
$\{\{A, Lo, PT\}, \{La\}\}$
$\{\{A, Lo\}, \{La, PT\}\}$
$\{\{A, La\}, \{Lo, PT\}\}$
$\{\{A, PT\}, \{Lo, La\}\}$
$\{\{A, La, PT\}, \{Lo\}\}$
$\{\{Lo, La, PT\}, \{A\}\}$
$\{\{A, Lo\}, \{La\}, \{PT\}\}$
$\{\{A, La\}, \{Lo\}, \{PT\}\}$
$\{\{A, PT\}, \{Lo\}, \{La\}\}$
$\{\{Lo, La\}, \{A\}, \{PT\}\}$
$\{\{Lo, PT\}, \{A\}, \{La\}\}$
$\{\{La, PT\}, \{A\}, \{Lo\}\}$
$\{\{A\}, \{Lo\}, \{La\}, \{PT\}\}$

Figure 8. The partition lattice for *Park*



Label	Corresponding Partition of Atoms
L	The original ontology (finest partition: $\{\{A\},\{Lo\},\{La\},\{PT\}\}$)
M_8 - M_{13}	Modules generated from partitions with 3 blocks
(magenta node)	Module M_4 from partition $\{\{A \oplus La\},\{Lo \oplus PT\}\}$. Its internal lattice structure is shown explicitly.
$M_1, M_2, M_3, M_5, M_6, M_7$	Other modules generated from partitions with 2 blocks
M_0	The trivial module (coarsest partition: $\{\{A, Lo, La, PT\}\}$)

Figure 9. The lattice of Boolean subalgebras for *Park*. Each node is an algebraic module. The internal lattice structure is shown explicitly for the highlighted module (magenta). The table provides the mapping from the other labels to their corresponding atom partitions.

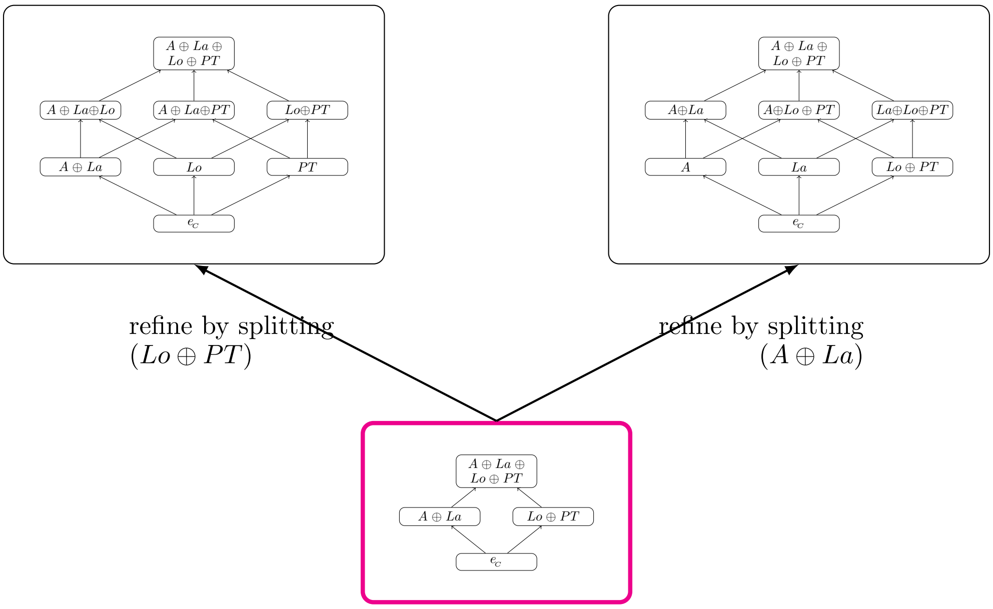


Figure 10. A detailed illustration of the module refinement process for module M_4, highlighted in Figure 9. The module M_4 shows the full Boolean lattice generated by its two atoms. It can be refined into M_9 or M_{12} by splitting one of its atoms, resulting in modules whose lattices are generated by three atoms, as shown.

Figure 7 shows the Boolean lattice for our running Park ontology example. Figure 8 depicts the partition lattice for the set of its atoms $X = \{A, Lo, La, PT, N, C\}$ (using abbreviations). Figure 9 shows the corresponding lattice of Boolean subalgebras (algebraic modules). Each node in Figure 9 represents an algebraic module, whose Boolean lattice is generated using Equation 2 from the corresponding partition in Figure 8. For example, the magenta highlighted partition $P = \{\{A, La\}, \{Lo, PT\}\}$ in Figure 8 yields the atoms $A_P = \{A \oplus La, Lo \oplus PT\}$, generating the algebraic module shown in the magenta box in Figure 9.

This lattice structure provides a formal way to compare the knowledge granularity of different modules.

Definition 3.4 (Module Coarsening/Refinement)**.** Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two algebraic modules built from partitions P_1 and P_2 of the atoms of the original ontology, respectively. We say that $\mathcal{M}_1$ *is a coarsening of* $\mathcal{M}_2$, or $\mathcal{M}_2$ *is a refinement of* $\mathcal{M}_1$, iff $P_2 \leq_p P_1$, and this is denoted by $\mathcal{M}_1 \leq_M \mathcal{M}_2$.

In Figure 9, the relation $\leq_M$ corresponds to moving downwards in the lattice, towards coarser modules with fewer, more combined atoms. The module At_1 (magenta box) is a refinement ($\mathcal{M}_{bottom} \leq_M \mathrm{At}_1$) of the bottom-most non-trivial module (generated by partitioning atoms into just two groups). Furthermore, At_1 itself can be refined into either At_2 (atoms $\{A \oplus La, Lo, PT\}$) or At_3 (atoms $\{A, La, Lo \oplus PT\}$), meaning $\mathrm{At}_1 \leq_M \mathrm{At}_2$ and $\mathrm{At}_1 \leq_M \mathrm{At}_3$.

To make this refinement process more concrete, we can examine the algebraic basis for the relationship between these modules. Figure 10 provides a detailed illustration of how At_1 (labeled M_4 in our diagrams) is refined. The module M_4 is generated from a partition containing the composite atom $(Lo \oplus PT)$. It is refined into module M_9 by splitting this atom back into its constituent parts, $\{Lo\}$ and $\{PT\}$, resulting in a finer partition with three atoms. The other composite atom $(A \oplus PT)$ can be split to its constituent parts to produce M_{12}. This visually demonstrates that the refinement relation $\leq_M$ is not merely an abstract ordering, but corresponds to a concrete algebraic operation on the module's atoms.

The lattice operations $\vee$ (join) and $\wedge$ (meet) on this lattice of algebraic modules have meaningful interpretations related to knowledge management:

- $\mathcal{M}_A \vee \mathcal{M}_B$: The join yields the least common refinement of $\mathcal{M}_A$ and $\mathcal{M}_B$. This corresponds to finding the finest partition P such that $P \leq_p P_A$ and $P \leq_p P_B$. It constructs a module that incorporates the combined atomic distinctions present in both $\mathcal{M}_A$ and $\mathcal{M}_B$, effectively 'breaking up' atoms where necessary to achieve a higher granularity that encompasses both input modules.
- $\mathcal{M}_A \wedge \mathcal{M}_B$: The meet yields the greatest common coarsening of $\mathcal{M}_A$ and $\mathcal{M}_B$. This corresponds to finding the coarsest partition P such that $P_A \leq_p P$ and $P_B \leq_p P$. It constructs a module that retains only the knowledge granularity common to both $\mathcal{M}_A$ and $\mathcal{M}_B$, effectively 'merging' atoms that are not distinguished in the same way by both input modules.

These operations allow users to navigate the space of possible modules systematically. If two modules $\mathcal{M}_A$ and $\mathcal{M}_B$ are directly related by the $\leq_M$ ordering, their join and meet simplify according to:

$$\mathcal{M}_A \leq_M \mathcal{M}_B \iff \mathcal{M}_A \vee \mathcal{M}_B = \mathcal{M}_B \iff \mathcal{M}_A \wedge \mathcal{M}_B = \mathcal{M}_A \tag{3}$$

Consider again modules $\mathrm{At}_1, \mathrm{At}_2, \mathrm{At}_3$ from Figure 9. Since $\mathrm{At}_1 \leq_M \mathrm{At}_2$ and $\mathrm{At}_1 \leq_M \mathrm{At}_3$, we have $\mathrm{At}_1 \wedge \mathrm{At}_2 = \mathrm{At}_1$ and $\mathrm{At}_1 \wedge \mathrm{At}_3 = \mathrm{At}_1$. Interestingly, At_2 and At_3 are not directly comparable via $\leq_M$, but their meet is $\mathrm{At}_2 \wedge \mathrm{At}_3 = \mathrm{At}_1$. This shows that At_1 represents the common coarsened knowledge shared between At_2 and At_3.

The lattice structure also illuminates how knowledge lost during coarsening can potentially be recovered. The coarsening of At_2 into At_1 (i.e., $\mathrm{At}_1 = \mathrm{At}_1 \wedge \mathrm{At}_2$) implies knowledge specific to At_2 was lost (specifically, the distinction between Lo and PT). This lost knowledge must be embodied in some other algebraic module(s). Let us call the algebraic module that contains the lost knowledge as At_4, such that $\mathrm{At}_1 \vee \mathrm{At}_4 = \mathrm{At}_2$. Referring to Figure 9, we can see there are two such modules At_4 satisfying this: the one generated from partition $\{\{A \oplus Lo \oplus La\}, \{PT\}\}$ and the one from $\{\{A \oplus Lo\}, \{PT\}, \{La\}\}$. The existence of such At_4 modules demonstrates that the 'lost' information needed to refine At_1 back to At_2 exists within the lattice framework, potentially distributed across multiple alternative modules. This provides a formal basis for understanding and managing knowledge transformations during modularization.

4. Related Work

Ontology modularization techniques are broadly categorized in the literature as graphical or logical Del Vescovo et al. (2010); LeClair, Marinache, Ghalayini, Khedri, and MacCaull (2020). Graphical approaches typically leverage the explicit graph structure of an ontology, extracting modules by traversing relations, clustering concepts, or identifying subgraphs based on specific relational patterns (e.g., Noy and Musen (2009); Stuckenschmidt and Schlicht (2009)). These methods are often favored in workflows requiring rapid, lightweight module extraction, such as for query processing over a focused subset of a larger reference ontology Ahmetaj et al. (2021). While efficient, the primary focus is often on structural connectivity rather than formal guarantees of knowledge preservation.

Logical approaches, which are predominantly developed for Description Logic (DL) based ontologies, aim to extract modules that preserve specific aspects of the original ontology's knowledge, often through notions like conservative extension, locality, or interpolation (e.g., Del Vescovo, Parsia, Sattler, and Schneider (2011); Grau, Horrocks, Kazakov, and Sattler (2009)). These techniques provide stronger semantic guarantees but can be sensitive to the specific DL fragment used. For instance, the complexity and even feasibility of modularization can vary significantly across DLs like $\mathcal{SROIQ}$ Del Vescovo and Penaloza (2014). Furthermore, incorporating data directly into the logical modularization process, or ensuring modules reflect data-driven structures, can present additional challenges, sometimes requiring pre-processing to align with first-order constraints Zhao, Li, and Yang (2021).

The algebraic modularization framework for DIS ontologies presented in this paper offers a distinct perspective that bridges aspects of both graphical and logical approaches, while being intrinsically tied to the underlying data structures via DIS's design. Our core contribution, the definition of an *algebraic module* as a Boolean subalgebra of the original ontology's Boolean algebra, directly addresses knowledge preservation. By ensuring the module is a subalgebra, all Boolean operations (including complement) are preserved, and the module is inherently a conservative extension of the original ontology with respect to its signature. This provides a strong semantic guarantee comparable to those sought by logical approaches. The atoms of a DIS ontology's Boolean lattice are formed from the data layer view, meaning data characteristics are intrinsically factored into the structure from which modules are derived.

The construction of algebraic modules via partitioning the atoms of the source ontology's Boolean algebra allows for a systematic exploration of all possible modules. This contrasts with many graphical techniques that might identify one or a few modules based on local criteria, or PISM construction which generates a specific type of algebraic module by extending a principal ideal. While the use of principal ideals in PISM has parallels with some structural traversal techniques, our overarching framework based on the complete lattice of atom partitions provides a global view of all possible granularities.

The concept of refinement is central to our work. Antoniou and Kehagias Antoniou and Kehagias (2000) define an ontology O' as a refinement of O if O' is a conservative extension of O. Our definition of module refinement ($\mathcal{M}_A$ is a refinement of $\mathcal{M}_B$ if $\mathcal{M}_B$'s Boolean algebra is a subalgebra of $\mathcal{M}_A$'s) aligns directly with this, as being a subalgebra implies a conservative extension. The lattice structure of algebraic modules (Figure 9) provides a formal and operational means to realize such refinements. It allows us not only to compare any two algebraic modules but also to precisely identify the "most refined" module (the original ontology, corresponding to the finest atom partition) and the "least refined" non-trivial module (corresponding to the coarsest non-trivial atom partition). Furthermore, this lattice facilitates the articulation and computation of "least common refinements" or "greatest common coarsenings" of modules using lattice join and meet operations, offering powerful tools for navigating semantic granularity. This systematic characterization of the entire space of algebraically sound modules and their interrelations via refinement is a key differentiator of our approach.

5. Discussion

This paper introduced a novel algebraic approach to ontology modularization within the DIS framework, centered on the concept of modules as Boolean subalgebras of the original ontology's Boolean algebra. This construction, based on partitioning the atoms of the source ontology, yields a complete lattice of all possible algebraic modules. This lattice structure provides a formal means to address semantic granularity, systematically compare the knowledge content of different modules, and construct new modules that precisely refine or coarsen existing knowledge representations. This discussion explores the implications and practical applications of this framework.

A key implication of our algebraic approach is the ability to precisely control and reason about knowledge granularity. In many practical scenarios, not all information in a source ontology is relevant for a specific task, or data attributes might be better understood when grouped. For example, attributes like "created_date" or "last_edited_user" in a dataset, while present, may offer little value for certain analytical tasks. Within our framework, handling such situations corresponds to selecting a specific partition of the original ontology's atoms. Irrelevant atoms can be effectively coarsened into the bottom element (e_C) of the resulting algebraic module, or grouped into a single "don't care" block in the partition, thus removing them from explicit consideration in the module. This leads to a more concise and focused Boolean lattice for the module, without the overhead of unnecessary distinctions.

Beyond simply omitting information, the framework facilitates deliberate knowledge coarsening as a management tool. Consider the attributes *Longitude* and *Latitude*. While distinct atoms in the original ontology, they are often needed together to represent a geographic location. An agent or user can choose an algebraic module where *Longitude* and *Latitude* belong to the same block in the atom partition. This results in a module where $Lo \oplus La$ (Location) is a single atom, as illustrated by modules in the lower parts of the lattice in Figure 9. Such a module, while coarser, might be perfectly suited for tasks requiring only location context, simplifying reasoning and data handling. If, in a rare case, *only* Longitude were needed, the original ontology or a finer-grained algebraic module (where Longitude remains a distinct atom) could be used. The ability to choose the appropriate algebraic module based on the desired atomic structure (granularity) is a direct outcome of our approach. The $\leq_M$ relation (Definition 3.4) on the lattice of algebraic modules then provides a formal way to compare any two such modules in terms of their knowledge granularity.

The lattice of all algebraic modules, derived from the partition lattice of the original ontology's atoms (Section 3.3), offers a complete map of possible granularities. The lattice join ($\vee$) and meet ($\wedge$) operations become powerful tools for dynamic module construction. If an existing set of modules does not precisely meet a user's needs—perhaps one is too coarse, another too fine, or a combination of perspectives is required—these operations allow for the systematic generation of a new algebraic module. For instance, $\mathcal{M}_A \vee \mathcal{M}_B$ yields the least common refinement, incorporating the finest distinctions from both $\mathcal{M}_A$ and $\mathcal{M}_B$. Conversely, $\mathcal{M}_A \wedge \mathcal{M}_B$ provides the greatest common coarsening, capturing only the shared level of detail. This capability allows users or autonomous agents to dynamically interact with and tailor the ontology to specific contextual requirements.

Furthermore, this algebraic formalization allows us to reason about an agent's scope of knowledge. If an agent's interaction with the domain is restricted to a specific algebraic module $\mathcal{M}_P$, its "perception" of the world and its reasoning capabilities are strictly defined by the Boolean algebra $\mathcal{B}_P$ of that module. Concepts that are distinct atoms in the original ontology but are merged into a single atom in $\mathcal{M}_P$ (due to the chosen partition P) become indistinguishable for that agent. The agent cannot access or reason about sub-atomic details that were coarsened away. This provides a formal basis for defining and enforcing knowledge boundaries, which is a step towards addressing what an agent can (or cannot) formally know.

The prospect of constructing any module from the lattice of all algebraic modules raises questions about scalability, given that the number of partitions of n atoms is given by the Bell number (B_n).

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k, \quad \text{with } B_0 = 1. \tag{4}$$

For example, $B_6 = 203$ for our Park example, and $B_{10} = 115,975$. While B_n grows rapidly, several factors mitigate this concern in practice. First, the generation of all set partitions for a given set of atoms is computationally efficient for typical input sizes Orlov (2002). Second, as observed in Vassiliadis, Zarras, and Skoulis (2015) (see Table 2), the majority of database tables (which often form the basis for DIS atoms) have a relatively small number of attributes (typically less than ten). This keeps n relatively small for many practical ontologies. Additionally, modularization is typically goal-driven. Users will seek modules that satisfy specific criteria, such as a particular level of coarseness for certain concepts, the inclusion or exclusion of specific information, or a bounded module size. These constraints drastically prune the search space of relevant algebraic modules. Therefore, while the theoretical number of modules can be large, the combination of realistic data schema sizes and goal-oriented modularization suggests that the growth of the Bell number is not an insurmountable practical limitation for this algebraic approach.

Table 2. Percent of Tables with Number of Attributes (borrowed from Vassiliadis et al. (2015))

	< 5	$5 - 10$	> 10
atlas	10.23%	68.18%	21.59%
biosql	0.00%	24.44%	75.56%
coppermine	17.39%	30.43%	52.17%
ensembl	7.10%	38.06%	54.84%
mediawiki	18.31%	19.72%	61.97%
phpbb	15.71%	44.29%	40.00%
typo3	46.88%	31.25%	21.88%
opencart	9.75%	33.05%	57.20%
Average	17.09%	36.18%	46.73%

In summary, the algebraic modularization framework presented provides a robust and formal foundation for creating, comparing, and manipulating ontology modules. It directly addresses the need for clear semantics of granularity and knowledge transformation, paving the way for more sophisticated knowledge management and reasoning in systems utilizing DIS ontologies.

6. Conclusion

This paper introduced a novel algebraic framework for ontology modularization within the Domain Information System (DIS). By defining modules as Boolean subalgebras derived from partitions of the source ontology's atoms, we have established a principled approach that overcomes ambiguities in prior module definitions and formally characterizes the knowledge preserved or transformed. We demonstrated how the Principal Ideal Subalgebra Module (PISM) extends previous view traversal techniques to ensure algebraic closure, and situated it within the broader and more comprehensive class of general algebraic modules.

A key contribution is the demonstration that the set of all such algebraic modules forms a complete lattice, dually isomorphic to the partition lattice of the ontology's atoms. This lattice structure provides a formal mechanism for addressing semantic granularity: the refinement relation ($\leq_M$) allows for rigorous comparison of modules, while lattice operations (join and meet) enable the systematic construction of new modules with desired levels of coarseness or refinement. This directly addresses the challenge of managing and navigating different views or abstractions of an ontology's knowledge.

The algebraic nature of these modules, particularly their foundation as Boolean subalgebras, ensures that they are conservative extensions of the original ontology (when restricted to the module's signature). This property, reminiscent of desirable features in Description Logic module extraction, is crucial. Furthermore, the framework provides a mathematical basis for analyzing an agent's epistemic state when its knowledge is confined to a specific algebraic module, offering a foundational step towards provable properties concerning what an agent can or cannot infer.

While the number of potential algebraic modules, given by the Bell number, can be large, we have argued that practical considerations such as typical data schema sizes and goal-oriented modularization make this approach computationally feasible, especially with automation. Future work will focus on developing efficient algorithms for navigating this lattice of modules based on user-specified criteria for knowledge preservation or transformation, and further exploring the implications for multi-agent systems and knowledge integration. This work lays a formal groundwork for more robust, adaptable, and semantically well-defined modular ontologies.

References

Ahmetaj, S., Efthymiou, V., Fagin, R., Kolaitis, P. G., Lei, C., Özcan, F., & Popa, L. (2021). Ontology-enriched query answering on relational databases. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, pp. 15247–15254).

Antoniou, G., & Kehagias, A. (2000). A note on the refinement of ontologies. *International Journal of Intelligent Systems*, *15*(7), 623–632.

Barton, A., Tarbouriech, C., Vieu, L., & Ethier, J.-F. (2022). A mereological system for informational templates. In *6th workshop on foundational ontology (foust 2022)@ the 8th joint ontology workshops (jowo 2022)* (pp. paper–4).

Birkhoff, G. (1940). *Lattice theory* (Vol. 25). American Mathematical Soc.

Bittner, T. (2004). A mereological theory of frames of reference. *International Journal on Artificial Intelligence Tools*, *13*(01), 171–198.

Bittner, T., Donnelly, M., & Winter, S. (2005). Ontology and semantic interoperability. In *Large-scale 3d data integration* (pp. 139–160). CRC Press.

Calosi, C., & Graziani, P. (2014). *Mereology and the sciences.* Springer.

Davey, B. A., & Priestley, H. A. (2002). *Introduction to lattices and order.* Cambridge university press.

Del Vescovo, C., Horridge, M., Parsia, B., Sattler, U., Schneider, T., & Zhao, H. (2020). Modular structures and atomic decomposition in ontologies. *Journal of Artificial Intelligence*

Research, *69*, 963–1021.

Del Vescovo, C., Parsia, B., Sattler, U., & Schneider, T. (2010). The modular structure of an ontology: an empirical study. In *23rd international workshop on description logics dl2010* (p. 232).

Del Vescovo, C., Parsia, B., Sattler, U., & Schneider, T. (2011). The modular structure of an ontology: Atomic decomposition and module count. In *Womo* (pp. 25–39).

Del Vescovo, C., & Penaloza, R. (2014). Dealing with ontologies using cods. In *Description logics* (pp. 157–168).

Fouad, M. M., Mahmood, M. A., Mahmoud, H., Mohamed, A., & Hassanien, A. E. (2014). Intelligent road surface quality evaluation using rough mereology. In *2014 14th international conference on hybrid intelligent systems* (pp. 18–22).

Grau, B. C., Horrocks, I., Kazakov, Y., & Sattler, U. (2009). Extracting modules from ontologies: A logic-based approach. In *Modular ontologies* (pp. 159–186). Springer.

Guarino, N., Oberle, D., & Staab, S. (2009). What is an ontology? In *Handbook on ontologies* (pp. 1–17). Springer.

Guizzardi, G., & Guarino, N. (2024). Explanation, semantics, and ontology. *Data & Knowledge Engineering*, *153*, 102325.

Halmos, P. R. (2018). *Lectures on boolean algebras.* Courier Dover Publications.

Harding, J., Heunen, C., Lindenhovius, B., & Navara, M. (2017). Boolean subalgebras of orthoalgebras. *Order*, 1–47.

Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., & Michalek, H. (2006). General formal ontology (gfo). *Part I: Basic Principles. Onto-Med Report*, *8*.

Hirsch, R., & Hodkinson, I. (2002). *Relation algebras by games.* Elsevier.

Horrocks, I., Li, L., Turi, D., & Bechhofer, S. (2004). The instance store: Dl reasoning with large numbers of individuals. In *Proc. of the 2004 description logic workshop (dl 2004)* (pp. 31–40).

Kleshchev, A. S., & Artemjeva, I. L. (2005). How can domain ontologies relate to one another? In *Xi-th international conference* (Vol. 1, p. 132).

Krieger, H.-U., & Willms, C. (2015). Extending owl ontologies by cartesian types to represent n-ary relations in natural language. In *Proceedings of the 1st workshop on language and ontologies.*

LeClair., A., Khedri., R., & Marinache, A. (2019). Formalizing graphical modularization approaches for ontologies and the knowledge loss. In *International joint conference on knowledge discovery, knowledge engineering, and knowledge management.*

LeClair, A., Marinache, A., El Ghalayini, H., MacCaull, W., & Khedri, R. (2022). A review on ontology modularization techniques-a multi-dimensional perspective. *IEEE Transactions on Knowledge and Data Engineering*, *35*(5), 4376–4394.

LeClair, A., Marinache, A., Ghalayini, H., Khedri, R., & MacCaull, W. (2020). A systematic literature review and discussion on ontology modularization techniques. *IEEE Transactions on Knowledge and Data Engineering.* ((Review))

Lutz, C., Walther, D., & Wolter, F. (2007). Conservative extensions in expressive description logics. In *Ijcai* (Vol. 7, pp. 453–458).

Marinache, A. (2016). *On the structural link between ontologies and organised data sets* (Unpublished master's thesis).

Noy, N., & Musen, M. (2009). Traversing ontologies to extract views. *Modular Ontologies*, 245–260.

Orlov, M. (2002). Efficient generation of set partitions. *Engineering and Computer Sciences, University of Ulm, Tech. Rep.*

Pinkel, C., Binnig, C., Jiménez-Ruiz, E., Kharlamov, E., Nikolov, A., Schwarte, A., … Kraska, T. (2017). Incmap: A journey towards ontology-based data integration. *Lecture Notes in Informatics (LNI), Proceedings-Series of the Gesellschaft fur Informatik (GI)*, *265*, 145–164.

Pliatsios, A., Kotis, K., & Goumopoulos, C. (2023). A systematic review on semantic interoperability in the ioe-enabled smart cities. *Internet of Things*, *22*, 100754.

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., & Rosati, R. (2008). Linking data to ontologies. In *Journal on data semantics x* (pp. 133–173). Springer.

Recreation and parks properties. (n.d.). https://data.sfgov.org/Culture-and-Recreation/Recreation-and-Par (Accessed: 2021-07-18)

Sikorski, R. (1969). *Boolean algebras* (Vol. 2). Springer.

Stuckenschmidt, H., & Schlicht, A. (2009). Structure-based partitioning of large ontologies. In *Modular ontologies* (pp. 187–210). Springer.

Vassiliadis, P., Zarras, A. V., & Skoulis, I. (2015). How is life for a table in an evolving relational schema? birth, death and everything in between. In *International conference on conceptual modeling* (pp. 453–466).

Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., & Zakharyaschev, M. (2018). Ontology-based data access: A survey..

Zhao, X., Li, F., & Yang, H. (2021). Extracting satisfiability-preserving modules from the owl rl ontology for efficient reasoning. *IEEE Access*, *9*, 30833–30844.