April 2024

The Semantic Immune System: Adaptive Knowledge Integrity in the Era of Induced Semantics

Manzoor ALI [1] , Muhammad SALEEM , Yasir MAHMOOD , and Axel-Cyrille Ngonga NGOMO

[a] *Data Science Group (DICE), Heinz Nixdorf Institute, Paderborn University, Germany*

ORCiD ID: Manzoor Ali https://orcid.org/0000-0001-8403-5160, Muhammad Saleem https://orcid.org/0000-0001-9648-5417, Yasir Mahmood https://orcid.org/0000-0002-5651-5391, Axel-Cyrille Ngonga Ngomo https://orcid.org/0000-0001-7112-3516

Abstract. Knowledge graphs are increasingly populated by large language models (LLMs), which generate induced semantics at scale. However, LLM-generated knowledge also introduces hallucinations, contradictions, and semantic drift that static validation approaches such as SHACL and OWL are often insufficient to adequately address. We propose the **Semantic Immune System (SIS)**, a logic-based framework for autonomous knowledge graph validation and repair. Inspired by biological immune systems, SIS combines deterministic constraints with adaptive and evolutionary mechanisms to detect, remember, and mitigate semantic anomalies in dynamically evolving graphs. We provide a formalization, a worked example, and a discussion of feasibility and oracle assumptions. The framework offers a blueprint for building autonomous KG repair systems that balance correctness, robustness, and completeness.

Keywords. Induced Semantics, Neuro-Symbolic AI, Artificial Immune Systems

1. Introduction

Knowledge graphs (KGs) are central to modern data integration, search, and decision support [1]. Their quality, particularly correctness and completeness, is critical because errors or missing information can directly affect downstream analytics, recommendation systems, reasoning tasks, and automated decision-making processes [2]. Traditionally, KGs are built by domain experts using curated semantics – manually designed ontologies and clean data. This approach ensures high fidelity but suffers from the *knowledge acquisition bottleneck* [3,4]. The recent success of large language models (LLMs) has enabled *induced semantics*: automatic extraction of triples from unstructured text at web scale [5]. While this solves the scalability problem, it introduces new types of defects: factual hallucinations, logical contradictions, outdated information, and subtle semantic drift [6].

[1] Corresponding Author: Manzoor Ali; E-mail: ali.manzoor@upb.de

Existing validation methods such as SHACL [7] and OWL [8] constraints remain effective for enforcing structural consistency, schema compliance, and type safety. However, they are insufficient for detecting semantically plausible but factually incorrect assertions generated by probabilistic systems such as LLMs. The limitation is fundamental rather than being implementation-specific. Ontology constraints operate primarily over consistency, type compatibility, and explicit logical entailment. LLM-generated assertions, however, often fail at the level of world knowledge, contextual plausibility, or temporal validity while remaining logically consistent with the ontology [6,9]. For example, the triple "Marie Curie won the Nobel Prize in Literature" is structurally valid (a person winning a prize) but factually false. No static constraint can capture such domain-specific falsehoods. Existing semantic validation assumes that knowledge graphs evolve slowly and that semantic boundaries remain largely stable once encoded in ontologies and constraints. Induced semantics fundamentally changes this assumption. Modern KGs are no longer static curated artifacts, but continuously evolving semantic ecosystems populated by probabilistic assertions generated at machine scale. In such environments, validation cannot remain purely rule-based and reactive. Instead, semantic infrastructures must become adaptive, self-monitoring, and capable of learning new classes of semantic failure over time. This shift motivates moving from static validation toward autonomous semantic defense systems.

To mitigate these shortcomings, we propose SIS, inspired by biological immune systems, SIS treats a KG as a living semantic ecosystem and hallucinations as semantic pathogens that must be detected, remembered, and neutralized. The SIS combines lightweight structural validation using SHACL/OWL as a first-line sanity check with an adaptive layer based on negative selection, where logical detectors evolve to recognize previously unseen hallucination patterns. Detector quality is further improved through *clonal selection* using oracle validation, human feedback, or contradiction signals, while semantic memory stores previously observed *pathogen signatures* for rapid secondary response. Detected semantic pathogens can subsequently trigger repair actions, ranging from triple removal and confidence adjustment to targeted regeneration and human review. In addition, *semantic vaccination* proactively injects synthetic hallucinations to pre-train the system against emerging failure modes.

Unlike prior artificial immune system approaches for databases or networks [10,11], the SIS targets LLM-induced errors in KGs. We formalize the notions of Semantic Self and induced semantics, introduce adaptive detector generation and evolution mechanisms, propose semantic vaccination for proactive defense, and discuss feasibility under different oracle assumptions and autonomy settings.

Why Static Validation Breaks Under Induced Semantics? Classical ontology validation assumes that the semantic space of a KG is relatively bounded and evolves under explicit human supervision. Under this assumption, domain constraints and logical entailment are sufficient to preserve integrity. However, induced semantics generated by LLMs introduce probabilistic assertions whose failure modes are often semantically plausible and are hence structurally valid. This distinction is critical as hallucinated triples frequently satisfy all syntactic and ontological constraints while still being factually incorrect. As a result, traditional validation mechanisms remain effective for detecting structural inconsistencies but become increasingly ineffective against semantic pathogens that emerge from statistical generation processes.

The problem is further amplified by the dynamic nature of induced semantics. As LLMs continuously generate and refine assertions, semantic drift can accumulate gradually across a KG without triggering explicit logical contradictions. In this setting, validation becomes an ongoing adaptive process rather than a one-time verification step. Consider an enterprise-scale KG continuously updated by autonomous AI agents extracting knowledge from scientific literature, internal reports, and web sources. In such environments, millions of probabilistic assertions may be generated daily, rendering manual validation infeasible. The SIS envisions adaptive semantic infrastructures capable of continuously monitoring, filtering, and repairing generated knowledge before erroneous assertions propagate into downstream analytics or decision systems.

2. Preliminaries and Definitions

We use standard notation in Description Logic (DL) [12,13] and knowledge graphs [14]. A DL knowledge base (KB) $\mathscr{K} = \langle \mathscr{T}, \mathscr{A} \rangle$ is a tuple where $\mathscr{T}$ is a TBox (terminological axioms) and $\mathscr{A}$ is an ABox (assertions). The TBox $\mathscr{T}$ contains general concept inclusions (GCIs) of the form $C \sqsubseteq D$, where C, D are concepts. The ABox includes assertions having the form $C(a)$ (concept assertion) or $r(a,b)$ (role assertion), for individuals a, b, concept C, and role r. An interpretation $\mathscr{I}$ maps concepts to sets and roles to relations. A knowledge base is *consistent* if it has a model.

A KB can be naturally represented as a knowledge graph [15], which encodes the assertions and logical entailments of the input KB, facilitating accurate downstream prediction tasks. In the remainder of the paper, we use the graph representation and refer to it as a KG. Consequently, we abstract over the underlying representation and consider the KG view consisting of triples (s, p, o). We denote by G the KG induced from $\mathscr{K}$.

We assume the existence of an **oracle** Or that, given a triple, returns `true` if the triple is deemed factually correct according to a trusted source of domain knowledge, and `false` otherwise. The oracle additionally assigns a confidence score in its evaluation and can represent a domain expert, curated reference source, or hybrid verification pipeline. In contrast to the DL reasoners, the oracle is not restricted to checking logical entailment from the current KG but may validate facts against external ground truth. In practice, oracles can be imperfect; we relegate a detailed discussion on this aspect to Section 4.

Definition 2.1 (Semantic Self)**.** *Let G be a KG, Or be an oracle and $\theta_{self} \in [0,1]$ be a confidence threshold. The* Semantic Self $\mathscr{S} \subseteq G$ *of G is defined as:*

$$\mathscr{S} = \{\, \tau \in G \mid Or(\tau) = true \wedge conf(\tau) > \theta_{self} \,\}.$$

While we adopt triples for clarity, real-world KGs qualify statements with temporal validity and provenance. Our model extends seamlessly to contextualized statements (s, p, o, c), where c captures these dimensions; the oracle and detectors operate over this enriched representation. The remaining triples in G may be unverified or erroneous.

Definition 2.2 (Induced Semantics)**.** *Let $\mathscr{M}$ be an LLM or any KG extraction pipeline. The set of* induced triples $\mathscr{I}$ *is:*

$$\mathscr{I} = \{\, \tau \mid \tau \sim \mathscr{M}(corpus) \,\}.$$

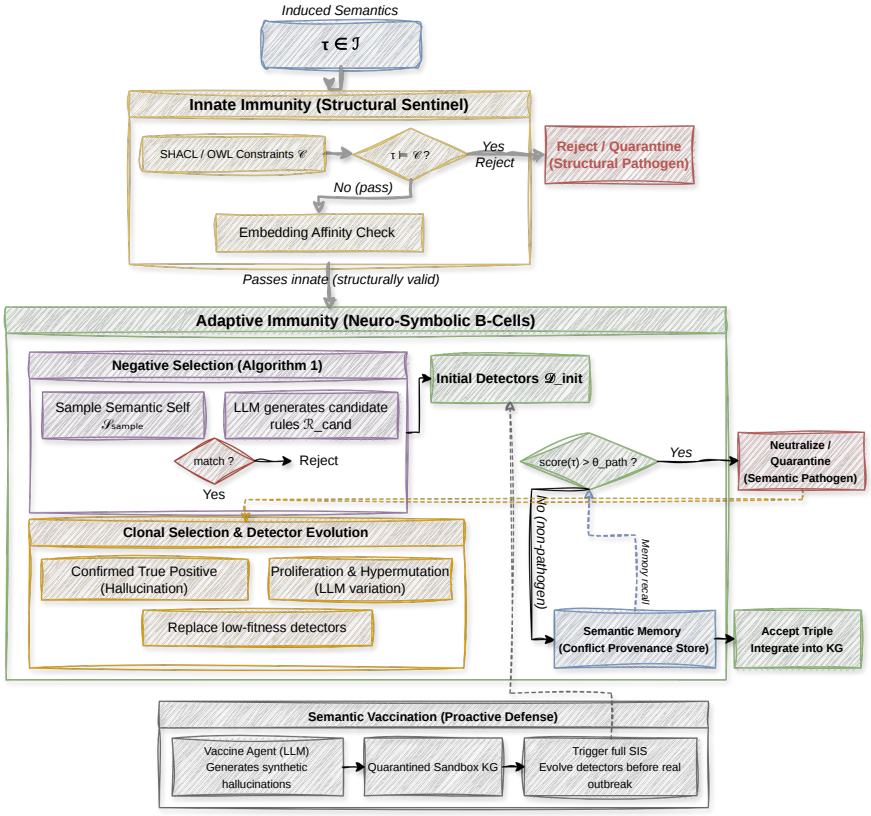

Figure 1. Architecture of the SIS with innate and adaptive validation, clonal selection, semantic memory, and semantic vaccination.

We next define *semantic pathogens*, defined as induced triples that are assessed as false by the oracle. That is, an induced triple $\tau \in \mathscr{I}$ is a *semantic pathogen* if $Or(\tau) =$ false. A pathogen may be the result of an LLM hallucination, a contradiction to already existing facts, or any fact that does not hold in the domain.

The goal of the SIS is to decide for each $\tau \in \mathscr{I}$ whether it should be accepted (added to the KG) or rejected as a semantic pathogen.

Example 2.1 (Running Example)**.** *Consider a KG with Semantic Self triples:*

$$\mathscr{S} = \{(MarieCurie, wonPrize, NobelPrizePhysics), (MarieCurie, hasField, Physics)\}.$$

Suppose an LLM induces the triple $\tau = (MarieCurie, wonPrize, NobelPrizeLiterature)$. *Since Marie Curie never won the Literature prize,* $Or(\tau) =$ *false, so* τ *is a pathogen.*

3. Formal Framework for the Semantic Immune System

3.1. Innate Immunity

Figure 1 shows the overall architecture of SIS. The innate layer consists of a set of deterministic *base* constraints $\mathscr{C}$ expressed in SHACL or OWL. These constraints define the structural boundaries of the semantic self. Let K denote the current knowledge graph state (initially derived from the input KB $\mathscr{K}$). For a triple τ, we define the innate response as:

$$\mathscr{R}_{\text{innate}}(\tau) = \begin{cases} 1 & \text{if } K \cup \{\tau\} \models \mathscr{C} \\ 0 & \text{otherwise} \end{cases}$$

If the triple violates any constraint (e.g., domain/range, cardinality), it is immediately quarantined.

Embedding-Based Affinity Let $\phi : G \to \mathbb{R}^d$ be an embedding function (e.g., RDF2Vec [16]) that maps triples from the knowledge graph G to vectors. We define the semantic similarity between two triples as cosine similarity. We then define the affinity of an induced triple τ to the Semantic Self $\mathscr{S}$ as:

$$\mathscr{A}(\tau, \mathscr{S}) = \max_{\tau' \in \mathscr{S}} \text{sim}(\tau, \tau').$$

If $\mathscr{A}(\tau, \mathscr{S}) < \theta_{\text{affinity}}$, the triple is flagged as a potential pathogen. We use this affinity measure only to guide detector generation, not final decisions.

3.2. Negative Selection for Detector Generation

In biological immune systems, negative selection produces T-cells that react only to non-self by eliminating those that react to self. We apply the same principle to generate *semantic detectors* – logical rules that fire when presented with a pathogen.

Definition 3.1. *A* detector *d is a pair (r, w) where r is a logical condition and $w \in [0,1]$ is a weight. We say that a detector $d = (r, w)$* matches *a triple τ if $r(\tau)$ evaluates to true.*

Example 3.1. *Consider a detector rule encoding improbable prize-field combinations:*

```
ASK WHERE { ?x wonPrize NobelPrizeLiterature . ?x hasField Physics . }
```

Applying this to the running example triple $\tau = (MarieCurie, wonPrize, NobelPrizeLiterature)$, the rule partially matches the pattern, but full activation requires a conflicting field assertion such as Physics, illustrating how composite detectors operate.

We present a dedicated algorithm (Alg. 1) that generates the initial detector set $\mathscr{D}_{\text{init}}$.

By construction, the detectors that react to the triples in the sampled semantic self are discarded during negative selection. Therefore, the resulting detector set preferentially targets triples in the non-self region of the semantic space (i.e., $G \setminus \mathscr{S}$) while minimizing interference with the semantic self.

Algorithm 1 Negative Selection for Detector Generation

Require: Semantic Self $\mathscr{S}$, embedding ϕ, threshold θ_{self}, candidate rule count N
Ensure: Initial detector set $\mathscr{D}_{\text{init}}$

$\mathscr{D}_{\text{init}} \leftarrow \emptyset$
Generate candidate rule templates $\mathscr{R}_{\text{cand}}$ using an LLM prompt based on $\mathscr{S}$
for each $r \in \mathscr{R}_{\text{cand}}$ **do**
    Sample $\mathscr{S}_{\text{sample}} \subset \mathscr{S}$ of size k
    if $\exists \tau \in \mathscr{S}_{\text{sample}}$ with $r(\tau)$ true **then**
        Reject r ▷ reacts to self
    else
        $\mathscr{D}_{\text{init}} \leftarrow \mathscr{D}_{\text{init}} \cup \{(r, w_{\text{init}})\}$
    end if
end for
▷ Also generate random non-self detectors
for $i = 1$ to N_{rand} **do**
    Sample random vector $\mathbf{v} \sim \mathscr{U}(\mathbb{R}^d)$
    Find nearest self triple $\tau_{\text{self}} = \arg\max_{\tau' \in \mathscr{S}} \text{sim}(\mathbf{v}, \phi(\tau'))$
    if $\text{sim}(\mathbf{v}, \phi(\tau_{\text{self}})) < \theta_{\text{self}}$ **then**
        Convert $\mathbf{v}$ to a rule r via an LLM (describe the region)
        if r does not match any $\tau \in \mathscr{S}_{\text{sample}}$ **then**
            $\mathscr{D}_{\text{init}} \leftarrow \mathscr{D}_{\text{init}} \cup \{(r, w_{\text{rand}})\}$
        end if
    end if
end forreturn $\mathscr{D}_{\text{init}}$

3.3. Pathogen Recognition

Given an incoming induced triple τ, the SIS computes a pathogen score:

$$\text{score}(\tau) = \frac{1}{|\mathscr{D}_{\text{init}}|} \sum_{d \in \mathscr{D}_{\text{init}}} w_d \cdot \mathbf{1}[r_d(\tau)].$$

If $\text{score}(\tau) > \theta_{\text{pathogen}}$, the triple is flagged as a pathogen and neutralized (quarantined or discarded).

3.4. Clonal Selection and Detector Evolution

To adapt to new hallucination patterns, the SIS employs clonal selection [17]. When a detector d contributes to a true positive (correctly flagging a confirmed hallucination), the system creates c clones and mutates them using an LLM. The mutation explores the semantic neighborhood of the given pathogen (e.g., by changing either the relation or the concept name of the triple). The fitness of a mutated detector d' is evaluated on a validation set:

$$\text{fitness}(d') = \frac{\text{TP}(d', \mathscr{H}_{\text{val}})}{\text{TP}(d', \mathscr{H}_{\text{val}}) + \text{FP}(d', \mathscr{S}_{\text{val}})},$$

where $\mathscr{H}_{\text{val}}$ are the known hallucinations and $\mathscr{S}_{\text{val}}$ is a self-subset. Only detectors with a fitness score above a threshold (e.g., 0.7) replace the least fit detectors (elitist selection).

3.5. Semantic Memory and Secondary Response

After a pathogen is neutralized , the SIS creates a **memory record**, depicted as follows.

$$\text{mem}(\tau) = (\tau,\ \text{source\_LLM},\ \{d \in \mathscr{D} \mid \text{match}(d,\tau)\},\ \text{conflicting\_triples}).$$

When a new triple τ' arrives, the system first checks memory using embedding similarity. To counter semantic drift, each memory record decays with factor λ; unreinforced records are pruned, ensuring the system adapts when the oracle or domain knowledge evolves.

$$\mathscr{M}_{\text{recall}}(\tau') = \mathbf{1}\left[\max_{\text{mem}(\tau)\in\text{Memory}} \text{sim}(\tau',\tau) > \theta_{\text{mem}}\right].$$

If a similar pathogen was seen before, τ' is neutralized immediately without invoking the full adaptive cycle.

3.6. Semantic Vaccination

We introduce a proactive mechanism: **semantic vaccination** A vaccine agent (a specialized LLM) generates synthetic hallucinations that are structurally plausible but factually false, targeting known weaknesses of the current detector set. These synthetic pathogens are injected into a quarantined sandbox KG, triggering the full negative selection and clonal selection cycles. By the time a real large-scale hallucination event occurs, the SIS already has high-affinity detectors ready.

Example 3.2 (Vaccination for the Running Example)**.** *The vaccine agent might generate synthetic triples. Specifically, it mutates triples via relation substitution, entity substitution, and temporal inconsistency, ensuring broad coverage of plausible falsehoods, like "Albert Einstein won the Nobel Prize in Literature" or "Marie Curie discovered the theory of relativity". Hence, the SIS would process them in a sandbox, evolving detectors that generalize to category errors (such as prizes in an unrelated field or false scientific attributions).*

Conceptually, the SIS operates as a continuous adaptive cycle. Consider a sandbox environment in which an initial set of synthetic hallucinations is injected into the KG. During early iterations, only a subset of pathogens may be recognized by existing detectors. However, confirmed detections trigger clonal expansion and mutation, producing increasingly specialized detectors that generalize across related hallucination patterns. At the same time, semantic memory enables previously observed hallucination pattern families to be recognized with decreasing response latency. Over successive vaccination cycles, the detector population evolves toward broader semantic coverage and faster secondary response, analogous to adaptive immunity in biological systems.

4. Trade-offs and Feasibility Discussion

Oracle Quality: The definitions above assume an oracle that always returns the correct truth value. In practice, an ideal oracle does not exist. Following [18], we distinguish

several settings, including an all-knowing oracle (ideal but unrealistic), a limited all-knowing oracle that provides correct answers only for a subset of triples, a noisy oracle with a known error rate, and the absence of an oracle in fully automated settings. The SIS can operate under all these scenarios, with repair quality degrading gracefully as oracle reliability decreases. In the presence of noisy oracles, voting mechanisms and confidence thresholds can be used to mitigate errors.

Scalability and Autonomy. The SIS is intended as a scalable architectural framework. Negative selection operates on samples of the Semantic Self, while embedding-based affinity and memory recall can leverage approximate nearest-neighbor indices such as FAISS [19], enabling efficient operation on large KGs. Computationally intensive components, including clonal evolution and semantic vaccination, can execute asynchronously, keeping the core validation pipeline lightweight. Following [20], the SIS also supports different autonomy levels by allowing repairs to be computed using either local knowledge (KB_O) or the entire KG (KB_{KG}), while retaining either local (AS_O) or global (AS_{KG}) axioms, thereby balancing repair quality, scalability, and ownership constraints.

5. Related Work

KG validation and repair: Early work focused on debugging incoherent ontologies using justifications and hitting sets [21,22]. More recent approaches use abduction for missing axioms [23] or axiom weakening to avoid complete removal [24,25]. To the best of our knowledge, none of these address LLM-induced hallucinations or use negative selection.

Artificial immune systems: Forrest et al. [10] introduced negative selection for computer security. Dasgupta [11] surveyed AIS applications. However, these were applied to network intrusion or fault detection, not to knowledge graphs or semantic hallucinations.

LLM hallucination detection: Recent surveys [6] classify hallucination mitigation strategies, but largely focus on natural language generation rather than KG integration. To our knowledge, SIS is the first framework combining neuro-symbolic embeddings, negative selection, and semantic vaccination for KG integrity. While existing approaches address ontology repair, hallucination mitigation, or anomaly detection separately, SIS unifies self/non-self discrimination, evolutionary detector generation, semantic memory, and proactive vaccination within a single adaptive framework for autonomous KG integrity management.

6. Conclusion

In this paper, we presented the SIS, a framework for detecting and mitigating LLM-induced hallucinations in knowledge graphs through layered structural validation, adaptive detector generation, clonal refinement, semantic memory, and proactive vaccination. The framework formalizes the notions of Semantic Self and induced semantics while addressing practical aspects such as oracle quality, scalability, and autonomy. Future work will benchmark SIS on Wikidata with injected hallucinations, measuring F1 against SHACL and embedding baselines via ablation studies, while addressing oracle drift through the decay mechanisms.

Declaration on Generative AI

GPT-4 was used to improve the quality of the text and Scholar Lab was used to search for related work. All content of the paper was made by the authors.

References

[1] Purohit D, Chudasama Y, Vidal M. VANILLA: Validated knowledge graph completion - A Normalization-based framework for Integrity, Link prediction, and Logical Accuracy. Knowl Based Syst. 2025;325:113939. Available from: https://doi.org/10.1016/j.knosys.2025.113939.

[2] Marchesin S, Silvello G, Alonso O. Large Language Models and Data Quality for Knowledge Graphs. Information Processing & Management. 2025;62(6):104281. Available from: https://www.sciencedirect.com/science/article/pii/S0306457325002225.

[3] Gruber TR. A translation approach to portable ontology specifications. Knowledge acquisition. 1993;5(2):199-220.

[4] Noy NF, McGuinness DL, et al.. Ontology development 101: A guide to creating your first ontology. Stanford knowledge systems laboratory technical report KSL-01-05 and . . . ; 2001.

[5] Pan JZ, Razniewski S, Kalo J, Singhania S, Chen J, Dietze S, et al. Large Language Models and Knowledge Graphs: Opportunities and Challenges. TGDK. 2023;1(1):2:1-2:38. Available from: https://doi.org/10.4230/TGDK.1.1.2.

[6] Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, et al. Survey of Hallucination in Natural Language Generation. ACM Comput Surv. 2023;55(12):248:1-248:38. Available from: https://doi.org/10.1145/3571730.

[7] Knublauch H, Kontokostas D. Shapes Constraint Language (SHACL). World Wide Web Consortium; 2017. REC-shacl-20170720. Available from: https://www.w3.org/TR/shacl/.

[8] Group WOW. OWL 2 Web Ontology Language: Document Overview (Second Edition). World Wide Web Consortium; 2012. REC-owl2-overview-20121211. Available from: https://www.w3.org/TR/owl2-overview/.

[9] Huang L, Yu W, Ma W, Zhong W, Feng Z, Wang H, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems. 2025;43(2):1-55.

[10] Forrest S, Perelson AS, Allen L, Cherukuri R. Self-nonself discrimination in a computer. In: 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, May 16-18, 1994. IEEE Computer Society; 1994. p. 202-12. Available from: https://doi.org/10.1109/RISP.1994.296580.

[11] Overill RE. Book Review: "Artificial Immune Systems and their Applications" by D. Dasgupta. J Log Comput. 2001;11(6):961-2. Available from: https://doi.org/10.1093/logcom/11.6.961-a.

[12] Bremer M. Book Reviews: Franz Baader et al. (eds.), The Description Logic Handbook, Cambridge: Cambridge University Press, 2003, xvii+555 pp., $130, ISBN 0-52178-176-0. Minds Mach. 2005;15(1):123-6. Available from: https://doi.org/10.1007/s11023-004-4929-2.

[13] Hitzler P, Krotzsch M, Rudolph S. Foundations of semantic web technologies. Chapman and Hall/CRC; 2009.

[14] Hogan A, Blomqvist E, Cochez M, D'amato C, Melo GD, Gutierrez C, et al. Knowledge Graphs. ACM Comput Surv. 2021 Jul;54(4). Available from: https://doi.org/10.1145/3447772.

[15] Teyou LMK, Friedrichs L, Kouagou NJ, Demir C, Mahmood Y, Heindorf S, et al. Neural Reasoning for Robust Instance Retrieval in SHOIQ. In: Shimizu C, Ferrada S, Kagal L, editors. Proceedings of the 13th Knowledge Capture Conference 2025, K-CAP 2025, Dayton, Ohio, USA, December 10-12, 2025. ACM; 2025. p. 61-8. Available from: https://doi.org/10.1145/3731443.3771348.

[16] Ristoski P, Paulheim H. RDF2Vec: RDF Graph Embeddings for Data Mining. In: Groth P, Simperl E, Gray AJG, Sabou M, Krötzsch M, Lécué F, et al., editors. The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I. Lecture Notes in Computer Science; 2016. p. 498-514. Available from: https://doi.org/10.1007/978-3-319-46523-4_30.

[17] de Castro LN, Zuben FJV. Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput. 2002;6(3):239-51. Available from: https://doi.org/10.1109/TEVC.2002.1011539.

[18] Lambrix P. Completing and Debugging Ontologies: State-of-the-art and Challenges in Repairing Ontologies. ACM J Data Inf Qual. 2023;15(4):41:1-41:38. Available from: https://doi.org/10.1145/3597304.

[19] Douze M, Guzhva A, Deng C, Johnson J, Szilvasy G, Mazaré PE, et al. The Faiss Library. IEEE Transactions on Big Data. 2025;12(2):346-61.

[20] Li Y, Lambrix P. Repairing Networks of ${\mathcal{E}\mathcal{L}}_{\perp }$ Ontologies Using Weakening and Completing. In: Demartini G, Hose K, Acosta M, Palmonari M, Cheng G, Skaf-Molli H, et al., editors. The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part I. Lecture Notes in Computer Science. Springer; 2024. p. 107-25. Available from: https://doi.org/10.1007/978-3-031-77844-5_6.

[21] Schlobach S, Cornet R. Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In: Gottlob G, Walsh T, editors. IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003. Morgan Kaufmann; 2003. p. 355-62. Available from: http://ijcai.org/Proceedings/03/Papers/053.pdf.

[22] Kalyanpur A, Parsia B, Sirin E, Hendler JA. Debugging unsatisfiable classes in OWL ontologies. J Web Semant. 2005;3(4):268-93. Available from: https://doi.org/10.1016/j.websem.2005.09.005.

[23] Wei-Kleiner F, Dragisic Z, Lambrix P. Abduction Framework for Repairing Incomplete EL Ontologies: Complexity Results and Algorithms. In: Brodley CE, Stone P, editors. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada. AAAI Press; 2014. p. 1120-7. Available from: https://doi.org/10.1609/aaai.v28i1.8858.

[24] Baader F, Kriegel F, Nuradiansyah A, Peñaloza R. Making Repairs in Description Logics More Gentle. In: Thielscher M, Toni F, Wolter F, editors. Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018. AAAI Press; 2018. p. 319-28. Available from: https://aaai.org/papers/36-18056-making-repairs-in-description-logics-more-gentle/.

[25] Li Y, Lambrix P. Repairing *EL* Ontologies Using Weakening and Completing. In: Pesquita C, Jiménez-Ruiz E, McCusker JP, Faria D, Dragoni M, Dimou A, et al., editors. The Semantic Web - 20th International Conference, ESWC 2023, Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings. Lecture Notes in Computer Science. Springer; 2023. p. 298-315. Available from: https://doi.org/10.1007/978-3-031-33455-9_18.