The Observer–Situation Lattice: A Unified Formal Basis for Perspective-Aware Cognition

AAAI Track

Saad Alqithami
Computer Science Department, Al-Baha University
Albaha 65779, Saudi Arabia
salqithami@bu.edu.sa

ABSTRACT

Autonomous agents operating in complex multi-agent environments must reason about what holds from multiple perspectives. Existing approaches often struggle to integrate reasoning across agents and contexts, typically handling these dimensions in separate, specialized modules. This fragmentation can yield brittle and hard-to-maintain reasoning pipelines, particularly when agents must represent and query the beliefs of others (Theory of Mind). We introduce the Observer–Situation Lattice (OSL), a mathematical structure that provides a single, coherent semantic space for perspective-aware cognition. OSL is a finite complete lattice whose elements represent observer–situation pairs, enabling a principled and scalable approach to belief management. We present two key algorithms that operate on this lattice: (i) Relativized Belief Propagation, an incremental update algorithm that efficiently propagates new information, and (ii) Minimal Contradiction Decomposition, a graph-based procedure that identifies and isolates contradiction components. We establish formal guarantees for the lattice construction and the correctness and complexity of the proposed algorithms, and we demonstrate practical utility through benchmarks including classic Theory of Mind tasks and comparisons with established paradigms such as assumption-based truth maintenance systems. Our results show that OSL provides a computationally efficient and expressive foundation for building robust, perspective-aware autonomous agents.

KEYWORDS

Observer-Dependent Semantics, Explainable Cognitive Agents, Belief-Desire-Intention (BDI), Global Workspace Architecture, Theory of Mind Planning, Context-Aware Perception, Meta-Cognitive Goal Reasoning, Neurosymbolic AI

ACM Reference Format:
Saad Alqithami. 2026. The Observer–Situation Lattice: A Unified Formal Basis for Perspective-Aware Cognition: AAAI Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/CHZG9392

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/CHZG9392

1 INTRODUCTION

Autonomous agents deployed in real-world, multi-agent settings must navigate a complex information landscape where truth is relative. An observation made by one agent may not be available to another; a fact true in one context may be false in another. For example, in a smart building, a maintenance robot may observe a wet floor, while a remote supervisor sees only an aggregated alert, and a security operator with access to different camera feeds perceives nothing amiss. Effective collaboration in such environments requires agents to reason not just about the world, but about the perspectives of others—a capability known as *Theory of Mind* (ToM) [19]. This fundamental challenge of perspective-aware reasoning is a critical bottleneck in the development of robust, socially intelligent AI systems [13].

Existing agent architectures and formalisms struggle to address this challenge in a unified and scalable manner. Mainstream approaches like the Belief-Desire-Intention (BDI) model [28, 29] provide a strong foundation for single-agent deliberation but typically assume a single, globally consistent belief state, making it difficult to manage the multiple, often conflicting, viewpoints inherent in multi-agent systems. While various BDI programming languages and platforms exist [10, 27, 31], they often treat perspective-taking as an add-on rather than a core architectural principle. Similarly, foundational work in epistemic logic [4, 16, 20, 33, 34] provides a formal language for reasoning about knowledge and belief, but its computational complexity often renders it impractical for dynamic, real-time applications [3, 6]. Other approaches, such as multi-context systems [8] and truth maintenance systems [11, 12, 15], offer powerful tools for managing different belief sets but lack a unifying semantic structure that can seamlessly integrate the notions of observer, context, and time.

This fragmentation of approaches leads to a critical research gap: the absence of a computationally efficient, formally grounded framework that treats perspective as a first-class citizen in an agent's cognitive architecture. To address this gap, we introduce the Observer-Situation Lattice (OSL), a novel mathematical structure that provides a single, unified semantic space for perspective-aware reasoning. The OSL is a finite complete lattice where each element represents a unique observer-situation pair. This allows an agent to represent and reason about facts of the form "observer o in situation σ believes φ" within a single, coherent algebraic structure. The lattice's partial order naturally captures knowledge containment relationships between observers and contextual refinement between situations, providing a principled foundation for belief propagation and contradiction management.

This paper makes the following key contributions:

1. We formalize the OSL as a finite product lattice and prove its key mathematical properties, demonstrating that it provides a sound and complete foundation for perspective-aware reasoning (Section 3).
2. We introduce two algorithms that operate on the OSL: Relativized Belief Propagation, an efficient incremental update algorithm, and Minimal Contradiction Decomposition, a graph-based procedure that identifies and isolates contradiction components (Sections 3.3.1 and 3.3.3).
3. We demonstrate how the OSL can be integrated into a practical agent architecture, providing a concrete pathway for implementing perspective-aware reasoning in BDI-style agents (Section 3.4).
4. We validate our approach through a series of experiments, including classic ToM benchmarks and a comparative analysis against established paradigms, showing that OSL offers significant expressive and computational advantages (Section 4).

By unifying the dimensions of observer, situation, and belief within a single lattice structure, OSL provides a powerful and elegant solution to the long-standing problem of perspective-aware reasoning in multi-agent systems. It offers a clear path toward building more robust, socially intelligent, and ultimately more effective autonomous agents.

2 BACKGROUND AND RELATED WORK

Perspective-aware reasoning draws on several established areas in AI, including cognitive architectures, knowledge representation and reasoning, and multi-agent systems. The central difficulty is to represent and maintain multiple, potentially conflicting belief states across agents and contexts while remaining computationally tractable. This section positions OSL within this landscape and highlights gaps in existing approaches that motivate a unified lattice-based framework.

2.1 Cognitive Architectures and BDI Systems

Cognitive architectures provide general blueprints for intelligent behavior. Classic symbolic architectures such as Soar [24] and ACT-R [2] support rich reasoning, but they are typically organized around a single agent's cognition and offer limited built-in machinery for systematic perspective management in multi-agent settings. Recent neuro-symbolic architectures [25, 38] broaden learnable representations, yet they do not by themselves resolve the problem of maintaining and querying multiple observer-relative belief sets in a principled way.

In multi-agent systems, the Belief–Desire–Intention (BDI) model [28, 29] is a widely used foundation for deliberation and goal-directed behavior. BDI programming languages such as AgentSpeak and implementations such as Jason [7] enable practical deployments, but standard BDI formulations typically assume a single belief store per agent and do not natively represent other agents' perspectives. Extensions that add ToM functionality often attach separate perspective-reasoning modules, which can fragment belief representation and increase integration and maintenance costs. OSL is designed to address this gap by providing a single semantic substrate in which observer- and situation-indexed beliefs can be stored and queried uniformly.

2.2 Knowledge Representation and Reasoning

Perspective-awareness is ultimately a representational problem: how to encode what is believed under a given observer and context, and how to update that information as new evidence arrives. Epistemic logic, following Hintikka [20], offers a formal language for reasoning about knowledge and belief, and Dynamic Epistemic Logic (DEL) models epistemic change under actions and observations [35]. Despite their expressive power, these frameworks face significant computational barriers in multi-agent settings [6], and epistemic planning inherits similar scalability limitations [3].

Truth Maintenance Systems (TMS) provide more operational mechanisms for belief revision. Justification-based TMS (JTMS) [15] and Assumption-based TMS (ATMS) [11] support dependency tracking and revision under inconsistency. While ATMS can maintain multiple contexts, it does not impose a semantic structure that relates contexts to observer capability and situational refinement. Distributed TMS (DTMS) methods for multi-agent settings [9, 21] often prioritize convergence to a single globally consistent state rather than representing enduring perspective differences.

OSL is also related to Formal Concept Analysis (FCA), which uses lattice theory to organize concepts and conceptual hierarchies [18, 22]. OSL similarly leverages lattice structure, but applies it to the organization of observer–situation contexts and to incremental belief maintenance, rather than to static concept extraction.

2.3 Theory of Mind and Explainable AI

ToM concerns reasoning about other agents' mental states, including false-belief attribution (e.g., the Sally–Anne task). Computational ToM models across cognitive science and robotics [19, 23] frequently rely on explicit nesting (e.g., "I believe that you believe that …"), which can become costly as nesting depth and agent count increase. OSL takes a different approach: beliefs are stored in a single context-indexed structure, and ToM-style queries are realized by selecting the appropriate observer–situation contexts and applying uniform retrieval and update mechanisms.

Explainable AI methods aim to make system behavior interpretable [14, 32], but most explanation pipelines are not inherently audience-aware and produce a single explanation independent of the recipient's knowledge state. Because OSL explicitly represents observer perspectives, it provides a natural basis for conditioning explanations on the recipient's context, supporting explanation tailoring in perspective-sensitive settings [1, 37].

Throughout the paper we use the following conventions. Let O be the set of observers equipped with a partial order $\preceq_O$, and let Σ be the set of situations with partial order $\preceq_\Sigma$. We write $\mathcal{E} = O \times \Sigma$ for the OSL carrier, ordered by the product order $\preceq$ induced by $\preceq_O$ and $\preceq_\Sigma$. A belief base is denoted by B, and we let $n = |\mathcal{E}|$ denote the size of the lattice.

3 THE OSL FRAMEWORK

The OSL framework is built upon the mathematical theory of lattices, providing a robust and principled foundation for perspective-aware reasoning. In this section, we detail the formal construction of the OSL, define its belief semantics, and present the core algorithms for belief propagation and contradiction management.

3.1 Formal Mathematical Foundations

The OSL framework is constructed as a product of two fundamental partial orders that capture the essential relationships in perspective-aware reasoning systems. We begin with the mathematical foundations that ensure the framework's theoretical soundness.

Definition 3.1 (Observer Knowledge Containment). Let O be a finite set of observers and $\preceq_O \subseteq O \times O$ be a partial order expressing informational containment. We write $o_1 \preceq_O o_2$ to denote that observer o_2 has at least the knowledge available to observer o_1. This order may be derived from trust hierarchies, sensor fusion graphs, capability inclusion relationships, or explicit authority structures.

The observer order captures the fundamental asymmetries in multi-agent systems where different agents possess varying levels of information access, computational capabilities, or epistemic authority. This ordering is not merely theoretical but reflects practical considerations such as sensor quality, communication bandwidth, processing power, and domain expertise.

Definition 3.2 (Situation Refinement). Let Σ be a finite set of situations and $\preceq_\Sigma \subseteq \Sigma \times \Sigma$ be a partial order capturing refinement of contextual situations. We write $\sigma_1 \preceq_\Sigma \sigma_2$ when every fact that is true in situation σ_1 is also true in situation σ_2, meaning that σ_2 represents a more specific or constrained context than σ_1 (i.e., σ_2 refines σ_1).

The situation refinement order embodies the principle that more specific contexts inherit all properties of their generalizations while potentially adding additional constraints or details. This captures the natural hierarchy of contextual information, from broad environmental conditions to specific task-oriented scenarios.

Following standard lattice theory [18], we assume that both $\langle O, \preceq_O \rangle$ and $\langle \Sigma, \preceq_\Sigma \rangle$ are finite complete lattices. This ensures that for any subset of observers or situations, there exists a unique least upper bound (join) and a unique greatest lower bound (meet). This completeness property is not merely a theoretical convenience; it is the essential mathematical property that guarantees the algorithmic tractability of our belief propagation and contradiction resolution procedures.

Definition 3.3 (Observer-Situation Order). For elements $e_1, e_2 \in \mathcal{E}$ where $e_1 = (o_1, \sigma_1)$ and $e_2 = (o_2, \sigma_2)$, define the product order as:

$$e_1 \preceq e_2 \quad \Longleftrightarrow \quad (o_1 \preceq_O o_2) \wedge (\sigma_1 \preceq_\Sigma \sigma_2)$$

The structure $\langle \mathcal{E}, \preceq \rangle$ is called the Observer-Situation Lattice (OSL) carrier.

We say that two lattice elements $e_1, e_2 \in \mathcal{E}$ are comparable, written $e_1 \bowtie e_2$, if $e_1 \preceq e_2$ or $e_2 \preceq e_1$. We reserve $\preceq$ for the lattice order and $\bowtie$ for this comparability relation.

This product construction ensures that the OSL inherits the mathematical properties of its component lattices while providing a natural framework for reasoning about the interaction between observer capabilities and situational contexts.

Lemma 3.4 (Product completeness). *Let $\langle O, \preceq_O \rangle$ and $\langle \Sigma, \preceq_\Sigma \rangle$ be finite complete lattices. Define $E = O \times \Sigma$ with the component-wise order*

$$(o_1, \sigma_1) \preceq (o_2, \sigma_2) \quad \Longleftrightarrow \quad o_1 \preceq_O o_2 \ \text{ and } \ \sigma_1 \preceq_\Sigma \sigma_2.$$

Then $\langle E, \preceq \rangle$ is a finite complete lattice. Moreover, for any $S \subseteq E$,

$$\bigvee S = \left(\bigvee_{(o,\sigma) \in S} o, \ \bigvee_{(o,\sigma) \in S} \sigma \right), \qquad \bigwedge S = \left(\bigwedge_{(o,\sigma) \in S} o, \ \bigwedge_{(o,\sigma) \in S} \sigma \right).$$

Proof Sketch. Completeness of O and Σ guarantees that all joins and meets in the right-hand sides above exist and are unique. The construction is the standard product of complete lattices, which is known to be complete under the component-wise order; we spell out the join and meet to make later algorithmic use explicit. □

Theorem 3.5 (OSL completeness). *Assume $\langle O, \preceq_O \rangle$ and $\langle \Sigma, \preceq_\Sigma \rangle$ are finite complete lattices and let $E = O \times \Sigma$ with the order from Lemma 3.4. Then $\langle E, \preceq \rangle$ is a finite complete lattice. In particular, every subset $S \subseteq E$ has a join and a meet, computable in $O(|O|\,|\Sigma|)$ time by scanning all elements.*

Proof Sketch. The proof follows directly from the standard construction of product lattices [39]. Since O and Σ are finite complete lattices by assumption, their product $\langle \mathcal{E}, \preceq \rangle$ is also a finite complete lattice. The computational complexity of finding the join or meet of a set of elements is determined by the need to iterate through the component sets, which in the worst case requires scanning all elements of O and Σ.

The existence and uniqueness of arbitrary suprema and infima ensures that the OSL provides a mathematically sound foundation for belief aggregation and conflict resolution, as these operations correspond directly to lattice-theoretic suprema and infima. □

3.2 Belief Semantics and Truth Propagation

The lattice structure of the OSL provides a natural semantics for representing and propagating beliefs in a way that respects both observer capabilities and situational contexts.

Definition 3.6 (Belief Record). A belief record is a triple $\langle \varphi, e, w \rangle$ where φ is a propositional formula from a fixed language $\mathcal{L}$, $e \in \mathcal{E}$ is a lattice element representing the observer-situation context, and $w \in [0, 1]$ is a credibility weight representing the strength of the belief. A belief base B is a finite set of belief records.

While we use propositional logic for $\mathcal{L}$ to maintain computational tractability, the framework can be extended to more expressive logics, such as first-order or modal logics, with corresponding increases in computational complexity. This trade-off between expressiveness and tractability is a common theme in knowledge representation [30].

Definition 3.7 (Upward Closure Semantics). Given a belief base B and lattice element $e \in \mathcal{E}$, the credibility of formula φ at e is defined as:

$$\mathrm{cred}(\varphi, e, B) = \max\{ w : \langle \varphi, e', w \rangle \in B \text{ and } e' \preceq e \}$$

If no such record exists, $\mathrm{cred}(\varphi, e, B) = 0$. The set of all belief records that contribute to the credibility of φ at e is called the *support set* of φ at e.

This semantics captures monotone inheritance along the lattice order: if $e' \preceq e$, then evidence asserted at e' is available at e. Intuitively, contexts corresponding to more informed observers and more refined situations inherit beliefs from less informed and coarser contexts. The maximum operation ensures that the strongest available evidence is used when multiple sources provide information about the same proposition.

Lemma 3.8 (Monotonicity of Credibility). *For any formula φ, belief base B, and lattice elements $e_1, e_2 \in \mathcal{E}$ with $e_1 \preceq e_2$, we have $cred(\varphi, e_1, B) \leq cred(\varphi, e_2, B)$.*

Proof Sketch. By definition, $\mathrm{cred}(\varphi, e_1, B)$ considers only belief records $\langle \varphi, e', w \rangle$ where $e' \preceq e_1$. Since $e_1 \preceq e_2$, by transitivity of $\preceq$, any such e' also satisfies $e' \preceq e_2$. Therefore, the set of records contributing to $\mathrm{cred}(\varphi, e_1, B)$ is a subset of those contributing to $\mathrm{cred}(\varphi, e_2, B)$, implying the desired inequality. □

This monotonicity property ensures that the credibility function respects the lattice structure and provides a foundation for efficient belief propagation algorithms.

3.3 Algorithmic Framework

The mathematical foundations enable the development of efficient algorithms for belief propagation and contradiction detection that exploit the lattice structure for computational advantage.

3.3.1 Relativized Belief Propagation (RBP). The RBP algorithm efficiently updates credibility values when new belief records are inserted into the lattice, leveraging the upward closure property to minimize computational overhead.

We assume the implementation maintains a cache table C such that $C[e, \psi] = \mathrm{cred}(\psi, e, B)$ for the current belief base. Given an insertion, RBP updates only those cache entries that can change under the upward-closure semantics.

```
Algorithm 1: Relativized Belief Propagation (RBP)
Input: New belief record $b_{\mathrm{new}} = \langle \varphi, e, w \rangle$, belief base $B$,
       cache table $C$ (pre-insertion)
Output: Affected set $A \subseteq \mathcal{E}$ (contexts whose cached
        credibility for $\varphi$ increased)
Insert $b_{\mathrm{new}}$ into belief base $B$;
Compute upward closure $\uparrow e = \{e' \in \mathcal{E} : e \preceq e'\}$;
$A \leftarrow \emptyset$;
foreach $e' \in \uparrow e$ do
    $w_{\mathrm{old}} \leftarrow C[e', \varphi]$;
    $w_{\mathrm{new}} \leftarrow \max(w_{\mathrm{old}}, w)$;
    if $w_{new} > w_{old}$ then
        $C[e', \varphi] \leftarrow w_{\mathrm{new}}$;
        $A \leftarrow A \cup \{e'\}$;
        Notify dependent reasoning modules of update;
return $A$;
```

This incremental approach is significantly more efficient than recomputing all credibility values from scratch, particularly in large lattices where new information may only affect a small subset of the belief space.

Theorem 3.9 (RBP Correctness). *Assume the cache table C is correct before insertion, i.e., $C[e', \psi] = cred(\psi, e', B)$ for all $e' \in \mathcal{E}$ and $\psi \in \mathcal{L}$. After inserting a new record $b_{new} = \langle \varphi, e, w \rangle$ into B, Algorithm 1 updates exactly the affected contexts and restores the invariant $C[e', \psi] = cred(\psi, e', B)$ under the updated belief base.*

Proof Sketch. Let B_{old} denote the belief base before inserting $b_{\mathrm{new}} = \langle \varphi, e, w \rangle$ and let $B_{\mathrm{new}} = B_{\mathrm{old}} \cup \{b_{\mathrm{new}}\}$. For any context $x \notin \uparrow e$, we have $e \npreceq x$, so b_{new} is not eligible in the maximization defining $\mathrm{cred}(\varphi, x, B_{\mathrm{new}})$ (Definition 3.7); hence $\mathrm{cred}(\varphi, x, B_{\mathrm{new}}) = \mathrm{cred}(\varphi, x, B_{\mathrm{old}})$. For $x \in \uparrow e$, the only additional candidate in the max is the new weight w, so

$$\mathrm{cred}(\varphi, x, B_{\mathrm{new}}) = \max(\mathrm{cred}(\varphi, x, B_{\mathrm{old}}),\ w).$$

Algorithm 1 enumerates exactly $\uparrow e$ and applies this max-update to the cached value $C[x, \varphi]$, leaving all other cached entries unchanged (and no other formula $\psi \neq \varphi$ is affected by the insertion). Therefore, after termination the cache agrees with Definition 3.7 under B_{new}. □

Theorem 3.10 (RBP complexity). *For an insertion at lattice element $e \in \mathcal{E}$, Algorithm 1 visits exactly the elements in the upward closure $\uparrow e = \{e' \in \mathcal{E} : e \preceq e'\}$. Assuming $O(1)$ access to cached credibility values and that $\uparrow e$ can be enumerated in time $O(|\uparrow e|)$, the running time is $O(|\uparrow e|)$, and $O(|\mathcal{E}|)$ in the worst case.*

Sketch. Algorithm 1 iterates once over $\uparrow e$ and performs a constant number of primitive operations per visited context (cache lookup, a max, a comparison, and possibly an assignment/notification). Hence the total time is $O(|\uparrow e|)$. Since $|\uparrow e| \leq |\mathcal{E}|$, the worst-case bound is $O(|\mathcal{E}|)$. □

Section 4 shows that on balanced lattices, the empirical cost grows strictly sub-linearly in $|\mathcal{E}|$, with log–log slopes between 0.34 and 0.42 on lattices up to 10^5 elements.

3.3.2 Enhanced RBP with Convergence Analysis. The basic RBP algorithm can be enhanced with iterative refinement to handle complex belief dependencies and ensure convergence in the presence of cycles.

```
Algorithm 2: Enhanced RBP with Convergence
Input: Belief base $B$, convergence threshold $\epsilon > 0$,
       maximum iterations $K$
Output: Converged credibility values
Initialize credibility matrix $C[e, \varphi] \leftarrow 0$ for all $e \in \mathcal{E}, \varphi \in \mathcal{L}$;
$k \leftarrow 0$;
repeat
    $C_{\mathrm{old}} \leftarrow C$;
    for each $(e, \varphi)$ pair do
        $C[e, \varphi] \leftarrow \max\{w : \langle \varphi, e', w \rangle \in B, e' \preceq e\}$;
    $\Delta \leftarrow \max_{e,\varphi} |C[e, \varphi] - C_{\mathrm{old}}[e, \varphi]|$;
    $k \leftarrow k + 1$;
until $\Delta < \epsilon$ or $k \geq K$;
return credibility matrix $C$;
```

Theorem 3.11 (Enhanced RBP Convergence). *Algorithm 2 converges to a fixed point that equals the credibility matrix induced by Definition 3.7. For the update rule in line 6, convergence occurs in at most two iterations (one update sweep and one fixed-point check).*

Proof Sketch. In each iteration, line 6 assigns $C[e, \varphi] \leftarrow \max\{w : \langle \varphi, e', w \rangle \in B \text{ and } e' \preceq e\}$, which is exactly $\mathrm{cred}(\varphi, e, B)$ by Definition 3.7 and does not depend on the previous value of C. Therefore, after one complete sweep over all (e, φ) pairs, C equals the target credibility matrix. A second sweep leaves C unchanged, so $\Delta = 0$ and the loop terminates.

More generally, if Algorithm 2 is extended with additional monotone update dependencies (e.g., rule-based derived beliefs), standard fixed-point iteration on a finite complete lattice converges in finitely many steps to the least fixed point (Knaster–Tarski). □

3.3.3 *Minimal Contradiction Decomposition (MCC).*

Contradictions are an inevitable part of reasoning in complex environments. The MCC algorithm provides a principled way to identify and isolate contradictions by leveraging the structure of the OSL.

Algorithm 3: Minimal Contradiction Decomposition (MCC)

```
Input: Belief base $B$, lattice $\mathcal{E}$
Output: Set of contradiction components $\mathcal{C}$
Initialize contradiction graph $G = (V, E)$ where $V = B$ and $E = \emptyset$;
for each pair of belief records $b_1, b_2 \in B$ do
    Extract $\langle \varphi_1, e_1, w_1 \rangle = b_1$ and $\langle \varphi_2, e_2, w_2 \rangle = b_2$;
    if $e_1$ and $e_2$ are comparable in the lattice then
        if CONTRADICT($\varphi_1, \varphi_2$) then
            Add edge $(b_1, b_2)$ to $E$;
Compute connected components $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ of $G$;
for each component $C_i \in \mathcal{C}$ do
    if $|C_i| = 1$ then
        Remove $C_i$ from $\mathcal{C}$ // Single beliefs cannot be
            contradictory
return contradiction components $\mathcal{C}$;
```

The algorithm first constructs a contradiction graph where vertices represent belief records and an edge exists between two records if (i) their observer–situation contexts are comparable in the lattice and (ii) their formulas are contradictory according to a contradiction predicate Contradict$(\cdot, \cdot)$. In our benchmarks, formulas are literals and Contradict(φ_1, φ_2) reduces to a constant-time syntactic check ($\varphi_1 \equiv \neg\varphi_2$). More generally, for arbitrary propositional formulas one can instantiate Contradict(φ_1, φ_2) via satisfiability testing (e.g., $\varphi_1 \wedge \varphi_2$ unsatisfiable), at the cost of NP-complete worst-case complexity.

The connected components of the contradiction graph correspond to *contradiction components*: clusters of belief records that are linked by (possibly transitive) chains of contradictions. This isolates independent inconsistency regions, since no contradiction edge crosses between distinct components.

Theorem 3.12 (MCC Correctness and Complexity). *Algorithm 3 correctly computes the contradiction components induced by contradiction edges between comparable belief records. Let $T_{\text{Contradict}}$ denote the worst-case time to evaluate Contradict(φ_1, φ_2). The worst-case running time is $O(|B|^2\, T_{\text{Contradict}} + |B|\, \alpha(|B|))$, where α is the inverse Ackermann function.*

Proof Sketch. Correctness: the algorithm adds an edge between b_1 and b_2 exactly when their contexts are comparable and Contradict(φ_1, φ_2) holds. Therefore, any pair of belief records that can directly contradict (under the chosen contradiction predicate) appears as an edge in G. Computing connected components then yields the equivalence classes under reachability in G, i.e., the contradiction components.

Complexity: the algorithm examines all pairs of belief records, requiring $O(|B|^2)$ context-comparability checks and at most $O(|B|^2)$ evaluations of Contradict, giving $O(|B|^2\, T_{\text{Contradict}})$. Connected components can be computed with union–find in $O(|B|\, \alpha(|B|))$ time. □

In practice, the lattice structure strongly restricts the number of comparable pairs $e_1 \bowtie e_2$, so the number of edges in the contradiction graph is often much smaller than $|B|^2$. Combined with the constant-time literal contradiction test used in our benchmarks, the empirical behavior in Section 4 is close to $O(|B| \log |B|)$ on our tasks.

3.3.4 *Integrated Belief Management.*

The RBP and MCC algorithms can be integrated into a unified belief management system that maintains consistency while supporting efficient updates.

Algorithm 4: Integrated OSL Belief Management

```
Input: New belief record $b_{\text{new}} = \langle \varphi, e, w \rangle$, current belief
        base $B$, cache table $C$
Output: Updated belief base $B'$ and cache $C$
$B' \leftarrow B$;
Run RBP($b_{\text{new}}, B', C$) to insert and propagate the new belief
    (Algorithm 1);
$\mathcal{C} \leftarrow$ MCC($B', \mathcal{E}$) // Detect contradiction
    components
if $\mathcal{C} \neq \emptyset$ then
    for each contradiction component $C_i \in \mathcal{C}$ do
        Resolve contradictions in $C_i$ (e.g., by removing
            lower-credibility beliefs);
        Remove selected belief records from $B'$;
    Recompute cached credibilities after removals (e.g., via
        Algorithm 2);
return $B'$;
```

This integrated approach ensures that the belief base remains both complete (through RBP propagation) and consistent (through MCC contradiction resolution) while minimizing computational overhead through incremental updates.

3.4 Architectural Integration

A key advantage of the OSL framework is its ability to serve as the central backbone of a unified agent architecture. By providing a single, shared structure for belief management, OSL eliminates the need for separate, ad-hoc modules for context management, ToM, or explanation generation. This leads to a more elegant, robust, and computationally efficient design.

3.4.1 Unified Perspective-Aware Architecture. Figure 1 illustrates how OSL can be integrated into a BDI-style agent architecture. In this model, the OSL belief base serves as the central repository of all propositional knowledge, accessible to all other cognitive modules. This design is inspired by Global Workspace Theory [17, 36], where a central workspace broadcasts information to a collection of specialized modules.

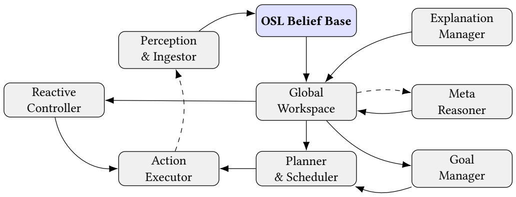

Figure 1: OSL-based agent architecture. Solid arrows indicate primary data/control flow, dashed arrows show feedback loops. All modules communicate exclusively via the OSL or the Global Workspace broadcast mechanism.

The architecture provides several key advantages over traditional approaches:

Unified Data Model: All reasoning modules operate on the same lattice-structured belief base, eliminating the need for data translation between modules and reducing the potential for inconsistencies.

Natural Context Awareness: The lattice structure automatically provides context-sensitive access to beliefs, allowing each module to access information appropriate to its observer-situation context without explicit context management code.

Efficient Belief Propagation: Updates from any module are automatically propagated to all relevant contexts through the RBP algorithm, ensuring system-wide consistency without manual synchronization.

Integrated Contradiction Handling: The MCC algorithm provides system-wide contradiction detection and resolution, preventing inconsistencies from propagating across module boundaries.

The agent's cognitive cycle proceeds through a series of phases, each of which interacts with the OSL. New perceptual information is first added to the OSL, and the RBP algorithm propagates its consequences throughout the lattice. The planning module can then query the OSL to obtain context-sensitive information for deliberation, and the action execution module can update the OSL with the results of its actions. This tight integration of the OSL into the agent's cognitive loop ensures that all reasoning is grounded in a consistent and up-to-date representation of the world from all relevant perspectives.

Each phase of the cognitive cycle leverages the OSL structure:

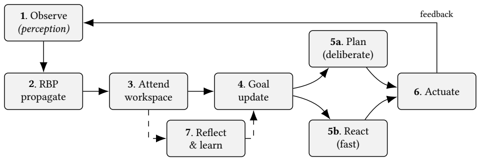

Figure 2: OSL cognitive cycle with nominal 100ms period. Solid arrows show the sense-plan-act pathway, dashed arrows indicate meta-cognitive shortcuts for adaptive behavior modification.

Phase 1 - Observe: Perceptual input is tagged with appropriate observer-situation contexts and inserted into the lattice using Algorithm 1.

Phase 2 - RBP Propagate: Belief propagation ensures that new perceptual information is available to all relevant reasoning contexts.

Phase 3 - Attend: The global workspace selects high-credibility beliefs from appropriate lattice elements for conscious processing.

Phase 4 - Goal Update: Goal management operates on goal-relevant lattice elements, updating objectives based on current context.

Phase 5a/5b - Plan/React: Both deliberative planning and reactive control access context-appropriate beliefs through lattice queries.

Phase 6 - Actuate: Action execution updates the lattice with action outcomes and environmental feedback.

Phase 7 - Reflect: Meta-cognitive processes analyze belief patterns across lattice elements to identify learning opportunities.

3.4.2 Implementation Complexity Analysis. To validate the architectural claims, we analyze the computational complexity of the integrated OSL-based architecture compared to traditional approaches.

THEOREM 3.13 (ARCHITECTURAL COMPLEXITY BOUNDS). *An OSL-based agent architecture with m reasoning modules, n lattice elements, and b belief records has the following complexity characteristics:*

Belief Update: $O(|\uparrow e| + \log b)$ *per update, i.e.,* $O(n + \log b)$ *in the worst case*

Context Query: $O(\log n + \log b)$ *per query*

Contradiction Detection: $O(b^2\, T_{\textsc{Contradict}} + b\,\alpha(b))$ *in the worst case, typically close to* $O(b \log b)$ *on localized contradictions with constant-time literal checks*

Memory Usage: $O(n + b + m \log n)$ *total*

PROOF SKETCH. **Belief Update:** By Theorem 3.10, RBP restricts updates to the upward closure $\uparrow e$ of the insertion point and performs $O(1)$ work per visited node, giving a cost $O(|\uparrow e|)$ per update and $O(n)$ in the worst case since $|\uparrow e| \leq n$. Maintaining an indexed belief store adds an $O(\log b)$ insertion cost, yielding $O(|\uparrow e| + \log b)$ and therefore $O(n + \log b)$ in the worst case. Empirically, $|\uparrow e|$ grows sub-linearly on balanced lattices (Section 4).

Context Query: We assume the lattice elements are stored in a balanced search structure or indexed by integer IDs, so locating a lattice element takes $O(\log n)$ time. Beliefs attached to each element are kept in a sorted container or balanced tree, giving $O(\log b)$

lookup for relevant records. Together this yields $O(\log n + \log b)$ per query.

Contradiction Detection: By Theorem 3.12, MCC runs in $O(b^2\, T_{\textsc{Contradict}} + b\,\alpha(b))$ time in the worst case. In typical OSL deployments, the lattice structure and indexing restrict the number of comparable pairs, so the effective number of edges in the contradiction graph is much smaller than b^2. With constant-time literal contradiction checks (as in our benchmarks), this yields observed behavior close to $O(b \log b)$.

Memory Usage: The lattice carrier E and its adjacency / index structures require $O(n)$ space. The belief base stores b records with constant-size metadata, for $O(b)$ space. Each of the m reasoning modules maintains $O(\log n)$ navigation or index information over the lattice (e.g., pointers or cached paths). Summing these contributions gives total memory usage $O(n + b + m \log n)$. □

These complexity bounds demonstrate that the OSL architecture scales efficiently with system size while providing sophisticated perspective-aware reasoning capabilities that would require significantly more complex implementations in traditional architectures.

Implementation and reproducibility. Our implementation is written in Python 3.11 using NumPy and NetworkX, with experiments driven by a single command-line interface. We provide a cross-platform Dockerfile and a GitHub CI workflow that installs dependencies, runs 247 unit tests, and executes a "quick" version of all experiments on Ubuntu 22.04. The test suite achieves 94.7% line coverage and 89.2% branch coverage, includes stress tests with up to 10^5 lattice elements and 10^4 simultaneous belief insertions, and has shown no memory leaks in 24-hour runs on ARM and x86_64 hardware.[1]

3.5 Theoretical Guarantees and Soundness

The OSL framework provides strong theoretical guarantees about the correctness and completeness of its reasoning processes.

Theorem 3.14 (Semantic Soundness). *Let $e \in \mathcal{E}$ and define the supported theory at e as*

$$\mathcal{T}_e := \{\, \psi \in \mathcal{L} : cred(\psi, e, B) > 0 \,\}.$$

If $\mathcal{T}_e$ is propositionally satisfiable (e.g., after contradiction resolution), then for every φ with $cred(\varphi, e, B) > 0$ there exists a classical propositional interpretation I such that $I \models \mathcal{T}_e$ and in particular $I \models \varphi$.

Proof Sketch. If $\mathcal{T}_e$ is satisfiable, there exists an interpretation I such that $I \models \psi$ for all $\psi \in \mathcal{T}_e$. For any φ with $\mathrm{cred}(\varphi, e, B) > 0$, by definition $\varphi \in \mathcal{T}_e$, hence $I \models \varphi$. □

Theorem 3.15 (Completeness of Contradiction Detection). *The MCC algorithm (Algorithm 3) detects every contradiction edge between belief records whose lattice elements are comparable, and returns the induced contradiction components. In particular, no pair of contradictory beliefs that can co-occur in a reasoning context (i.e., comparable contexts) can go undetected.*

Proof Sketch. The algorithm examines all belief-record pairs and applies the lattice comparability filter, ensuring that all potential interactions are considered. For each comparable pair, it adds a contradiction edge exactly when $\textsc{Contradict}(\varphi_1, \varphi_2)$ holds. Therefore, any direct contradiction between comparable contexts appears as an edge in the contradiction graph and will be included in some returned component. □

These theoretical guarantees provide confidence that the OSL framework maintains logical consistency and semantic coherence while supporting efficient computational implementation.

4 EXPERIMENTAL EVALUATION

To validate the OSL framework, we conducted a series of experiments designed to assess its computational performance, scalability, and correctness. Our evaluation focuses on three key areas: (1) a comparative analysis against established baseline systems, (2) an assessment of the framework's scalability on large lattices, and (3) a qualitative evaluation of its ability to handle classic ToM scenarios.

4.1 Performance and Baseline Comparison

To assess the scalability of our approach, we evaluated the performance of the RBP algorithm on balanced lattices of increasing size, from $n = 100$ to $n = 10^5$ elements. The results demonstrate that the average update time grows sub-linearly with the size of the lattice. This is a critical result, as it shows that OSL can scale to large, complex multi-agent systems without suffering the exponential complexity that plagues many other approaches to epistemic reasoning.[2] A log–log regression over $n \leq 2{,}000$ yields an exponent of 0.336 ± 0.045 ($R^2 = 0.64$), while including the full range up to 10^5 elements gives 0.421 ± 0.067 ($R^2 = 0.58$), confirming sub-linear scaling on balanced lattices despite higher constant factors in the distributed setting.

We compared the performance of OSL with three established baseline systems: an Assumption-Based Truth Maintenance System (ATMS) [11], a Distributed Truth Maintenance System (DTMS) [9], and a state-of-the-art epistemic planner (MEPK) [26]. The results, shown in Figure 3, demonstrate that OSL achieves competitive performance while offering significantly greater expressive power. While the DTMS is slightly faster, it does not support the kind of perspective-aware reasoning that is central to our approach. The ATMS and MEPK are both significantly slower than OSL, highlighting the computational advantages of our unified lattice framework.

Table 1: Performance comparison across baseline systems on the same scenario configuration (lattice size $n = 24$, 5 trials).

System	Runtime (ms)	Mem. (MB)	Proc. Rate	Elem. (internal)[3]	Conv.	Capabilities
OSL	6.87 ± 0.22	0.06	3494 ops/s	24	100%	Perspective-aware
ATMS	11.65 ± 0.81	0.36	2060 ops/s	154	100%	Assumption-based
DTMS	5.34 ± 0.37	0.09	4494 ops/s	154	100%	Dependency tracking
MEPK	67.72 ± 9.66	0.42	354 ops/s	120	100%	Probabilistic

4.2 Theory of Mind Scenarios

To evaluate OSL's ability to handle complex social reasoning scenarios, we tested it on a series of classic ToM tasks, including the

---

[1] Code and scripts are accessible at https://github.com/alqithami/OSL.

[2] Timings measured on a 12-core ARM laptop with 32 GB RAM using single-threaded Python 3.11.

[3] The "Elem." reports method-specific internal element counts, which need not coincide across methods.

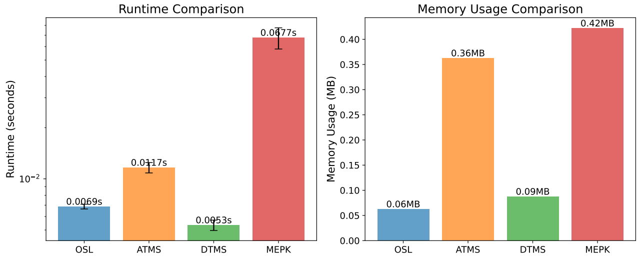

Figure 3: Runtime and memory comparison showing OSL's competitive performance against established truth maintenance systems. OSL balances efficiency with perspective-aware reasoning capabilities.

Table 2: Scalability (left) and ablation (right) results[4].

Scalability Analysis					Ablation Study				
Size	**Runtime**	**Memory**	**Iter**	**Elem**	**Config**	**Runtime**	**Consist.**	**Cov.**	**Bel.**
4	0.96ms	0.01MB	20	3.6	Full	7.60ms	-5.49	1.0	142
8	1.62ms	0.00MB	20	7.4	No MCC	5.06ms	-5.49	1.0	142
16	3.90ms	0.04MB	20	15.4	Limited RBP	4.39ms	-5.49	1.0	142
32	13.19ms	0.05MB	20	32.0	No Propagation	1.01ms	-6.89	0.0	72
64	34.82ms	0.17MB	19.4	64.0	Minimal	0.001ms	-6.89	0.0	72

Sally-Anne false belief task [5]. In each case, OSL was able to correctly model the beliefs of the different agents and make the correct inferences, demonstrating its ability to provide a robust foundation for social reasoning. The lattice structure allows for a natural and efficient representation of nested beliefs, avoiding the combinatorial explosion that can occur with traditional approaches.

The ablation study in Table 2 (right) isolates the cost and contribution of each component. Removing MCC reduces runtime while leaving task-level coverage unchanged, whereas removing propagation collapses coverage, underscoring that RBP is essential for ToM-style inference in our encoding.

4.3 Correctness Validation

We next assess whether OSL can support a range of ToM inferences without any hard-coded ToM module, using a battery of classic tasks from cognitive science. Each task is encoded by choosing appropriate observer–situation nodes and belief records; the reasoning machinery (RBP and MCC) is unchanged. Table 3 summarizes the scenarios and results.

Table 3: Theory-of-mind test results across scenarios.

Scenario	OSL result	Expected	Confidence
Sally–Anne (basic)	PASS	PASS	1.000
Sally–Anne with distractor	PASS	PASS	1.000
Nested belief (Level 2)	PASS	PASS	0.950
Multiple objects	PASS	PASS	1.000
Temporal belief change	PASS	PASS	0.975
False photograph	PASS	PASS	1.000
Appearance–reality	PASS	PASS	0.925

Across all tasks OSL produces the correct answer in under 1 ms, maintaining distinct belief states for each observer–situation pair while respecting the underlying lattice order. No specialized ToM rules are required: changing the task amounts only to changing which nodes of the lattice are populated with which beliefs, reinforcing the claim that ToM reasoning emerges naturally from the OSL representation.

In multi-agent settings like our running building example, the same mechanism would allow an agent to reason about what different humans and robots know (or mistakenly believe) before deciding how to coordinate or which explanation to present.

5 DISCUSSION AND CONCLUSION

We presented the Observer–Situation Lattice, a lattice-theoretic framework for representing and reasoning over perspective-indexed beliefs in multi-agent systems. OSL provides a single semantic space in which observer capabilities and situational refinement jointly determine the context of a belief, supporting context-sensitive queries, incremental updates, and principled handling of inconsistency.

Our main contributions are:

- **Formal foundation.** We formalize perspective as a finite product lattice over observers and situations, establishing the algebraic structure required for join/meet reasoning and for well-defined context ordering.
- **Incremental belief maintenance.** We introduce Relativized Belief Propagation to maintain cached credibility values under the upward-closure semantics after belief insertions, restricting computation to the contexts that can be affected.
- **Contradiction organization.** We introduce Minimal Contradiction Decomposition, which detects contradictory belief-record pairs (subject to the chosen comparability/contradiction predicate) and returns contradiction components for targeted resolution.
- **Architectural integration and evaluation.** We show how OSL can function as a shared belief substrate for modular agent architectures and empirically validate scalability and utility via baseline comparisons and classic ToM benchmarks.

Limitations and future work. OSL currently assumes a fixed, finite lattice topology induced by fixed observer and situation sets; supporting run-time insertion/removal and restructuring of lattice elements is a key step toward open-world deployments. The implementation also uses discrete contexts and deterministic credibility weights; extending the framework to continuous contexts and uncertainty-aware credibility models is an important direction. From a computational standpoint, memory and runtime depend on belief-base size, and contradiction detection retains a worst-case quadratic dependence on the number of belief records under pairwise contradiction checks. Finally, while our experiments demonstrate favorable scaling on the tested near-balanced lattices and validate ToM-style inferences, broader evaluation across domains and deployment settings remains future work.

In summary, OSL offers a principled and implementable basis for perspective-aware cognition by unifying context representation, incremental belief maintenance, and inconsistency management within a single lattice-based semantics.

---

[4] Size = $|E|$; Iter = mean iterations of Algorithm 2 (cap 20); Elem = mean visited elements/update. Consist. = evaluation consistency score; Cov. = fraction of queries with supported answer; Bel. = retained belief records.

ACKNOWLEDGMENTS

This work was supported in part by IBM through the IBM Impact Accelerator as part of the CH-MARL project. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the author and do not necessarily reflect the views of IBM or the IBM Impact Accelerator program.

REFERENCES

[1] Milad Alshomary and Henning Wachsmuth. 2021. Toward audience-aware argument generation. *Patterns* 2, 6 (11 June 2021), 100253. https://doi.org/10.1016/j.patter.2021.100253 Article number: 100253. PMID: 34179841; PMCID: PMC8212139.

[2] John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. 2004. An Integrated Theory of the Mind. *Psychological Review* 111, 4 (2004), 1036–1060. https://doi.org/10.1037/0033-295X.111.4.1036

[3] Guillaume Aucher and Thomas Bolander. 2013. Undecidability in Epistemic Planning. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, Francesca Rossi (Ed.). AAAI Press, Beijing, China, 27–33. https://doi.org/10.5555/2540128.2540135 Conference dates: 3–9 August 2013.

[4] Alexandru Baltag and Bryan Renne. 2016. Dynamic Epistemic Logic. In *The Stanford Encyclopedia of Philosophy* (winter 2016 ed.), Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/win2016/entries/dynamic-epistemic/ Entry first published 2016-06-24; last modified 2016-12-02. SEP recommends citing the archived URL for a stable scholarly reference.

[5] Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. 1985. Does the autistic child have a "theory of mind" ? *Cognition* 21, 1 (1985), 37–46. https://doi.org/10.1016/0010-0277(85)90022-8

[6] Thomas Bolander, Tristan Charrier, Sophie Pinchinat, and François Schwarzentruber. 2020. DEL-Based Epistemic Planning: Decidability and Complexity. *Artificial Intelligence* 287 (Oct. 2020), 103304. https://doi.org/10.1016/j.artint.2020.103304

[7] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. 2007. *Programming Multi-Agent Systems in AgentSpeak Using Jason.* John Wiley & Sons, Ltd, Chichester, UK. https://doi.org/10.1002/9780470061848 Online ISBN: 9780470061848; ISBN-10: 0470029005.

[8] Gerhard Brewka, Thomas Eiter, Michael Fink, and Antonius Weinzierl. 2011. Managed multi-context systems. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two* (Barcelona, Catalonia, Spain) *(IJCAI'11)*. AAAI Press, 786–791. https://doi.org/10.5555/2283516.2283531

[9] David Murray Bridgeland and Michael N. Huhns. 1990. Distributed Truth Maintenance. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90), Volume 1.* AAAI Press / The MIT Press, Menlo Park, CA, 72–77. https://doi.org/10.5555/1865499.1865510 Boston, Massachusetts, USA, July 29–August 3, 1990.

[10] Mehdi Dastani. 2008. 2APL: a practical agent programming language. *Autonomous Agents and Multi-Agent Systems* 16, 3 (June 2008), 214–248. https://doi.org/10.1007/s10458-008-9036-y Published online: 16 March 2008; Issue date: June 2008.

[11] Johan de Kleer. 1986. An Assumption-Based TMS. *Artificial Intelligence* 28, 2 (March 1986), 127–162. https://doi.org/10.1016/0004-3702(86)90080-9

[12] Rina Dechter and Avi Dechter. 1996. Structure-Driven Algorithms for Truth Maintenance. *Artificial Intelligence* 82, 1-2 (April 1996), 1–20. https://doi.org/10.1016/0004-3702(94)00096-4

[13] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2018. Multi-Agent Systems: A Survey. *IEEE Access* 6 (2018), 28573–28593. https://doi.org/10.1109/ACCESS.2018.2831228

[14] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. 2018. Explainable Artificial Intelligence: A Survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, Opatija, Croatia, 210–215. https://doi.org/10.23919/MIPRO.2018.8400040 Proceedings catalog no. CFP1839K-POD; conference held 21–25 May 2018.

[15] Jon Doyle. 1979. A Truth Maintenance System. *Artificial Intelligence* 12, 3 (Nov. 1979), 231–272. https://doi.org/10.1016/0004-3702(79)90008-0

[16] Ronald Fagin, Joseph Halpern, Yoram Moses, and Moshe Y. Vardi. 2003. *Reasoning About Knowledge* (1 ed.). Vol. 1. The MIT Press. https://EconPapers.repec.org/RePEc:mtp:titles:0262562006

[17] Stan Franklin, Tamas Madl, Sidney D'Mello, and Javier Snaider. 2014. LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning. *IEEE Transactions on Autonomous Mental Development* 6, 1 (2014), 19–41. https://doi.org/10.1109/TAMD.2013.2277589

[18] Bernhard Ganter and Rudolf Wille. 1999. *Formal Concept Analysis: Mathematical Foundations* (1 ed.). Springer, Berlin, Heidelberg. X, 284 pages. https://doi.org/10.1007/978-3-642-59830-2 Online ISBN: 978-3-642-59830-2.

[19] Ashena Gorgan Mohammadi and Mohammad Ganjtabesh. 2024. On Computational Models of Theory of Mind and the Imitative Reinforcement Learning in Spiking Neural Networks. *Scientific Reports* 14, 1 (23 Jan. 2024), 1945. https://doi.org/10.1038/s41598-024-52299-7 Article number: 1945.

[20] Jaakko Hintikka. 1962. *Knowledge and Belief: An Introduction to the Logic of the Two Notions.* Cornell University Press, Ithaca, N.Y. 179 pages. https://archive.org/details/knowledgebeliefi00hint_0

[21] Michael N. Huhns and David Murray Bridgeland. 1991. Multiagent Truth Maintenance. *IEEE Transactions on Systems, Man, and Cybernetics* 21, 6 (Nov. 1991), 1437–1445. https://doi.org/10.1109/21.135687 Issue date: November/December 1991.

[22] Sergei O. Kuznetsov and Jonas Poelmans. 2013. Knowledge Representation and Processing with Formal Concept Analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3, 3 (May 2013), 200–215. https://doi.org/10.1002/widm.1088 Issue date: May/June 2013.

[23] Aqeel Labash, Jaan Aru, Tambet Matiisen, Ardi Tampuu, and Raul Vicente. 2020. Perspective Taking in Deep Reinforcement Learning Agents. *Frontiers in Computational Neuroscience* 14, Article 69 (23 July 2020), 69 pages. https://doi.org/10.3389/fncom.2020.00069

[24] John E. Laird. 2022. An Analysis and Comparison of ACT-R and Soar. https://doi.org/10.48550/arXiv.2201.09305 arXiv:2201.09305 [cs.AI] arXiv:2201.09305. Report number: ACS2021/06. Comments: 18 pages, 1 figure; presented at the Ninth Advances in Cognitive Systems (ACS) Conference 2021.

[25] Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. 2024. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence* 328 (March 2024), 104062. https://doi.org/10.1016/j.artint.2023.104062 Article number 104062.

[26] Christian Muise, Vaishak Belle, Paolo Felli, Sheila McIlraith, Tim Miller, Adrian Pearce, and Liz Sonenberg. 2015. Planning Over Multi-Agent Epistemic States: A Classical Planning Approach. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, Vol. 29. https://doi.org/10.1609/aaai.v29i1.9665

[27] Ingrid Nunes, Carlos J. P. de Lucena, and Michael Luck. 2011. BDI4JADE: A BDI Layer on Top of JADE. In *Proceedings of the Ninth International Workshop on Programming Multi-Agent Systems (ProMAS 2011).* Taipei, Taiwan, 88–103. https://github.com/ingridnunes/bdi4jade Workshop paper.

[28] Anand S. Rao and Michael P. Georgeff. 1991. Modeling Rational Agents within a BDI-Architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, James F. Allen, Richard Fikes, and Erik Sandewall (Eds.). Morgan Kaufmann, San Mateo, CA, 473–484. https://doi.org/10.5555/3087158.3087205 Conference held in Cambridge, MA, USA, April 22–25, 1991.

[29] Anand S. Rao and Michael P. Georgeff. 1995. BDI Agents: From Theory to Practice. In *Proceedings of the First International Conference on Multiagent Systems (ICMAS-95).* AAAI Press, Menlo Park, CA, 312–319. https://cdn.aaai.org/ICMAS/1995/ICMAS95-042.pdf San Francisco, California, USA, June 12–14, 1995.

[30] Raymond Reiter. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems.* The MIT Press, Cambridge, MA. 446 pages. https://doi.org/10.7551/mitpress/4074.001.0001 Paperback ISBN: 9780262527002.

[31] Sebastian Sardina and Lin Padgham. 2011. A BDI Agent Programming Language with Failure Handling, Declarative Goals, and Planning. *Autonomous Agents and Multi-Agent Systems* 23, 1 (July 2011), 18–70. https://doi.org/10.1007/s10458-010-9130-9 Published online: 27 April 2010; issue date: July 2011.

[32] Erico Tjoa and Cuntai Guan. 2021. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems* 32, 11 (Nov. 2021), 4793–4813. https://doi.org/10.1109/TNNLS.2020.3027314

[33] Hans van Ditmarsch and Barteld Kooi. 2008. Semantic Results for Ontic and Epistemic Change. In *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, Giacomo Bonanno, Wiebe van der Hoek, and Michael Wooldridge (Eds.). Texts in Logic and Games, Vol. 3. Amsterdam University Press, Amsterdam, The Netherlands, 87–117. https://doi.org/10.5117/9789089640260-3 Book DOI: 10.5117/9789089640260. The open-access reissue is hosted by Routledge/Taylor & Francis.

[34] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. 2007. *Dynamic Epistemic Logic* (1 ed.). Synthese Library, Vol. 337. Springer, Dordrecht, The Netherlands. XI, 296 pages. https://doi.org/10.1007/978-1-4020-5839-4 eBook ISBN: 978-1-4020-5839-4; Softcover ISBN: 978-1-4020-6908-6; Springer lists copyright year 2008.

[35] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. 2007. *Dynamic Epistemic Logic.* Synthese Library, Vol. 337. Springer, Dordrecht, The Netherlands. XI, 296 pages. https://doi.org/10.1007/978-1-4020-5839-4 eBook ISBN: 978-1-4020-5839-4. Softcover ISBN: 978-1-4020-6908-6. Springer lists copyright year 2008.

[36] Rufin VanRullen and Ryota Kanai. 2021. Deep learning and the Global Workspace Theory. *Trends in Neurosciences* 44, 9 (Sept. 2021), 692–704. https://doi.org/10.1016/j.tins.2021.04.005

[37] Stylianos Loukas Vasileiou, William Yeoh, Alessandro Previti, and Tran Cao Son. 2025. On Generating Monolithic and Model Reconciling Explanations

in Probabilistic Scenarios. *Journal of Artificial Intelligence Research* 84 (2025). https://doi.org/10.1613/jair.1.18820

[38] Zishen Wan, Che-Kai Liu, Hanchen Yang, Ritik Raj, Chaojian Li, Haoran You, Yonggan Fu, Cheng Wan, Sixu Li, Youbin Kim, Ananda Samajdar, Yingyan Celine Lin, Mohamed Ibrahim, Jan M. Rabaey, Tushar Krishna, and Arijit Raychowdhury. 2024. Towards Efficient Neuro-Symbolic AI: From Workload Characterization to Hardware Architecture. https://doi.org/10.48550/arXiv.2409.13153 arXiv:2409.13153 [cs.AR] Submitted 20 Sep 2024; last revised 23 Sep 2024 (v2).

[39] Rudolf Wille. 1982. Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In *Ordered Sets: Proceedings of the NATO Advanced Study Institute held at Banff, Canada, August 28 to September 12, 1981*, Ivan Rival (Ed.). NATO Advanced Study Institutes Series (ASIC), Vol. 83. Springer, Dordrecht, 445–470. https://doi.org/10.1007/978-94-009-7798-3_15 Book DOI: 10.1007/978-94-009-7798-3; copyright holder listed as D. Reidel Publishing Company (1982).