VSAR: A Unified Encoding & Reasoning Framework (Clean Edition)

This document presents a **clean, self-contained specification** of VSAR: a maximally flexible reasoning substrate built on **Vector Symbolic Architectures (VSA)** using **Fourier Holographic Reduced Representations (FHRR)**, implemented in **VSAX**.

VSAR is not a single logic. It is a **logic substrate**: a common representational layer capable of supporting multiple reasoning paradigms via shared encodings and different inference control policies.

Supported reasoning modes include:

- classical & deductive reasoning
- Description Logics (DL)
- paraconsistent logic
- non-monotonic & default reasoning
- abductive & inductive reasoning
- analogical & case-based reasoning
- probabilistic / weighted reasoning
- argumentative (graph-based) reasoning
- epistemic (multi-agent knowledge & belief) reasoning

---

1. Design principles

1.1 Representation vs inference control

VSAR enforces a strict separation between:

- **Representation**: what symbols, terms, rules, beliefs *are* (this document)
- **Inference control**: how they are used (search, proof, defeat, revision)

The encoding layer is deliberately **logic-neutral**, **non-explosive**, and **compositional**.

1.2 Why FHRR

FHRR provides:

- invertible binding via complex conjugation
- near-orthogonal random bases
- FFT-based efficiency
- predictable noise accumulation

These properties allow **crisp symbolic decoding** via *unbind* → *cleanup*, even in the presence of noise.

2. Mathematical foundations

Let all vectors live in $\mathbb{C}^d$ with unit magnitude components.

2.1 Core operations

- **Binding**
  $x \otimes y$ (phase addition / FFT-domain multiplication)

- **Unbinding**
  $x \oslash y = x \otimes \overline{y}$

- **Bundling**
  $x \oplus y = \mathrm{normalize}(x + y)$

Binding is invertible in expectation; bundling introduces controlled noise.

3. Cleanup and similarity (critical distinction)

3.1 Unbinding vs cleanup

Unbinding is a **purely algebraic inverse**. Given a bound vector, unbinding isolates a signal but *does not discretize it*:

$$v = s + \epsilon$$

where s is the intended symbol and ϵ is cross-talk noise.

Unbind never selects a symbol.

3.2 Cleanup (symbol commitment)

Cleanup is the **symbol recovery and commitment operation**. Given a noisy vector v, cleanup selects the nearest basis vector from a **typed codebook**:

$$\mathrm{cleanup}_S(v) = \arg\max_{s \in S} \cos(v, s)$$

Cleanup converts continuous representations into discrete symbols.

3.3 Why cleanup works

- random FHRR vectors are nearly orthogonal
- noise distributes uniformly in high dimensions
- the correct symbol remains uniquely aligned

As dimensionality increases, cleanup error probability decreases exponentially.

3.4 Typed cleanup

Cleanup must be **typed**. Candidate sets are restricted by semantic role:

- individuals/constants
- concepts
- roles
- literals
- terms

Typed cleanup functions as VSAR's **type system**.

3.5 Cleanup vs similarity search

- **Cleanup** commits to a single intended symbol
- **Similarity search** explores related items

Similarity is used for analogy, induction, and case retrieval. Cleanup is used for deduction and decoding.

Rule: similarity explores; cleanup commits.

---

4. Symbol spaces

Separate VSAMemory codebooks are maintained for:

- individuals / constants
- concepts (unary predicates)
- roles (binary relations)
- function symbols
- argument roles (ARG_1, ARG_2, …)
- structural roles (HEAD, BODY, SRC, TGT, LEFT, RIGHT)
- logical tags (ATOM, TERM, RULE, LIT, META, AXIOM)
- logical operators (AND, OR, NOT, EXISTS, FORALL)
- epistemic operators (KNOW, BELIEF)
- graph operators (EDGE, SUPPORT, ATTACK)

---

5. Encoding basic logical structure

5.1 Constants

$$\mathrm{enc}(c) = E_c$$

5.2 Function terms

$$\mathrm{enc}(f(t_1,\ldots,t_k)) = (F_f \otimes TAG_{TERM}) \otimes \bigoplus_{i=1}^{k}(ARG_i \otimes \mathrm{enc}(t_i))$$

5.3 Atoms

$$\mathrm{enc}(p(t_1,\ldots,t_k)) = (P_p \otimes TAG_{ATOM}) \otimes \bigoplus_{i=1}^{k}(ARG_i \otimes \mathrm{enc}(t_i))$$

Decoding proceeds via unbinding followed by typed cleanup.

---

6. Description Logic (DL) encodings

6.1 Atomic concepts

$$\mathrm{enc}(C) = C_C$$

6.2 Concept constructors

- Conjunction:
  $enc(C \sqcap D) = (AND \otimes TAG_{CONCEPT}) \otimes (LEFT \otimes enc(C) \oplus RIGHT \otimes enc(D))$
- Disjunction: analogous with OR
- Negation:
  $enc(\neg C) = (NOT \otimes TAG_{CONCEPT}) \otimes enc(C)$
- Existential restriction:
  $enc(\exists R.C) = (EXISTS \otimes TAG_{CONCEPT}) \otimes (ROLE \otimes enc(R) \oplus FILLER \otimes enc(C))$
- Universal restriction: analogous with FORALL

6.3 DL axioms

- Subsumption:
  $enc(C \sqsubseteq D) = (SUBCLASS \otimes TAG_{AXIOM}) \otimes (LHS \otimes enc(C) \oplus RHS \otimes enc(D))$
- Assertions:
  $enc(C(a)) = (ASSERTION \otimes TAG_{ABOX}) \otimes (CONCEPT \otimes enc(C) \oplus IND \otimes enc(a))$

---

7. Negation, paraconsistency, and non-monotonicity

7.1 Classical negation

$$enc(\neg A) = (NEG \otimes TAG_{LIT}) \otimes enc(A)$$

7.2 Default negation (NAF)

$$enc(not\ A) = (NAF \otimes TAG_{META}) \otimes enc(A)$$

7.3 Paraconsistent belief state

Each literal maintains independent support:

$$\langle supp(A), supp(\neg A) \rangle$$

This yields four-valued semantics without explosion.

---

8. Rules, defaults, and abduction

8.1 Rules

$$enc(rule) = TAG_{RULE} \otimes (HEAD \otimes enc(H) \oplus BODY \otimes enc(B_1 \wedge \ldots \wedge B_n))$$

8.2 Defaults

Defaults are rules with priorities and optional exceptions.

8.3 Abduction

Abduction proceeds by inverse use of rules:

- unify goals with rule heads
- hypothesize missing premises
- score explanations by weight, simplicity, and consistency

---

9. Argumentative reasoning

Claims are nodes; support and attack relations are edges.

$$enc(support(A, B)) = (EDGE \otimes SUPPORT) \otimes (SRC \otimes enc(A) \oplus TGT \otimes enc(B))$$

Acceptability is computed by a graph semantics layer.

---

10. Epistemic logic

$$enc(K_a\varphi) = (KNOW \otimes TAG_{EPI}) \otimes (AGENT \otimes enc(a) \oplus CONTENT \otimes enc(\varphi))$$

Belief states are agent-indexed and may be inconsistent.

---

11. Analogical and case-based reasoning

Cases and mappings are structured objects retrievable by similarity and adapted via structural alignment.

---

12. Why this framework works

- high-dimensional orthogonality ensures slot separability
- invertible binding supports precise decoding
- cleanup restores discreteness
- similarity supports robustness and generalization
- inference control enforces logic-specific semantics

---

13. Core tests

- bind/unbind identity
- cleanup correctness under noise
- typed decoding invariants
- DL constructor recovery
- paraconsistent coexistence
- epistemic nesting
- argument graph propagation

---

14. Final perspective

VSAR is a **unified symbolic substrate**.

All reasoning modes arise from:

> shared encodings + different inference policies

This separation is what allows VSAR to combine logical rigor, robustness, and interpretability in a single system.