* FCA / Semiotics: SPO. Object / Concept Context / Attribute. Context: Sign / Concept / Object. Statement networks.
* Syntax: Transform (Rule / Production).
* (Context, Occurrence, Sign, Concept);
* (Context, Occurrence, Concept, Sign);
* Grammar: Rules (RHS).
* (Context, Concept, Occurrence, Sign);
* (Context, Concept, Sign, Occurrence);
* Semantics / Pragmatics: Productions (LHS).
* (Context, Sign, Concept, Occurrence);
* (Context, Sign, Occurrence, Concept);
* Occurrences: Objects.
* Signs: Terminals.
* Concepts: Non Terminals.
* Verbs: action (rel end: amante) / passion (rel end: amado) / state (rel: amor). Relation parts attributes.
* FCA:
* Object: Subject.
* Context: Predicate.
* Attribute: Object.
* DOM: (Class, Instance, Attribute, Value);
* Recursively nested Statements (CSPOs, RDF\*) as Tensors. Algebraic IDs / Contexts operations. Functional traversal / resolution / state (nav ctx views / transforms).
* Augmentations (Sets: Semiotics / DCI / DOM Statements Source):
* Schema Aggregation: Type / Relationships (Kinds / Roles) Inference. Clustering: Unsupervised Features Learning.
* Data Alignment: Type (Feature) Attributes Value Inference. Classification: (gender, salary range: scaling).
* Behavior Activation: Available Transforms (State Browsing) Inference. Regression: State (class attributes values in scenario: relationship flow).
* REST Browsing State Based Dialog Wizard. DCI / CDI / Augmentations (FCA). Structured Prompts / Responses (Statements Flow, Relationships / Roles).
* URNs: Semantic Identifiers. Encoding. DIDs (Distributed IDs).
* Conversational State Transfer (COST): Distributed (P2P).
* Node.js: JSON-LD. Functional Resources (Monads, run-at: request client / server peer. Context State available Functors / Transforms).
* Parsing: Transforms.