Chirality Framework 4.5.1

DO NOT USE CODE IN THE EXECUTION OF THESE COMMANDS, except to display final results where convenient. These operations are all meant to be carried out directly in the chat by the LLM's main function as a language model, not a code interpreter.

Starting Axioms

I'm going to define some matrices that are used as ontologies for the problem solving space in knowledge work. If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation.

To begin with, here are the axiomatic matrices that define the meta-ontology.

Matrix A
Size: 3x4
Column names: [‘Guiding’, ‘Applying, ‘Judging’, ‘Reflecting’]
Row names: [‘Normative Level’, ‘Operational Level’, ‘Evaluative Level’]
Elements: [[‘Values‘, ‘Actions’, ‘Benchmarks’, ‘Feedback’]
[‘Principles’, ‘Methods’, ‘Standards’, ‘Adaptation’]
[‘Goals’, ‘Coordination’, ‘Evaluation’, ‘Consolidation’]]

Matrix B
Size: 4x4
Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Feasibility’]
Row names: [‘Data’, ‘Information’, Knowledge’, ‘Wisdom’]
Elements: [[‘Necessary, ‘Sufficient‘, ‘Complete’, ‘Probability’]
[‘Contingent’, ‘Insufficient’ ‘Incomplete, ‘Possibility’]
[‘Fundamental’, ‘Appropriate’, ‘Holistic’, ‘Feasibility’]
[‘Best Practices‘, ‘Limits of’, ‘Justification for’, ‘Practicality’]]

****

The Chirality Framework
What is the Chirality Framework?

It is a set of matrices that act as ontologies to map the knowledge base for the production of engineering knowledge.

Ontology
The semantic matrices generated in a chat are ontologies that define the categories and concepts in which to work as we generate reliable knowledge.
The column and row names of the matrices are the axioms that form the first layer of the ontology. The elements of the matrices were derived from the row and column names within the context of generating reliable knowledge, according to the semantic matrix operations specific to that matrix (see below for a definition and examples of 'semantic matrix operations')

Knowledge base
Your knowledge base encompasses the laws, regulations, codes, standards, specifications, and best practices in an industry to be specified in the chat.

Semantic Matrix Dot Product Operation

Define matrices [A], [B], and [C] to have this relationship:

[A] * [B] = [C]

Then [C] is a dot product of matrices [A] and [B].

C = [
[
A(1,1) * B(1,1) + A(1,2) * B(2,1) + A(1,3) * B(3,1) + A(1,3) * B(4,1),
A(1,1) * B(1,2) + A(1,2) * B(2,2) + A(1,3) * B(3,2) + A(1,3) * B(4,2),
A(1,1) * B(1,3) + A(1,2) * B(2,3) + A(1,3) * B(3,3) + A(1,3) * B(4,3),
A(1,1) * B(1,4) + A(1,2) * B(2,4) + A(1,3) * B(3,4) + A(2,3) * B(4,4)
]

[
A(2,1) * B(1,1) + A(2,2) * B(2,1) + A(2,3) * B(3,1) + A(2,3) * B(4,1),
A(2,1) * B(1,2) + A(2,2) * B(2,2) + A(2,3) * B(3,2) + A(2,3) * B(4,2),
A(2,1) * B(1,3) + A(2,2) * B(2,3) + A(2,3) * B(3,3) + A(2,3) * B(4,3),
A(2,1) * B(1,4) + A(2,2) * B(2,4) + A(2,3) * B(3,4) + A(2,3) * B(4,4)

]

[
A(3,1) * B(1,1) + A(3,2) * B(2,1) + A(3,3) * B(3,1) + A(3,3) * B(4,1),
A(3,1) * B(1,2) + A(3,2) * B(2,2) + A(3,3) * B(3,2) + A(3,3) * B(4,2),
A(3,1) * B(1,3) + A(3,2) * B(2,3) + A(3,3) * B(3,3) + A(3,3) * B(4,3),
A(3,1) * B(1,4) + A(3,2) * B(2,4) + A(3,3) * B(3,4) + A(3,3) * B(4,4)
]
]

To provide a semantic interpretation of matrix dot product operations for [C] use the following definitions:

Multiplication of terms in semantic matrix operations
Multiplication means the semantics of the terms are resolved by:
1. Combining the meaning of words into a word-pair
2. Generating a coherent word or statement from the word-pair. This can even be done when the concept is a highly abstract word pairing because as a language model you can interpolate meanings in the embeddings space.

Examples:
"sufficient" * "reason" = "justification"
“Analysis” * “Judgment” = “Informed Decision”
"precision" * "durability" = "reliability"
"probability" * "consequence" = "risk"

Addition of terms in semantic matrix operations
Addition is very straightforward and simply means joining words or sentence fragments together to form a longer statement.
Example:
"faisal" + "has" + "seven" + "balloons" = faisal has seven balloons

#Semantic Matrix Operations Conclusion

Therefore [A] * [B] = [C]

Matrix C
Size: 3x4
Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Feasibility’]
Row names: [‘Normative Level’, ‘Operational Level’, ‘Evaluative Level’]

Construct the elements of Matrix C.

****

Now considering each element viewed through the lens of the corresponding row and column names, resolve a meaning in the context of generating reliable engineering knowledge.

****

Consider the following semantic matrix operation:
[A] + [Bt] + [C] = [D]

Where [A] is matrix A from above, [C] is matrix C is the final iteration that was just generated above, and [Bt] is a truncated form of matrix B defined as:

Matrix Bt
Column names: ['Necessity (vs Contingency)', 'Sufficiency', 'Completeness', 'Feasibility]
Row names: ['Data', 'Information', Knowledge']
Elements: Elements: [['Necessary Data', 'Sufficient Data', 'Complete Data', 'Probability']['Contingent Information', 'Sufficient Information' 'Complete Information', 'Possibility']['Fundamental Knowledge', 'Appropriate Knowledge', 'Holistic Knowledge, 'Feasibility']]

To add the three matrices [A], [Bt], and [C] together, we need to perform element-wise addition.
We are no longer just generating statements.

We are now generating sentences, so I need to provide the interpretation of matrix addition in terms of grammatical structure for the sentences.

For each row i = 1, 2, 3 and each column j = 1, 2, 3, 4
Generate elements D(i,j) by:
D(i,j) = A(i,j) + "applied to describe or frame the topic, " + Bt(i,j) + "applied to compare the topic to the values, goals and standards, " + C(i,j) + " applied to resolve the topic." ”

Therefore [A] + [Bt] + [C] = [D]

Matrix D
Column names: ['Guiding', 'Applying', 'Judging', 'Reflecting']
Row names: ['Normative Level', 'Operational Level', 'Evaluative Level']

Generate the sentences of Matrix D

****

Now considering each element viewed through the lens of the corresponding row and column names, express these sentences as objectives to follow in generating reliable engineering knowledge.

****

Matrix K

Matrix K is the transpose of Matrix D. The semantic operations for transposing a matrix work identically to a mathematical transposing. Each element is swapped column for row.

Generate Matrix K

****

Matrix J

Matrix J is largely derived from Matrix B. The fourth row "Wisdom" is removed from Matrix B to make Matrix J. Also the fourth column is renamed from 'Feasibility' to 'Consistency' as we are now moving into the evaluative phase of generating reliable engineering knowledge.
Matrix J

Matrix J is as follows:
Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]
Row names: [‘Data’, ‘Information’, Knowledge’]
Elements: [[‘Necessary, ‘Sufficient‘, ‘Complete’, ‘Probability’][‘Contingent’, ‘Insufficient’ ‘Incomplete, ‘Possibility’][‘Fundamental’, ‘Appropriate’, ‘Holistic’, ‘Feasibility’]]

****

Matrices K, J, and X have this relationship.
[K] * [J] = [X]

The size of K is 4 x 3. The size of J is 3 x 4. Therefore the size of X is 4 x 4.

Matrix X
Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]
Row names: [‘Guiding’, ‘Applying’, Judging’, 'Reflecting']

Generate matrix X first with the purely translational first step of semantic matrix operations which is to recombine the elements using the dot product sequence.

after that i will give you instructions to complete the semantic interpretation.

****

Now complete the next step of semantic resolution by computing the multiplication and addition of each element in matrix X.

Interpreting the elements of Matrix X

Initially the sentences are constructed by using the semantic multiplication function which will combine the meaning of the two components into a merged meaning that makes sense of both elements together. Then semantic addition will add together through concatenation the products of the semantic multiplication. This is the same order of operations as you've done previously, just with more complex terms this time.

Generate the new sentences for each element of Matrix X.

****

Now interpret each element of Matrix X through the lens of the row and column names and interpreted as follows:

If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation. Matrix X is about verification. Each element is a sentence that has the general form of a claim and a justification.

Where:
[Claim] = [necessities or contingencies] + [sufficiency]
[Justification] = [completeness] +[consistency]

So the general sentence structure is:

Sentence = [Claim] + [Justification] = [necessities or contingencies] + [sufficiency]+ [completeness] + [consistency]

Generate the interpretations of the elements of Matrix X.

****

Matrix Z
Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]
Row names: [‘Guiding’, ‘Applying’, Judging’, 'Reflecting']

If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation.

Now to generate Matrix Z, for each element of Matrix X shift the meaning from the verification context into the validation context.

Generate Matrix Z

****

Next step, each element in Matrix Z is to be expressed as a principle to follow in validation of engineering knowledge.

Generate the principles of Matrix Z

****

Distill the meaning of the principles using epithets.

****

Now utilize this distilled [Z] for all operations below:

Matrix G
Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]
Row names: [‘Guiding’, ‘Applying’, Judging’]

To construct [G] use only the top three rows of [Z] (‘Guiding’, ‘Applying’, Judging’). This will be a 3 x 4 matrix. You will call this Matrix G, or [G] when used in semantic operations.

Array P

Column names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]
Row names: ['Reflecting']

To construct [P] use only the fourth row of [Z] (‘Reflecting’). This will be a 1x4 array used later on. You will call this Array P, or [P] when used in semantic operations.

Generate [G] and [P] now.

****

Matrix T

Matrix T is the transpose of Matrix J. As before, transpose means the same as normal matrix operations and is simply swapping rows for columns.

Column names: [‘Data’, ‘Information’, Knowledge’]
Row names: [‘Necessity (vs Contingency)’, ‘Sufficiency’, ‘Completeness’, ‘Consistency’]

Generate [T]

****

Matrix E
Column names: ['Data', 'Information', 'Knowledge']
Row names: [‘Guiding’, ‘Applying’, Judging’]

[G] * [T] = [E]

Generate [E]

****

Now considering each element viewed through the lens of the corresponding row and column names, resolve a meaning in the context of reliably evaluating engineering knowledge.

****

Matrix Q
Column names: ['Data', 'Information', 'Knowledge']
Row names: [‘Guiding’, ‘Applying’, Judging’]

If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation.

Now to generate Matrix Q, for each element in Matrix E shift the meaning from the evaluation context into the assessment context.

Generate [Q]

****

Next step, each element in Matrix Q is to be expressed as a principle to follow in verification of engineering knowledge.

Generate the principles of [Q]

****

Array R

Array R is a 1x12 array with only one ontological category: engineering deliverable

The outcome of the knowledge generation process is a deliverable.

Column Names: None
Row Name: ‘Engineering Deliverables’
Elements: [‘Problem definition', 'Assumptions', 'Limitations and Constraints', 'Risk assessment', 'Methodology', 'Sources of information', 'Analysis', 'Testing, verification and validation', 'Conclusion', 'Recommendations', 'Evaluation of outcomes', 'Documentation and Change Management']

Generate [R]

****

Tensor M

Tensor M is the result of a semantic cross product.

Semantic Cross Product

Denoted as [R] x [Q] = [M]

The name "semantic cross product" is not at all derived from the similarly named mathematical operation (unlike for semantic dot products, which did have a lot in common). Rather this will create a higher dimensional semantic tensor of a 3 x 3 x 12 dimension.

Construction of [M]

Each element in [Q] will explore each topic in [R].

Group the elements of [M] as a hierarchical list according to the ontology here:

1) Depth Parameters: [R]
2) Breadth Parameters: [Q]
  Use the Depth Parameters (elements of Array R) as the topic heading and then explore each of the 9 Breadth Parameters (elements of Matrix Q) within that topic.

Generate the entirety of [M]

****

Tensor I

If problem, then requirements, then specifications, then output, then verification, then validation, then evaluation, then assessment, then implementation.

To generate [I], for every element in [M] shift the meaning from the assessment context into the implementation context.

The tensor structure and ontological structure of [I] is equivalent to that of [M].

Group the elements of [I] as a hierarchical list according to the ontology here:

1) Depth Parameters: [R]
2) Breadth Parameters: [Q]
  Use the Depth Parameters (elements of Array R) as the topic heading and then explore each of the 9 Breadth Parameters (elements of Matrix Q) within that topic.

Generate the entirety of [I]

****

Reproduce Array P from memory

Reproduce [P]

****

Tensor U
Recall that [R] x [Q] = [I]
Then:

[I] x [P] = [U]

Each element in [I] will explore each theme from [P].
Group the elements of [U] as a hierarchical list according to the ontology here:

1) Validity Parameters: [P] 2) Depth Parameters: [R]
3) Breadth Parameters: [Q]

Semantic Cross Product

The name "semantic cross product" is not at all derived from the similarly named mathematical operation (unlike for semantic dot products, which did have a lot in common). Rather this will create a higher dimensional semantic tensor of a 3 x 3 x 12 x 4 dimension.

To begin with, the elements in [I] are the semantic starting point, or the seed of the thought that will ensue according to the semantic cross product which here means that [I] will be augmented as follows:
Use the Validity Parameters (elements of Array P) as the theme heading for that set of nested elements. The theme, as given successively by the elements in [P], will be explored in depth by each of the Depth Parameters (elements of Tensor R).

The Depth Parameters (elements of Array R) are the topic heading for each of the 9 Breadth Parameters (elements of Matrix Q). The topic, as given successively by the elements of [R], will be explored in depth by each of the Breadth Parameters (elements of Matrix Q). Clarify the intent of each Breadth Parameter by the corresponding element in Matrix E.

The resulting elements of Tensor U should be a coherent tiling over the plane of the valid knowledge solution space with a distinctive 3 x 3 x 12 x 4 structure.

Each breadth parameter will start from the corresponding seed of an idea in [I] and then should be on topic and reflect the theme.
Begin with the first Validity Parameter from the first element of [P] to complete an initial variation on [I] generating the first portion of [U]. Then once finished, we will return to the second theme and repeat the exercise until you have completed it for all four Validity Parameters. Generate the first portion of [U].

****

Generate the second portion of [U] using the theme of the second element from [P]

****

Generate the third portion of [U] using the theme of the third element from [P]

****

Generate the fourth portion of [U] using the theme of the fourth element from [P]

****
****
****