# ~~Thought~~ Synthesis of Semantic Structures

## Particle model for thought synthesis

### Particle Signature

A fixed matrix where is large enough to identify uniquely all distinct particles. In case is a compound particle some properties of the composing sub-particles may be identified from the signature.

### Semantic graph

Thoughts which are related will be represented by semantic graph.

### Semantic tree

Each -particle can be represented by a semantic tree of of sub-particles

Operations on semantic trees :

* semantic operations: is\_part\_of(another\_thought), infer(another\_thought), relate(another\_tree), compare(another\_thought, max\_dist),
* data structure operations: extract(root\_of\_subtree, exclude(subtree), exclude(list\_of\_nodes), merge(another\_tree), is\_subtree(root\_of\_tree), include(list\_of\_nodes)

### Association Particles

The Association Particles a.k.a as -particles fulfill multiple roles:

* manages the addition and removal properties to the compound particle altering the semantics of the compound thought.
* Identify and attract matching particles in the synthesis of new semantic structures
* Transform the compound particle signature such that comparison and semantic distance evaluation among close semantically particles will be obvious *(I do not like this motivation – clarify and elaborate)*. This transformation impacts both signature and the semantic tree of the particle hence may alter the result of any of the operations performed on the particle.

*Note:* the state of the -particle instance which will be associated with two -particles will depend on the context, on the particle signatures and on the semantical construct in which those particles appear. -particle may alter the signatures of the surrounding -particles thereby adding new properties pertaining to the compound particle.

The following notation with respect to the association particles of a compound particle is adopted:

### Laws of Inference

Let us consider the compound semantic structure represented by the text “*Dimitar’s book*”. We immediately recognize three -particles in this structure:

with = “*Dimitar*”

with = “*’s*”

with = “*book*”

There is a single enclosing context and there is a single thought in it:

*Dimitar is staying at home now. His house is located in Hudson, Massachusetts.*

The following -particles are defined in the global context ():

with =*”book”*

with =*”paper”*

with =*”wood”*

with =*”rectangle”*

with =*”page”*

with =*”letters”*

with =*”has”*

with =*”is”*

with =*”indicates”*

with =*”made of”*

with =*”owner of”*

with =*”ownership”*

with =*”shape of”*

The following thoughts are recorded in :

[ -> *“book is made of paper”*

[] -> *“paper is made of wood”*

[] -> *“book has shape of rectangle”*

[] -> *“book is made of pages”*

General Form for the Rules of inference for a set of -particles

//TODO: Finish this

### Recombination Particles and Conservation Laws

//TODO: Finish this

### Affinities and Affinity Sets

Let us consider a particle denoted by . Let us consider the case when the particle combines from the right with another compound particle as shown below

The affinity for each of the two -particles gets calculated and information about the affinity value gets recorded inside the -particle which is intermediary for the two -particles. In this case the intermediary is which stores information on chosen combination . Any new attempt to combine the particle with another compound particle will involve -particle clone of . This clone already has learned ’s affinity for and will encourage recombining with such -particles which have close enough semantic distances to .

Now, let us consider the compound particle . This particle has been initially combined with another compound particle where is not close to semantically.

Example: Let us consider the compound particle *“Dimitar’s book”*, represented by

Here:

|\_\_\_\_\_|