PATTERNS OF THOUGHT [1]

Ulf Grenander

Version 5.5

March 2005

ABSTRACT

We offer an axiomatic system for the representation of human thought processes including emotions, affects and diffuse thinking. The system employs the regular structures of metric pattern theory and is probabilistic in order to account for non-deterministc thinking. It is controlled by an algebra with thoughts as objects and with the algebraic operations SIMILAR, COMPOSE, ABSTRACT, GENERALIZE/SPECIALIZE, MUTATE, CROSSOVER, MOD, COMPLETE, as well as the transformations GENRE, CREATE IDEAS and MEMORY of memory parameters . It accounts for free associations, generalization, abstraction, deep thought, inference, dreaming, inventing new concepts and recurrent thought. The personality is parametrized by inputted values. Thoughts in a random chatter compete for domination to reach the conscious level. Generalization is formalized in terms of an additive semi-group $\mathcal{G} = \{\mathcal{G}^{power}; power \in \mathbf{N}\}$ *involving the generalization operator* $\mathcal{G}^{power}$*. Thought patterns are sets of thoughts invariant w.r.t. the modality group. A software package GOLEM is intended to illustrate the actual working of such a system; it serves as an illustration of how well (or how badly) the model produces anthropomorhic behavior.*

TABLE OF CONTENTS

[1]This work has been supported by DAAH04-96- NSF DMS-00774276

OUR WORKING HYPOTHESIS:

HUMAN THOUGHT CAN BE REPRESENTED BY THE REGULAR STRUCTURES OF GENERAL PATTERN THEORY

1 A Theory of Mind ?

The human mind is a mystery. Although it is so close to us - we live in and with it - we do not really understand how it works. Philosophers and thinkers in general have struggled with this question for millenia and much has been learnt, most in a vague and unspecific form. Many attempts have been tried to describe it through logical schemata. But human thought is (normally) not completeley rigid; it is only partly predictable. Say that an external stimulus makes us think of a fast automobile racing toward us. It is quite likely that our thoughts will then be of fear but someone may react without fear. The thinking process is certainly not fully deterministic.

We instinctively avoid believing that our thoughts are generated by a more or less mechanical device. We do not want to be seen as machines. Hence we tend to reject statements like the one by Karl Vogt, a 19th century German philosopher, who stated that the brain produces thoughts as the liver produces bile, or the kidneys produce urine. But few would deny that the material substrate of thought, the neural system of the brain, obeys the laws of physics/chemistry, so that it is not impossible that *there may exist mathematical laws of thought* in principle derivable from physic/chemistry. Such laws could be probabilistic, as is statistical mechanics, with the ability to represent thought processes in terms of random variations. It would be premature to try to derive such laws from first principles; instead we shall present speculations with no firm support in empirics, just ideas that seem plausible (to the author).

We shall consider thought processes that include logical thinking, but this is only one mode among many. We follow Damasio (1999) who discusses the dominating role of emotions for human thought in an elegant and convincing way. We shall include fear, love, emotions...But recall Pascal's dictum: "The heart has its reasons, of which reason knows nothing." Indeed, we know only little about the functioning of emotional thought processes. But wait! *We are not after a general theory of human thought*, indeed we do not believe in such an endeavor. Instead we will try to present only a shell, a scheme only, of human thought that will have to be filled with content different for each individual, setting different values to the (many) mind parameters. This content can have its origin in the genetic and cultural background in which the individual lives, as well as being formed by earlier experiences leading to a dynamically changing mind. Thus we will concentrate on the *general architecture* of the building rather than on its detailed specification.

We shall deal with the mind without reference to the brain. A completely reductionist mind theory would be based on neuro-physiological knowledge, deriving mental processes from what is known about their cerebral substrate. We are certainly in favor of such an approach, but in the absence of a complete brain theory it is not feasible at present. Instead we shall base the construction on introspection and on what has been learnt over the centuries in a less formal setting about the working of the mind by clinicians and what can be found in novels, poetry and plays. This non-positivist attitude is open to the criticism that it leads to no testable hypothesis. We admit that this is true, at least in the immediate future, and accept the criticism.

The last several decades have witnessed remarkable process in the neurophysiology of the brain - many elegant experiments have thrown light on the functioning of neurons, at first for single neurons and more recently for cell assemblies. This has led to an impressive body of empirical knowledge about the brain. Some researchers have tried to increase our understanding of the human mind through mathematical studies of the firing rates of neurons. It seems doubtful to this author whether mathematical work of this type has led to more insight in the human mind than what the purely experimental results have shown. This author is all in favor of such a reductionist approach: it is necessary but not sufficient! Perhaps such mathematical studies can help in understanding how Ratus Ratus run in mazes, but for the analysis of the mind of Homo Sapiens they are flagrantly insufficient. We are aware of the many talented and knowledgable researchers applying mathematical analysis to neural rates, myopically concentrating on neural behavior while neglecting high level activities of the human mind. Alas, they include even such personalities as *sagax* Mumford. We beg the indulgence of the diligent firing rate mathematicians if we put more trust in the introspective wisdom of Aristotle, Shakespeare and William James (perhaps also that of his brother) as well as in the collected clinical experience of psychiatrists/neurologists when it comes to describing and analyzing the high leval mental activities. Expressed differently, our approach could perhaps be stated as studying the software of the mind rather than the hardware.

1.1 What We Shall Do.

Immanuel Kant:" Human reason is
by nature architectonic"

Our goal is to build a model of the mind in pattern theoretic terms: Starting from simple, atomic, mental entities (the generators of pattern theory) we shall combine them into regular structures (configurations) controlled by probabilistic rules of connections. In this way patterns of thought will be built *pace* Kant as hierarchies of more and more complex structures in which we shall introduce a *calculus of thoughts*. Note that we are aiming for representations of ideas of different types: deductive reasoning (including mistakes), feelings like love and hate, doubts and questions and many others.

We shall limit ourselves in this paper to outlining a mathematical representation theory but hope that it will be applied to knowledge available to neurologists/psychiatrists.

So we shall search for answers to the following questions:

What are the mental objects that make up the mind?

What are the mental operations that act upon the objects?

How do these objects combine to form thoughts?

1.2 Judging a Mind Model.

Carver Mead: " ...you understand something when you can build it"

But here is the rub. Since we are admitting that our mind model does not rely on firmly established facts, neither on neurophysiological theory, nor on objective cognitive facts, how are we going to judge it? What criterion will be applied to evalate its validity? It is easy and tempting to speculate, but without self criticism we will have no guarantee that we have achieved more than an amusing (?) thought experiment. It is tempting to get immersed in abstract and too general speculations: here, as elsewhere, the devil is in the details. But we shall spend much time on working out the details.

Appealing to Carver Mead's motto we shall *build* a mind model in software, expressing our theoretical constructs in program modules. We shall be satisfied with the model, at least temporarily, if the program executes in a way that seems reasonably close to what our intuition expects of a human mind. This is somewhat related to Turing's celebrated test, but out goal is less ambitious. We are not in the business of artificial intelligence, we do not intend to create intelligence or a simile of it. Instead, our more modest goal is to present a shell that can be filled with specifically chosen entities resulting in a coherent scheme consistent with what we believe is human thought.

In passing we mention Joseph Weizenbaum's celebrated program ELIZA that mimics conversation between a patient and an analyst. It attracted a lot of attention, even a belief in the psychoterapist it simulates, to the extent that its inventor came to be surprised and even embarrassed by the misguided enthusiasm that the ELIZA program generated. The code supporting the program is simple, but the behavior is, at first, quite impressive. What we are after, however, is code that rests on a pattern theoretic analysis of the human mind specifying the details of mental processes..

As we shall see it will take some complex software to achieve our goal, even roughly. To facilitate programming we shall write in MATLAB although this will result in fairly slow execution. In a later stage we may compile the code into C++ or into executables, but at the moment we are not concerned with computational speed.

2 Mental Architecture

Hence we shall *build* mind states from primitives, elements that express simple mental entities: feelings and emotions, thoughts about the external world as well as about the inner self, doubts and assertions, logical deductions and inferences. We shall allow the reasoning of the mind to be incomplete, inconsistent and, well, unreasonable. Influenced by Damasio (1999), and perhaps by Vygotskij (1962), we shall include feelings and their interaction with conscious thought. We shall be guided by introspection, our own of course, but also by that of others accumulated over eons in novels, poetry, plays. Perhaps we can also find help in figurative paintings and other art forms. In addition, a multitude of philosophers and psychologists have offered insight into the working of the human psyche in a more technical sense. Recently, scholars in cognitive science and artificial intelligence have presented schemes for the understanding of natural and man-made minds, often in a controversial form. We shall borrow from many of these sources, somtimes without explicit attribution. The basic idea in what we shall be doing was suggested in Grenander (1981).

Our approach will be hierarchical, architectonic, so that we will successively combine simple mental entities into complex and larger ones. In software engineering this attitude is known as the "divide and conquer strategy", in image processing as "coarse to fine algoritms", in cognitive science as "compositional". Whatever its name, this approach is based on the belief that it will enable computations whether these are carried out by a neural system or by a silicon substrate.

3 An Algebra of Human Thought

Wittgenstein: "The picture is a model of reality. To the objects correspond in the picture the elements of the picture. The picture consists in the fact that its elements are combined with one another in a definite way".

Let us begin with an axiomatic description of the algebra, to be followed by a concrete discussion elucidating the axioms.

3.1 Primitive Ideas

Thoughts are formed as compositions of generators, primitive ideas, in some generator space, $g \in G$. G is finite but its cardinality can vary with time as the mind develops. A generator g has an arbitrary (variable) number of in-bonds with the same bond value $\beta_i(g)$, and a fixed number $\omega_{out}(g)$ of outbonds with bond values $\beta_j(g); j = 1, 2, ...\omega(g)$.

3.2 Modalities

Bond values are from a lattice $\mathcal{M}$ of subsets of G.

3.3 Similarities of Ideas

. On the generator space G *there is defined a permutatiuon group* S*, the modality group. Two generators* g_1 *and* g_2 *are said to be similar if* $\exists s \in S \ni g_1 = sg_2$*. The s-operation preserves bonds.*

3.4 Compositions of Primitive Ideas

A thought is a labelled acyclic directed graph $thought = \sigma(g_1, g_2, ...g_n); g_i \in G$ *where the connector graph* σ *connects some* j*th out-bond* $\beta_j(g_{i_1})$ *of generator* g_{i_1} *to an in-bond of generator* g_{i_2}*. The modality group is extended to thoughts by* $s\ thought = \sigma(sg_1, sg_2, ...sg_n)$*.*

3.5 Regular Thoughts

A thought is said to be regular if only outbonds connect to inbonds carrying the same bond value: regularity $\mathcal{R}$*. The set of all regular thoughts for specified* $G, \mathcal{M}...$ *is called* $MIND(\mathcal{R})$*.* A given $MIND(\mathcal{R})$ is called a personality.

3.6 Thought Patterns

A subset $\mathcal{P} \subset MIND(\mathcal{R})$ *is called a thought pattern if it is invariant with respect to the modality group* S*.*

3.7 Probabilities of Thoughts

The probability measure P *generating regular thoughts has a density (a function of the variable "thought") that is proportional to the value of a weight function* $Q(g_i)$ *for every primitive ideat* $g_i \in thought$*, proportional to the value of an acceptor function* $A(\beta_1, \beta_2)$ *of any connected pair of bond values* β_1, β_2*, and proportional to* π_n*, the probability of the thought consisting of* n *primitive thoughts.* The power of thinking is measured by the $\pi' s$

3.8 Inference

Inferential thought processes (logically correct or not) are organized in terms of conditional probabilities w.r.t. P*.*

3.9 Completion

Thoughts are made meaningful by the application of the COMPLETE operation that closes out-bonds.

3.10 Generalization

Thoughts are generalized by the application of the MOD operation from a semi-group G*.*

3.11 Abstraction

The device of encapsulation abstracts thoughts to ideas that can be referred to as independent units; they are added to the generator space G.

3.12 Mind Development

The MIND develops by changes in its Q and A parameters, as well as in the generator space, due to mental experiences.

4 Building Mental States: Chemistry of the Mind

Now let us discuss the axiomatics in more detail in the context of of Pattern Theory [2]. First the primitives of the mind. The primitive ideas, the generators, usually to be denoted by symbols like g or g_1 or g_i and so on. The role of the generators is to represent the simplest of the ideas of a particular mind. The set of all generators available to a particular mind will be denoted by G, the *generator space*. We are thinking of the mind as a dynamic entity, changing over time: new primitives may be created, others forgotten, but to begin with we shall treat the generator space as fixed.

But how should we choose it? The choice must depend upon the environment, physical and psychological, in which the mind lives. Also upon what it has experienced earlier and on its previous thoughts. The choice ought to express the personality and peculiarities of a particular mind, as will be made clearer in section 2.3. We shall be guided in making this choice by the discussion of Human Universals in Brown (1991).

Also we shall appeal to a

PRINCIPLE OF ISOLATION: *The MIND strives to make thoughts meaningful so that they can standing alone; hence they should be complete (see below for this concept). We can speak of a completion pressure.*

The environment contains things, objects, but also events that are happening or have happened recently, and other non-physical facts. Recall Wittgenstein's dictum:"the world consists of facts, not of things",Tractatus Logicus-Philosophicus (see References). We shall include physical things like

$$\{dog, cat, human, John, table, car...\} \subset G$$

but also non-physical ideas like

$$\{thought, hate, walk, fear, say, ...\} \subset G$$

[2]A complete presentation of pattern theory can be found in Grenander: General Pattern Theory, referred to as GPT, see References

and events like

$$\{wedding, fight, transaction\} \subset G$$

to mention but a few.

But how should we organize such generators? One way is to order them through a Linnean taxonomy in organizational trees like the one shown in Figure 4.1 (or forests)

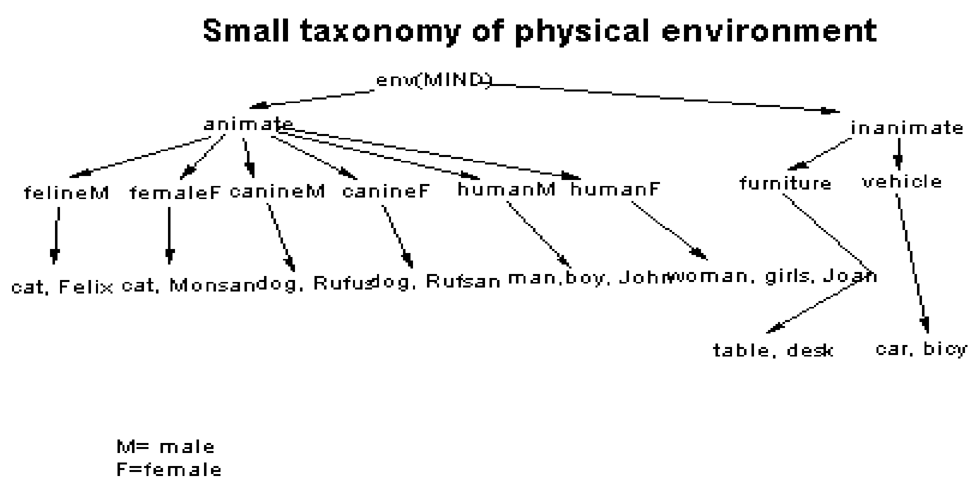


Figure 4.1

Most of the elements in this taxonomy are self-explanatory, with one exception: note that the generator "dogM" is a generic symbol for male dogs in general, while "Rufus" signifies a particular dog. The observant reader will notice, however, that in order that this represent a real partition, the set "dogM" must be defined as different from "Rufus". We shall return to this later.

Non-physical generators are at least as important as things. For example, $g = think$ representing someone's thinking, or $g = say$ meaning a statement is being made by someone. Here that *someone* can be "self" or another human member of G. There will be many non-physical generators: "doubt", "question", "answer", "write", and so on. Combining them we get diagrams like those in Figure 4.2 where the interpretation of a diagram is given on the right side. We have used notation "think1" to indicate that it has one arrow (out-bond) emanating out from it, "question2" has two arrows from it and so on so that "question2" is different from "question3". This is formalized through the notion of arity to be discussed in section 4.2.

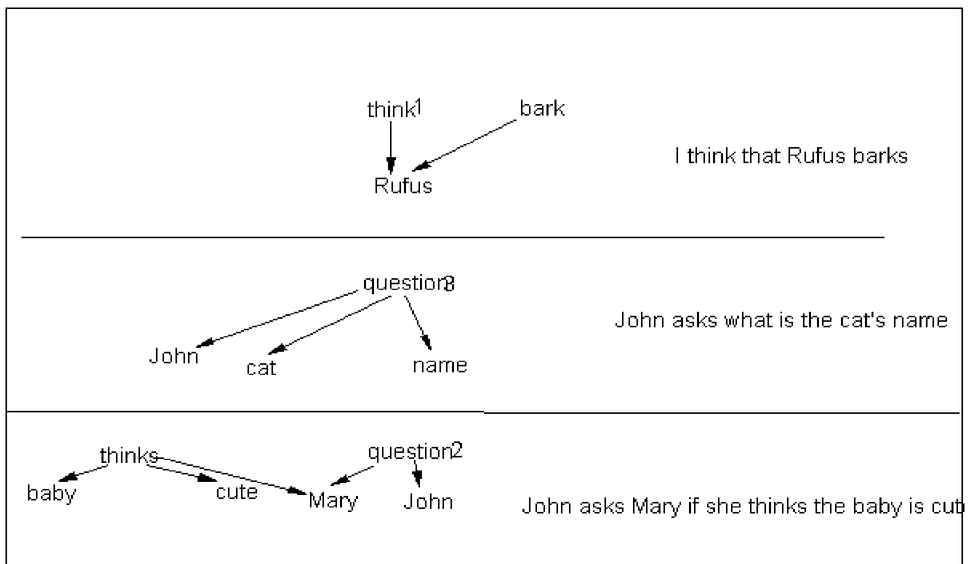


Figure 4.2 Composition of simple ideas

4.1 Caveat.

It is tempting to think of the generators as words and the diagrams as sentences, but this is not at all what we have in mind. Recall the Sapir-Whorf famous hypothesis: "...the fact of the matter is that the *real world* is to a large extent unconsciously built up on the language habits of the group" and that our thought processes are directly or indirectly made up of words. We do *not* subscribe to this hypothesis. On the contrary, our construction of a formal mind will be done independently of language to the extent that this is possible. It is not easy to free onself from the straightjacket of language, but we shall try to do this in the following to the extent it is possible. We shall deal with *concepts - not words*. Actually, we will be forced to use notation more precise than words alone. As an example we may distinguish between generators like $g_1 = activity_1$ and $g_2 = activity_2$, with the usage of g_1: "John works" and of g_2: "John works with a hammer"; see the remarks at the end of last section. We shall make many such distinctions and insist that they are more than mere technicalities; they are needed in order that the mind representation be precise. But we do not insist that the mind and its thinking be precise, only that our representations of the thinking be presice.

To examplify the above: the meaning of the generator $g = dog$ is reasonbly clear, while $g = question$ requires some explantion. It is certainly not the word "question" itself; instead we intend it to represent the *act* of questioning, someone asks someone else about something.

Therefore we shall strive for a *language independent mind theory*, admitting that we have only partially realized this goal, an extra-linguistic representation of a mind.

4.2 Levels, Modalities, and Arities in Mind Space.

In Figure 2.1.1 we have arranged the generators in *levels*:$g = catM$ is situated below $g = felineM$ which is on the next higher level in the taxonomy partition. But we shall formalize the concept of *level* in another way. We first introduce the concept of *modality*.

The generator space will be partitioned into subsets, modalities $M(m) \subset G; m = 1, 2, ...card(\mathcal{M})$,

$$G = \cup_{m=1}^{\mathcal{M}} M(M) \tag{1}$$

together with a *partial* ordering so that $m_1 \downarrow m_2$ for some, pairs $m_1, m_2 = 1, 2, \dots M$ while other pairs may not be ordered with respect to each other. A modality will contain generators with related meaning, for example

$$color = \{red, blue, green, yellow, \dots\} \in \mathcal{M} \tag{2}$$

or

$$movement = \{run, jump, turn, still, \dots\} \in \mathcal{M} \tag{3}$$

where the set of all modalities has been denoted by $\mathcal{M}$ and enumerated $m = 1, 2, \dots card(\mathcal{M})$ It is the *modality lattice*. Occassionally we shall make use of the concept *modality mixes*, meaning unions of modalities. An example of a modality mix is $action1 \cup action2$. An extreme modality is $mod = \mathcal{M}$ itself, all modalities together. Modalities are denoted by capital letters in contrast to the primitive ideas which are written with lower case letters.

The generators $g_1 = bark$ and $g_2 = dog$ are naturally ordered, $g_1 \downarrow g_2$, but $g_3 = yellow$ and $g_4 = smooth$ do not appear to have any natural order. Thus the ordering is partial rather than complelete.

With the convention that all 'object'-generators, animate or inanimate, are put on *level* one we shall use the

DEFINITION: The level $level(g)$ of a generator g is the shortest length l of chains

$$g \downarrow g_{l-1} \downarrow g_{l-2} \downarrow g_{l-3} \downarrow \ldots \downarrow g_1; level(g_1) = 1 \tag{4}$$

Thus a generator g with $l = level(g) > 1$ can be *connected downwards* to a number of generators on level $l - 1$. We shall need a concept characterizing the connectivity of generators, namely the *out-arity*, somtimes called down-arity.

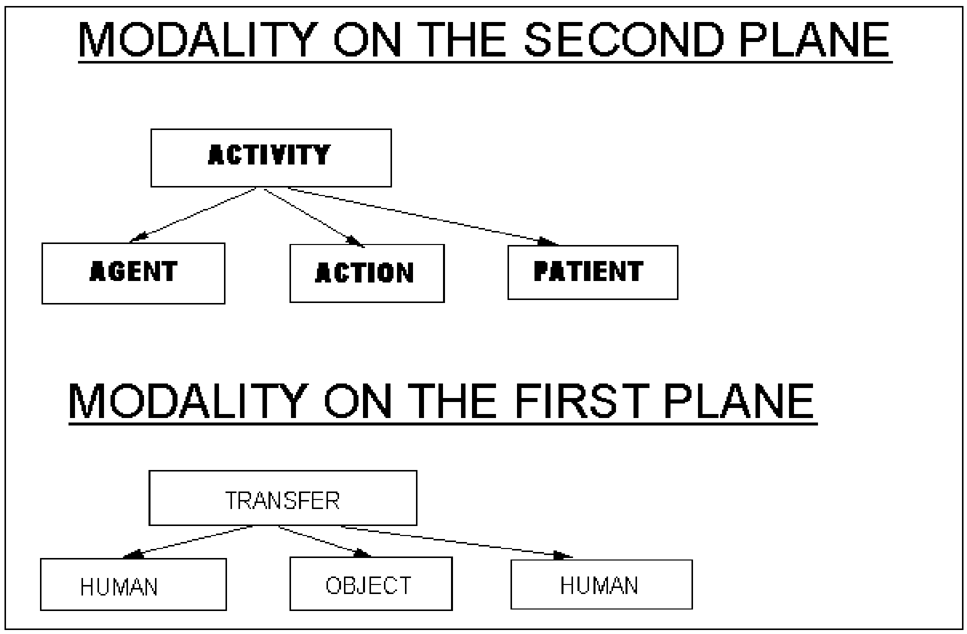
Behind this constructuion is the PRINCIPLE OF ISOLATION. The primitive thoughts on level 1 can stand alone and still be meaningful. The concept of *new Idea*, to be introduced later, is meant to be meaningful standing alone; hence it should belong to level 1. For a primitive thought to be on level L it should be possible to make it meaningful standing alone by adding primitive thougts from level L-1 and lower.

DEFINITION: The number of generators that can be connected downwards from g is called the arity $\omega(g)$ of g

In particular the generators on level 1, the 'things', all have arity 0. Hence $g_1 = bark$ in Figure 2.1.2 belongs to level 2 and arity 1, while $g_2 = Rufus$ belongs to level 1 and arity 0. But we need more information about the connectivity of generators. If $\omega = \omega(g) > 0$ we must decide to what other generators it can connect. This is the purpose of *bonds*, more precisely downward bonds. To each generator g and its downward jth bond we associate a subset of G denoted $\beta_j(g) \subset G; g \in G, ; j = 1, 2, \ldots \omega(g)$. We shall choose the subsets as modalities or modality mixes. For example, we may choose $\beta_1(love) = humanM$ and $\beta_2(love) = humanF$ for a heterosexual relation. The up-bonds will be the modality of the generator itself.

Of special importance are the "regular modalities", i.e. modalities such that its generators have the same arity and level that will lead to regular thoughts. The others, the irregular modalities, will be used for taxonomy but not for the formation of meaningful thoughts. In Appendix 3 the regular modalities are shown as rectangular boxes, while the irregular ones are indicated as diamond shaped boxes.

Modalities can be ordered by inclusion. For example, $ANIMAL \subset ANIMATE$. Note that this ordering is different from the partial order discussed above. It is clear that $\mathcal{M}$ forms a lattice, a POSET. This means that the ordering of modalities produces entities on different *planes of modality*. We have been denoting modalities (on the first plane) by capital letters and shall use bold faced capitals for the next plane. For example, we could have



REMARK. It may be natural to include in G together with a g also $mod(g)$. For example, in the subset of G with modality 'animalH' we can also include a $g = animalH$. Of course this works against seeing G as a partition but we shall do it anyway in order to make the mind more expressive when it comes to abstract thinking. The above construction is a refinement of the set up in GPT, Section 2.3.6.2.3.

4.3 A Concept of Concept.

We shall make the idea of a modality clearer. A concept, a modality M, is an item that can be used as an independent unit: it can connect to primitive thoughts as well as to other modalities as long as regularity is observed. The size of the set $M \in G$ can be just 1, but it should be bigger in order to serve as a concept usful for abstract thinking. As an example look at Figure 4.3.1

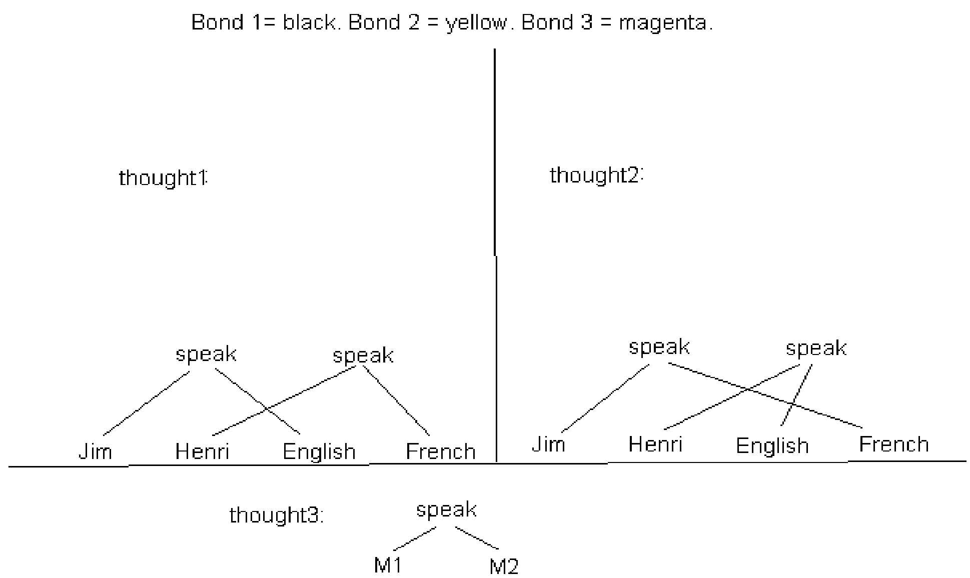


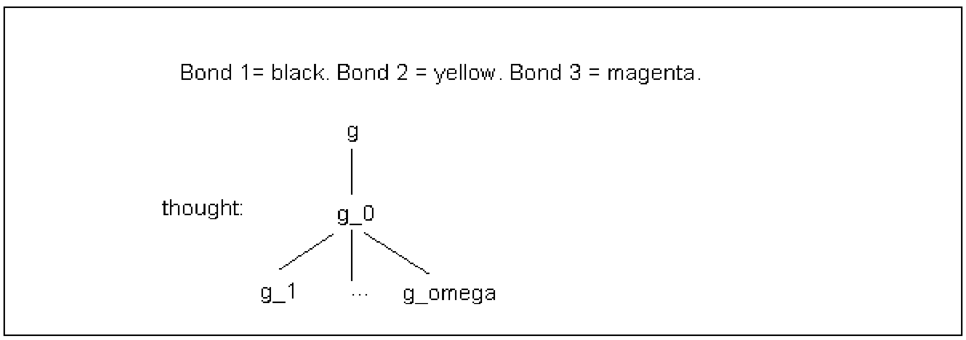
Figure 4.3.1
where *thought1* means that Jim speaks English and Henri speaks French,

while *thought2* says that Jim speaks French and Henri English. If $thought1, thought2 \in MIND$, we could form the modality $M1 = \{Jim, Henri\}$ and $M2 = \{English, French\}$ and consider thought3 regular, $thought3 \in MIND$. But if $thought1 \in MIND, thought2 \notin MIND$ the creation of the modalities $M1, M2$ would not be legal. We would have to introduce the contrived primitive ideas $speak1$ and $speak2$, the first one with out-bonds (Jim,English) and the second one with (Henri,French).

It is now apparent that *introducing a primitive thought g_0 with in-bond M and out-bonds $M1, ...M_\omega$, where ω is the arity of g_0, is allowed only if the thought in Figure 4.3.2 is regular for*

$$CONCEPT\ CONDITION : \forall g \in M \& \forall g_1 \in M_1 \& ... \& \forall g_\omega \in M_\omega \tag{5}$$

Note the repeated conjunctions in this condition, or, equivalently, Cartesian products of sets.



It is only if this condition is satisfied that the concept is allowed and the MOD operator can be used for abstract thinking.

Figure 4.3.2

4.4 Regularity of Mind States: Conformation of Thoughts

H. Poincare: "...ideas hooked together as links in a chain..."

Now let us combine generators (elementary ideas) into *configurations*, or thoughts, represented by diagrams like those in Figure 2.1.2 and written formally as

$$thought = \sigma(g_1, g_2, \ldots g_i, \ldots g_n) \tag{6}$$

where σ is a graph joining the generators $g_1, g_2, \ldots g_n$ in the diagram. In the first configuration in Figure 2.1.2 the diagram has three *sites* called 1) "think", 2) "Rufus" and 3) "bark", meaning "I think that Rufus barks". This graph has two *edges*, namely $1 \to 2$ and $1 \to 3$. We shall use subscripts $i = 1, ...n$ and so on to enumerate the generators, and $j = 1, ...m$ and so on for the edges (sometimes called connections) of the graph (sometimes for the down bonds of a single generator). Hence

$$n = size(c), m = size(\sigma) \tag{7}$$

so that in the above figure $n = 3, m = 2$.

A central concept in Pattern Theory is that of *regularity*. In the following we shall us two types of regularity:

DEFINITION:*A configuration* $thought = \sigma(g_1, g_2, \ldots g_i, \ldots g_n)$ *is said to be COMPLETELY REGULAR if any* j*th downbond* $\beta_j(g_i)$ *of any generator* g_i *in it is connected to a generator* g'_i *satisfying the bond relation*

$$\rho : g'_i \in \beta_j(g_i) \tag{8}$$

and a weaker concept:

DEFINITION:*A configuration, or thought,* $c = \sigma(g_1, g_2, \ldots g_i, \ldots g_n)$ *is said to be REGULAR if any connected* j*th downbond* $\beta_j(g_i)$ *satisfies the bond relation*

$$\rho : g'_i \in \beta_j(g_i) \tag{9}$$

In other words, a completely regular configuration has all its downbonds connected, but an incomplete has some downbond left open. In Figure 2.1.2 the second configuration is complete but if the connection $question \downarrow cat$ is broken it is incomplete (assuming that $\omega(question) = 2$).

We shall use the terms *complete* and *incomplete thoughts* when talking about configurations. When the configuration is made up of a single generator g it is called a *primitive (or elementary) idea*.

An incomplete or irregular thought may not have any acceptable interpretation and will therefore not always reach the level of consciousness. Nevertheles we shall study them, in accordance with our goal of studying thinking in all its imperfections, lack of consistency and with logical mistakes. At any instance there is a *chatter of competing thoughts* most of which will not reach the conscious level. More precisely, an incomplete thought or, more generally, an irregular configuration of elementary ideas will have a high energy (low probability). It will therefore quickly be deleted or modified to lower the energy; if it appears at all in consciousness it would be only for a fleeting moment.

The set of all regular configurations is called the (regular or completely regular) *configuration space*, the MIND, and is denoted by $MIND(\mathcal{C}(\mathcal{R}))$; it represents the set of all the thoughts that this mind is capable of. Note that the *regularity requirement of an idea means that its constituent sub-thoughts (ideas) conform.*

Note the resemblence to chemistry. Generators correspond to atoms, configurations (thoughts) to molecules, and bonds to bonds.

4.5 Creation of New Ideas

The MIND will be dynamic in that the generator space is not static, it changes over time. A complete thought (recall: no unconnected outbonds) can be made into an independent unit, a new generator that can be dealt with without reference to its internal structure. Hence $thought = \sigma(g_1, g_2, ...g_n)$ can be made into an *idea*, a new generator added to G on level 1 and hence with no out-bonds. We can think of this procedure as an *encapsulation process*.

For example, the complete thought in Figure 4.5.1 means that one should love one's neighbor. When encapsulated it becomes a new generator that could perhaps be named "CommandX", but in the automated working of the mind we shall use more neutral notation like $idea_k \in G$ with a counter k.

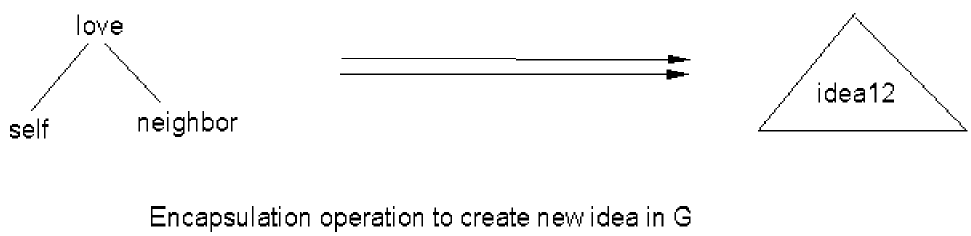


Figure 4.5.1

Now let us make this more precise. Say that the MIND has produced a conscious *thought* with the size $n = size(thought)$, and the generators $g_1, g_2, ...g_n$. With the probability $p_{create}(n)$ we shall abstract *thought* to a new idea $idea_k \in G$, where k is a counter that will be successively updated as new ideas are created. The probability distribution $\{p_{create}(\cdot)\}$ expresses the sophistication of MIND: if it allows big values of n with considerable probabilities, the MIND is capable of powerful abstraction and vice versa.

If the MIND's decision is "create", a new idea is created and it will be put in a new modality $IDEA_k$ on level 1 with the in-bond *idea*. The observant reader will have noticed that this differs slightly from our convention for defining modalities but will be useful for the coding. REMARK. In the NATLAB code for GOLEM this has not yet been implemented. Instead all new ideas are put in the same modality on level 1.

4.6 Patterns of Thought

Following the general principles of Pattern Theory [3] we introduce a *similarity group* S, the *modality group*, on the generator space G:

$$S = S_1 \times S_2 \times \times ... S_m \times ... \tag{10}$$

where S_m is the permutation group, the symmetric group, over the set of generators in the regular modality $mod \in \mathcal{M}$. If two generators g_1 and g_2 are similar in the sense that there is a group element $s \in S$ such that $g_1 = sg_2$, it is clear that this similarity induces a partition of the generator space into modalities as equivalence classes.

For example, $g_1 =$ "$John$" and $g_2 =$ "Jim" may be equivalent but probably not $g_1 =$ "$John$" and $g_2 =$ "$Mary$": this is an expression of the principle "arbitrariness of the sign" to quote de Saussure. This *modality group* enables the mind to *substitute mental entities* for another, i.e. abstract thinking, but preserving modalities, and avoiding incorrect references by not allowing primitive idea to be substituted for more than one other primitive idea. Hence the substitutions do indeed form a bijactive map: a permutation within modalities.

As in all algebras homomorphisms play an important role the calculus of thought [4]. The above transformations constitute configuration homomorphisms.

Also form subgroups of S *over the modalities* $m_1, m_2, ...$

$$S_{m_1, m_2, ...} = S_{m_1} \times S_{m_2} \times ... \tag{11}$$

A set T of thoughts, $T \subset MIND$ is called a *thought pattern* if it is invariant with respect to the modality group S. It is called a *(restricted) thought pattern over the modalities* $m_1, m_2, ...$ if it is invariant with respect to the similarities over these modalities. Thus all modalities are thought patterns but we shall encounter much more complicated patterns in what follow. - Two examples are

[3] See GPT, Chapter 1

[4] See GPT p. 43 and p. 106.

shown in Figure 4.6.1

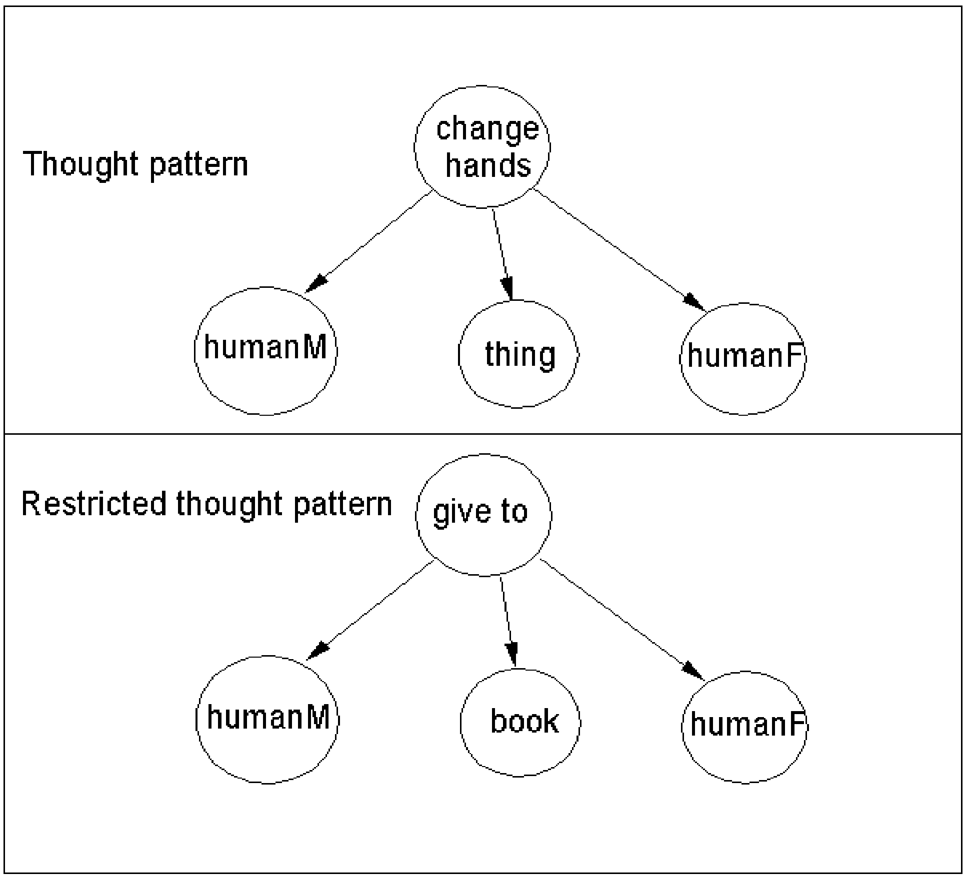


Figure 4.6.1

The set of all thought patterns in MIND will be denoted $\mathcal{P}$. It represents the power of MIND's ability of abstract thinking.

In General Pattern Theory a clear distiction is made between configurations and images [5]. While a configuration specifies generators and connections between them, an image is what can be observed. This is analogous to the distinction between a formula and a function in mathematics. For the

[5] See GPT, Section 2.1 concerning identification rules

elements in the MIND the identification rule $R$for two configurations $c_1 = \sigma_1(g_{11}, g_{21}, ...g_{n1}), c_2 = \sigma_2(g_{12}, g_{22}, ...g_{n2})$is given $c_1 R c_2$ iff there is a permutation $(1, 2, 3, ...n) \leftrightarrow (i_1, i_2, i_3, ...i_n)$ that maps generators and connections from c_1 to c_2. Hence $content(c_1) = content(c_2)$ and the topology of connectors is preserved. In other words, the image is the invariant set under the group of graph automorphisms.

It is known that the graph isomorphism problem is computationally demanding although perhaps not NP-complete. In the present context, however, we are dealing with a more restricted problem where computing may not be overwhelming, see Jean-Loup Faulon (1998).

4.7 Charateristics of an Individual Mind.

In this section we shall limit ourselves to a simple mind, incapable of abstractions and generalizations and not subject to inputs from the external world. In later sections these limitations will be removed.

We have seen the set of all regular thoughts, complete and incomplete, constitute the MIND. It represents all the thoughts that are possible currently, whether likely to occur or not. For a particular individual its MIND may change over time, modifying the generator space G, but momentarily we shall not let the MIND be capable of thinking any new thoughts. That does not mean that all thoughts in the MIND are equally likely to occur. On the contrary, some will occur more often than others: due to external stimuli and remembered events, some are more likely than others. To formalize this we introduce a Q-function taking positive values over the generator space, $Q(g) > 0; g \in G$. A large value of $Q(g)$ means that the elementary idea g is likely and vice versa. The Q-values need not be normalized to probabilities, for example $Q \equiv 1$ is allowed and means no preference for any generator.

So a person overly concerned about his wealth will have large values for $Q(money)$,$Q(stocks)$,$Q(rich)$, $Q(acquire)$..., while someone more concerned about physical appearance will emphasize $Q(looks)$, $Q(Vogue)$, $Q(mascara)$,.... As the situation changes from one genre to another the Q-function will change; more about this in section 4.2.

But the specification of the Q-function does not describe how one simple idea is likely to associate another. To do this we introduce a positive *acceptor function* $A(g_1, g_2)$: a large value of $A(g_1, g_2)$ means that the generators g_1 and g_2 are likely to be associated with each other in the thinking of MIND and vice versa; see GPT, Chapter 7.

Combining the Q and A functions we get a probability measure P with the probability for a regular configuration $c = thought \in MIND = \mathcal{C}(\mathcal{R})$

$$thought = \sigma(g_1, g_2, \ldots g_n) \tag{12}$$

given by the structure formula (see GPT p. 366 for a more general version)

$$p(thought) = \frac{\kappa_n}{n!Z(T)} \prod_{i=1}^{n} Q(g_i) \prod_{(k,k\prime)\in\sigma} A^{1/T}[b_j(g_i), b_{j\prime}(g_{i\prime})] \tag{13}$$

with bonds represented by the coordinates $k = (i, j), k\prime = (i\prime, j\prime)$, edges $(k, k\prime)$ in the connector graph, a temperature T, and a partition function $Z(T)$. Recall that the i's are generator coordinates and the j's bond coordinates. The positive parameter T, the *temperature*, expresses the mobility of the mind: high temperature mean a lively, perhaps unruly mind, and low temperature characterizes a rigid mind. The factor κ_n makes the probability depend upon the size n of the thought, so that a mind capable of abstract thinking has a good deal of probability mass for large values of n.

In order that equation (?) make mathematical sense it is necessary that the κ_n decrease fast enough, preventing infinite thoughts to occur. Precisely how this is done is proven in APPENDIX 1 where the condition takes the form $\kappa_n = O(\rho^n)$ where ρ is less than a certain constant. Later, and in the software for GOLEM, we shall assume that $\kappa_n = \rho^n$.

The bonds take values depending upon what mind modality *mod* a generator belongs to. A generator $g \in mod \subset \mathcal{M}$ with arity ω will have out-bonds $b_j(g); j = 1, 2, \ldots \omega(g)$ and all in-bonds equal to *mod*. Note that the *connector σ in (13) is variable and random* which motivates the appearence of κ_n which controls how likely are thoughts of different complexities; large values in the support of κ means that the mind is capable of complex thoughts. The factor $n!$ means that permutations of the connector graph with its generators would have no real significance. It will sometimes be convenient to work with energies q, a instead of Q- and A-functions

$$Q(g) = exp[-q(g)]; A(b_1, b_2) = exp[-a(b_1, b_2)] \tag{14}$$

Then the energy of a thought can be written as

$$E(thought) = log(n!) - log(\kappa_n) - \sum_{i=1}^{n} q(g_i) - 1/T \sum_{(k,k\prime)\in\sigma} a[b_j(g_i), b_{j\prime}(g_{i\prime})] \tag{15}$$

Here we have left out the term corrseponding to th partitiom function Z; energies are determined up to an additive constant so that we have just normalized the expression for convenience. It has to be reintroduced when we use the relation $E = log(p)$.

If two bonds $k_1 = (i_1, j_1), k_2 = (i_2, j_2)$ have an interaction energy that is positive, $a(k_1, k_2) > 0$, the bond couple is *repellent*, the bonds are unlikely to close. On the other hand if $a)k_1, k_2) < 0$, attractive bonds, the bond couple is more likely to close. Note that open bonds are not automatically likely to close, it depends upon whether the couple is repellent or attractive.

ore precisely we have the following

PROPOSITION, *For a thought $T = (T_1, T_2)$ consisting of two independent thoughts (see Section ?) we have the energy relation*

$$E(T) = E(T_1) + E(T_1) + \binom{n_1+n_2}{n_1} \tag{16}$$

PROOF: We have, using the geometrix series form of κ_n,

$$E(T1) = log(n_1!) - n_1 log(\rho) - Q1 - A1 \tag{17}$$

$$E(T2) = log(n_2!) - n_2 log(\rho) - Q2 - A2 \tag{18}$$

$$E(T) = log(n!) - nlog(\rho) - Q - A \tag{19}$$

where the Q's and A's mean the respected sums in equation (?)and n_1, n_2, n are the sizes of the thoughts T_1, T_2, T. Then

$$E(T) = E(T_1) + E(T_2) + log(n!) - log(n_1!) - log(n_2!) \tag{20}$$

which reduces to the stated result in (?).

Hence, the energy for independent thoughts is additive except for a term $log[B(n_1, n_2)]$, the log of a binomial coeffficient. Since binomial coefficients are always bigger than (or equal to) one, it follows that energy is super-additive. Combining thoughts demand more and more mental power as the sizes increase: the MIND is limited in the complexity of thoughts.

We should think of Q as specifying the constitution of the *mental soup* in which the MIND is immersed at the time. This soup will depend upon the external world that we shall characterize in terms of themes (or genres). This is likely to change during over time. It will also depend upon internal characteristics that may be more persistent, the personality profile, to be treated later,

In order that this definition be mathematically correct an additional condition must be satisfied. This is more technical so that we postpone the discussion to Appendix 2.

The Q and A's determine the *character of an individual mind*: two minds, MIND1 and MIND2, can have the same mental potential, MIND1=MIND2, but different characters, same competence but different performance to borrow Chomsky's terminology.

It should be pointed out that the probability measure defined by the structure formula can be an adequate description of the mental activities only when MIND is at rest, not subject to input from the external world and not conditioned by any fact requiring attention: we are dealing with uncontrolled thinking. Otherwise P must be modified; this will be discussed in depth later. To distinguish the above P from more intelligent ones we shall call it the probability measure of *free associations*.

This defines the configuration space $\mathcal{C}_{complete}(\mathcal{R})$ consisting of all complete thoughts and the configuration space $\mathcal{C}(\mathcal{R})$ that also includes incomplete thoughts.

4.8 Personality Profile

Each MIND has a $self \in G$. The behavior of "self" is regulated by personality parameters *greedy, scholastic, aggressive, selfish,....* The values of the parameters are in the interval (0,1) so that for example "generous" controls the value of $A(self, g)$ with "g" = "give", "lend",... Their values constitute a personality profile that remains fixed after having been se.

The concept of *personality* should be compared to that of "genre" (or "theme") which can vary quickly over time and controls the values of "Q". The genre is not associated with any "self"; it describes the current situation. See Section ? for more details.

4.8.1 An Intelligent Mind?

A mind that deserves to be called intelligent must be able to handle complex ideas, for example the way three simple ideas combine to give rise to a new one. This is related to the celebrated Hammersley-Clifford theorem, see Hammersley-Clifford (1968), which says that on a fixed, finite graph σ with assigned neigborhood relations a probability density p is Markovian iff it takes the form

$$p = exp[-E(c)]; E(c) = \sum_{cliques \subset \sigma} E_{cliques}(g_1, g_2, \ldots g_r) \tag{21}$$

The sum is over the cliques of σ. A clique is a subset of the graph all whose sites are neigbors in the topology of σ. Note, however, that this theorem does not apply without modification to our situation, since the σ's we are dealing with are not fixed but random. Anyway, it gives us a hint on how to organize a more powerful mind.

Instead of using only functions of a single generator, like $Q(g)$, or of two, like $A(g_1, g_2)$, we are led to use energies that depend upon more than two generators. In other words, the mind is controlled by a randomness that involves ideas of higher complexity than size 2. For the specification of P in the previous section we could let the acceptor function depend upon the triple $\{man, love, woman\}$, not just on the pairs $\{man, love\}$ and $\{woman, love\}$.

Having said this, it should be pointed out that this increase in mental complexity could also be achieved by increasing the generator space as described in GPT, section 7.3, that is by forming macrogenerators by combining the original generators. Which of these two procedures we should choose is a matter of convenience and practicality, not of principle: are we most concerned with keeping the cardinality of the generator space manageable or with dealing with small dimensions of energy functions? Whichever alternative we choose, we extend the intellectual power of the synthetic mind.

4.8.2 Randomness and Thinking.

We emphasize that thought processes must include random elements, we do not consider them deterministic. Let us think of a concept like "DOG", perhaps one of the modalities. It is not a well defined scientific entity. "German Shepherd" might belong to it but probably not "wolf". How about "wolf hound"? We are not thinking of the word "dog" but the concept of a dog that we share with others , at least in our own culture. Such *man made concepts are seldom precise*, they always involve some fuzzyness.

This difficulty cannot be avoided, randomness is forced upon us. A purely deterministic, completely rigid, theory of mind is doomed to fail.

5 Mental Dynamics.

The above was dealing with the mind at rest, a static system. Now let us consider the development in time.

5.1 Topologies of Thinking

We need a concept "near" for thoughts: one thought may be close to another thought but not to a third one, and therefore we introduce neigborhoods $N(thought)$, in configuration space by

$$N(thought) = \{\forall thought' \ni thought' \text{ and } thought \text{ differ only in one generator or one connection}\} \tag{22}$$

similar to the discussion in GPT, Section 5.2. This imposes a topology on both $\mathcal{C}_{complete}(\mathcal{R})$ and $\mathcal{C}(\mathcal{R})$, formalizing the concept "thoughts close to each other".

With such topologies it makes sense to talk about *continuity of thought* (although with a discrete interpretation) and *jumps in thinking* , which will be done when discussing the algorithms giving rise to trajectories in mind space. In particular, composite moves, see Section 5.4.

5.2 Trajectories in Mind Space

But how can we compute the probabilities of possible thoughts in $MIND = \mathcal{C}(\mathcal{R})$? In particular, how can we avoid the computation of the infamous partition function? This will be accomplished by a variation of *stochastic relaxation*, see GPT p. 379. The main trick in this celebrated technique is to exploit the Markovian nature of the measure P over mind space (not to be confused with the fact that stochastic relaxation produces a chain that is Markovian over time).

Actually, we need not compute the probabilities of possible thoughts; instead we shall *synthesize* the random mental states by an iterative procedure where each step consists of a *simple move*, or later a composite move, through mind space. This technique is well known to practitioners of MCMC, Monte Carlo Markov Chain [6]. A difference to the usual way one applies MCMC, however,

[6] see GPT, Chapter 7

lies in the fact that for mind representations *the connector is also random*, not just the generators at the sites of a fixed graph. To develop a mental dynamics of the mind we shall think of a trajectory through mindscape, through MIND, as made up of steps, usually small, but occaccionally bigger. Among the simple moves that we have in mind we mention only a few here:

1) Place a generator at a new site; no new connections will be established in this move.

(2) Delete generator in the *thought* and the connections to it. This step automatically respects regularity since the regular structure $MIND = \mathcal{C}(\mathcal{R})$ is monotonic [7].

(3) Delete a connection in σ; also respects regularity (but not complete regularity).

(4) Create a connection between two generators in *thought* if regularity allows this.

(5) Select a generator $g \in thought$ and replace it by another one g' including the possibility of keeping it unchanged, observing the regularity constraint $mod(g) = mod(g')$

All of these moves represent *low level mental activity*, for example the transformations $dog- > dog, big$ and $man- > man, walk$. For each of them we define a random selection rule for choosing among the possible alternatives allowed by the regularity constraints.

REMARK. It should be observed that such simple moves actually map thoughts to *sets* of thoughts when the randomness of the transformation T is taken into account:

$$\mathcal{T} : MIND \rightarrow 2^{MIND} \tag{23}$$

But how do we randomize these choices so that we get the desired probability measure given in (13)?

To do this it is important to select the set $\mathcal{T}$ of moves, $T \in \mathcal{T}$, sufficiently big. More precisely, in order that they generate an *ergodic* Markov chain, which is required for the following argument, it is neccessary for any pair of regular configurations $c_1, c_N \in \mathcal{C}(\mathcal{R})$ that there exist a chain $c_2, c_3, \ldots c_{N-1}$ and $T_1, T_2, \ldots T_{N-1}$ such that

$$c_2 = T_1 c_1, c_3 = T_2 c_2, \ldots c_N = T_{N-1} c_{N-1}; c_i \in \mathcal{C}(\mathcal{R}) \; and \; T_i \in \mathcal{T} \; \forall i \tag{24}$$

In other words: any thought in $MIND$ can be continued to any other thought by a sequence of thoughts, one close to the next one. The chain may be long but finite. This makes the Markov chain (over time) irreducible and since we shall make it have P as an equilibrium measure, it follows [8] that the chain is ergodic. The importance of ergodicity was emphasized in the research program described in the CD-ROM "Windows on the World". It guarantees that the mind is not too rigid so that it is possible to pass from any mental state to any other. We shall assume that this is so in the following.

[7] see GPT, p.6

[8] see Feller (1957), section XV.6

REMARK. On the other hand it may be of some interest to study also situations when the mind is not ergodic, so that it is constrained to a proper subset of MIND. Such a mind just cannot realize transitions between certain thoughts and emotions that would otherwise be consistent with the mental setup, it is abnormally restricted. Therefore the importance of ergodicity is clear. The fact that the Markov chain is irreducible guarantees that *the mind is not too rigid*, so that it is possible to pass from any mental state to another. Otherwise it can be caught thinking constrained to a part of $MIND$, not being possible to exit to other (possible) mind states.

The above applies to fairly short time intervals, say minutes and hours, during which time the MIND has not had time to modify its parameters, G, Q, A substantially. However, for longer durations *the MIND is an open system*, successively modified due to new experiences and input from the surroundings. Also creating new ideas. Then ergodicity does not apply.

On the other hand, when we deal with associations that are not free but dominated by attention to some theme, we shall make the mind almost non-ergodic: the probability of reaching outside a give theme will be close but not equal to zero; see Section 5.5.

As the generators and/or connections are being changed successively we get a trajectory in mind space

$$thought_1 \rightarrow thought_2 \rightarrow thought_3 \ldots \tag{25}$$

which represents a *a train of thoughts*, some conscious, others not, a trajectory through mental domain $MIND$. The intermediate thoughts play the role of the links in Poincare's chain of thought.

5.3 Dynamics with Simple Moves

Let us still deal with a situation when no external stimuli impact on the mind and where the time duration is so short that we can neglect changes in the mind energies q and a.

Let us explain the way we make use of the Markovian nature of P. Say that we are dealing with a transformation $T : MIND \rightarrow MIND$ that only affects a single generator g_i at site $i \in \sigma$, see Figure 5.3.1

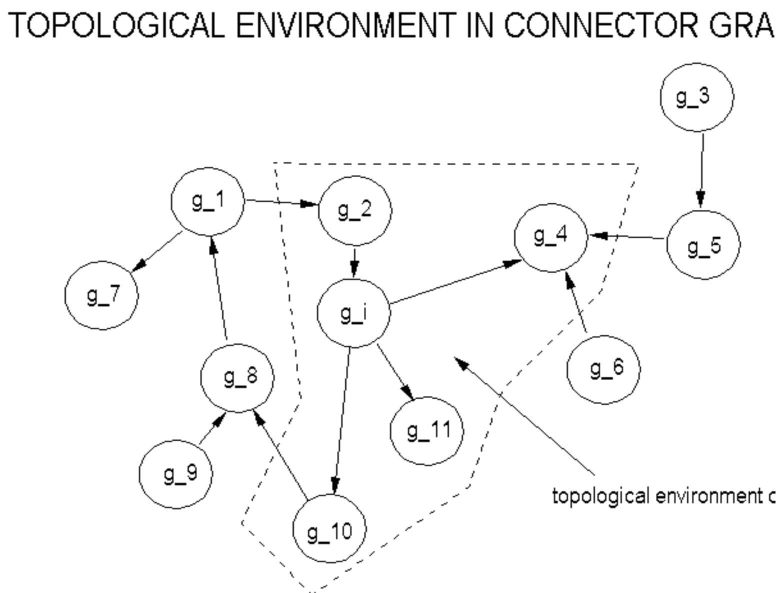


Figure 5.3.1

The site i has the neighbors $2, 4, 10, 11$ so that we can write the conditional probability

$$P(g_i|g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}) =$$

$$= \frac{P(g_i, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11})}{P(g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11})}$$

But we can use (13) to reduce this expression by cancelling common factors in numerator and denominator, leading to

$$P(g_i|g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}) =$$

$$= \frac{P(g_i, g_2, g_4, g_{10}, g_{11})}{P(g_2, g_4, g_{10}, g_{11})}$$

This simplification is not very useful for thoughts consisting of just a few generators, but if the number, n, is large, it amounts to a considerable gain in computing effort.

In this way we can express the conditional probabilities we need for stochastic relaxation in the form

$$P(A|B) = \frac{N}{D} \tag{26}$$

where N and D are joint probabilties of sets in $\mathcal{C}(\mathcal{R})$ of moderate dimension. This reasoning was for simple moves involving only changes of generators while leaving the connector σ unchanged. If the connections in the connector also can change, they have to be included among the variables that make up the sample space of the relaxation procedure. Then the topology induced by the neighborhood relations has to be adjusted in the obvious way, but the general procedure remains the same as just described.

We choose a set of configuration transformations $\mathcal{T} = \{T^1, T^2, \dots T^\nu\}$, for example $\mathcal{T} = \{(2), (5)\}$, see last section. It is large enough to span MIND,and we shall now construct updating algorithms for each T^l [9]. Apply the transformation $T = (2)$, with deletion at site m or no deletion at all with given probabilities, to the configuration $thought_{old}$ resulting in $thought_{new} = Tthought_{old}$. We need the probability for the new mental state which, using (13), is propotional to N/D with the numerator

$$N = \pi_{n-1} \prod_{i=1, i\neq m}^{n} Q(g_i) \prod_{(k,k\prime)\in\sigma^m} A^{1/T}[b_j(g_i), b_{j\prime}(g_{i\prime})] \tag{27}$$

where σ^m is the graph obtained from σ of *thought* by deleting the site m as well as bonds emanating from it. Similarly, the denominator is

$$D = \pi_n \prod_{i=1}^{n} Q(g_i) \prod_{(k,k\prime)\in\sigma} A^{1/T}[b_j(g_i), b_{j\prime}(g_{i\prime})] \tag{28}$$

This gives us

$$N/D = \frac{\pi_{n-1}}{\pi_n Q(g_m) \prod_{(k,k\prime)\in\sigma^-} A^{1/T}[b_j(g_i), b_{j\prime}(g_{i\prime})]} \tag{29}$$

where σ^- means the graph consisting of the site m together with the bonds emanating from it. This we do for $i = 1, 2, \dots n$ as well as for no deletion in which case (23) should be replaced by $N/D = 1$.

REMARK. If global regularity requires that deletion of a site also requires the deletion of other sites and their bonds, then (23) has to be modified accordingly.

[9] see e.g. GPT, section 7.6

Now $T = 5$. For an arbitrary generator $g \in G$ we need the equivalent of (23) placing g at a site with modality $mod(g)$ or not introducing any new generator at all, so that

$$N/D = \frac{\pi_{n+1}\pi_n Q(g)\prod_{(k,k\prime)\in\sigma^+} A^{1/T}[b_j(g), b_{j\prime}(g)]}{\pi_n} \tag{30}$$

where σ^+ is the graph consisting of the new generator g and bonds emanating from it. Note that in general there are several ways of connecting g to the old configuration and (24) must be evaluated for all these possibilities. For the case of no change, the right hand side of (24) should be replaced by 1.

The stochastic relaxation then proceeds by iteration as follows.

step $T = 2$: Compute the expression in (23) for $m = 1, 2, \ldots n$, normalize them to probabilities and simulate deletion at site m or no deletion. Get the new *thought*.

step $T = 5$: Compute the expression in (24) for this T, normalize and simulate. Get the new *thought*.

step $T = 2$:....

and continue until sufficient relaxation is believed to have been obtained. As in all applications of stochastic relaxation it is difficult to give precise advice about when this has been achieved. Trial and error may have to suffice.

The above development of thoughts, the *thought chatter*, is thus essentially random. Of course not purely random but controlled by the regularity constraint as well as by the mental parameters Q, A. This is reminiscent of chemical reactions: many reactions (thought developments) are possible, but only a few actually take place. For example the thought ($green- > cat, grass$) is regular but has low probability. A reaction would probably result in the thought ($cat, green- > grass$) which has higher probability, lower energy and would stay conscious for a while. The first, unlikely one, will only be glimpsed consciously, if at all, and be hidden in the thought chatter.

5.4 Mental Dynamics with Composite Moves

With the above set up only changes at a single site or at a single connection are made at each instance of a train of thought; the mental associations are simple in the sense that only short steps are taken in the mental trajectory space. The change in mind state only depends upon the neighboring states of mind . But we shall also allow *composite moves* where the changes involve larger sub-thoughts. We do not have in mind a strict cause and effect relation; we want to avoid determinism, so that we will continue to allow the changes to be random. The reason why we allow composite moves is not that it will speed up convergence to the equilibrium measure, which is the standard motivation behind similar devices in most applications of stochastic relaxation. Such speed up may indeed occur, but that is not our motivation. Instead we believe that the train of thought obtrained by composite moves mirrors more closely what goes on in real thought processes. Of course we have no empirical evidence for this, only introspective observations.

REMARK. The version of stochastic relaxation we have used here is only one of many, actually the most primitive. In the literature several others can be found that are guaranteed to have faster convergence properties, but as mentioned, we are not concerned with speed here. Or are we ? If our conjecture that thinking can proceed in large jumps is correct, it may be that this happens in order to speed up the thought process, omitting links in it that are known to the mind to be at least plausible. Worth thinking about!

Now let us mention some examples of composite moves. In Figure 5.4.1

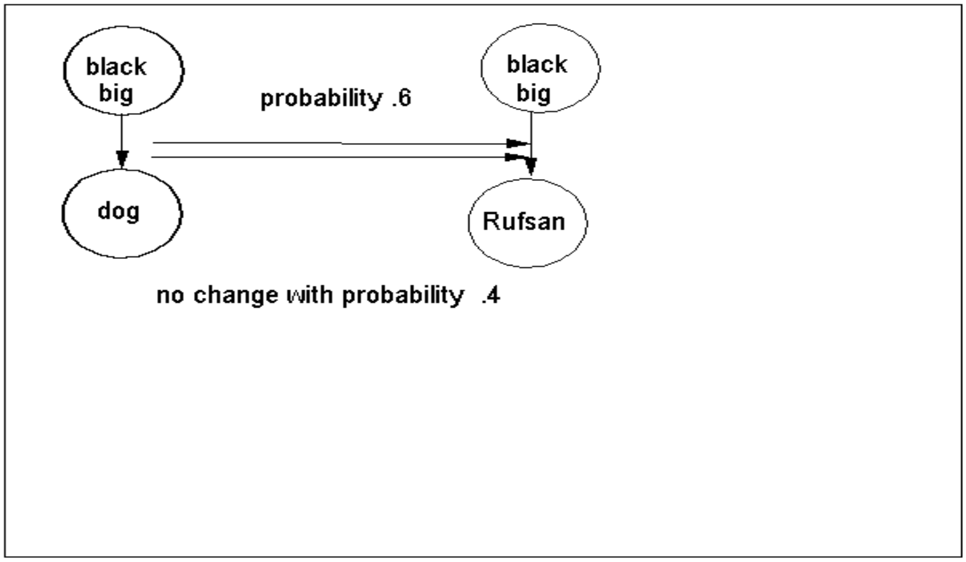


Figure 5.4.1

The thought "dog black big" is transformed into "black big Rufsan" with probability .6, expressing the knowledge possessed by this mind that if a dog is black,

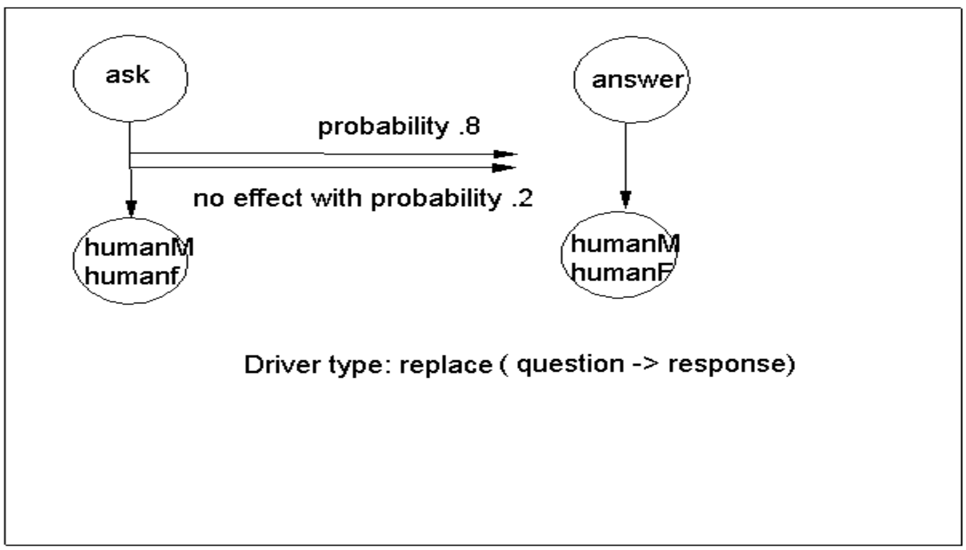


Figure 5.4.2
it is most likely to be Rufsan, at least in some MIND. Or, in Figure 5.4.2,

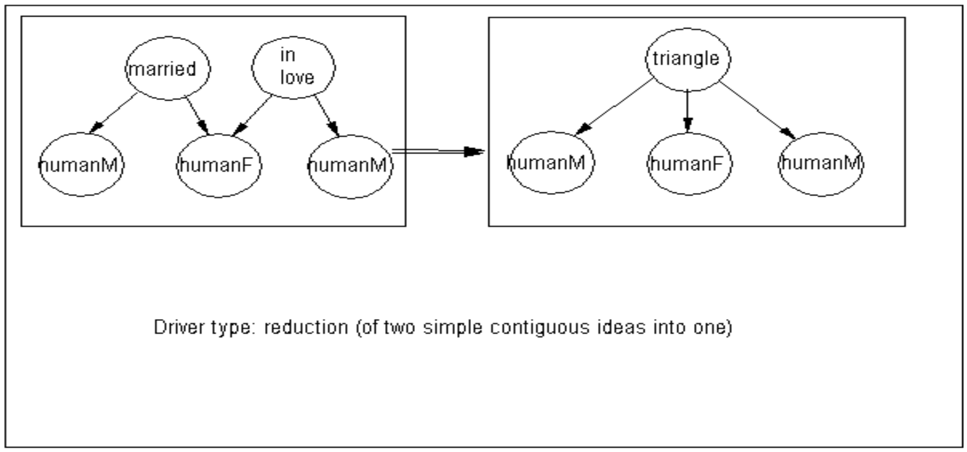


Figure 5.4.3

which desrcribes how a thought with the five generators "humanM,humanF,humanM,married,in love" is transformed into the eternal triangle. In Figure 5.4.4 we see how hungry humans or animals will become satisfied after eating.

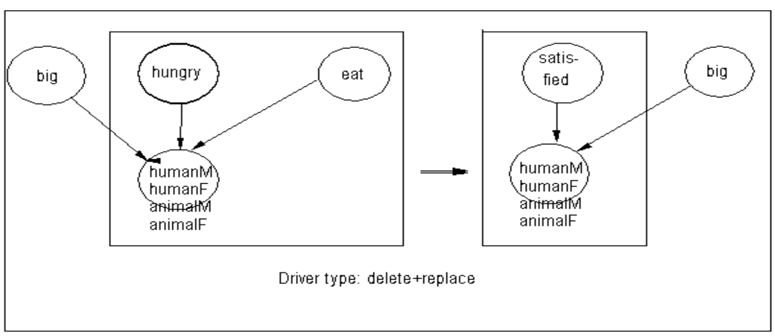


Figure 5.4.4
Some familiar drives are shown in Figures 5.4.5-7

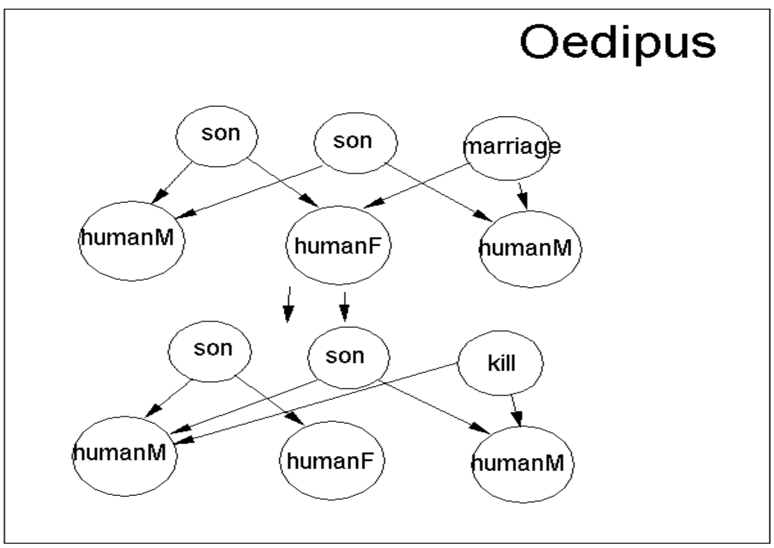


Figure 5.4.5
the Oedipus complex,

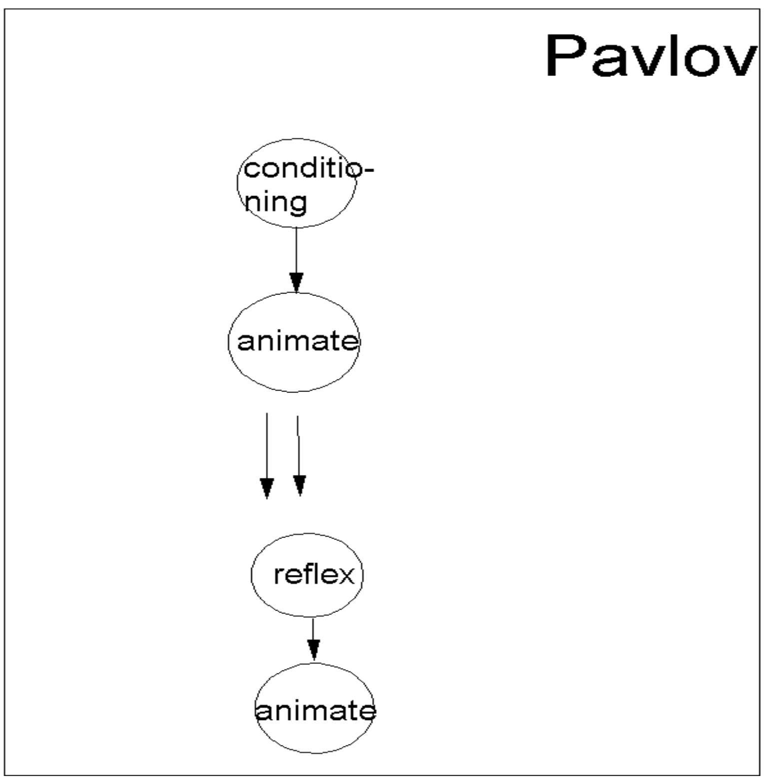


Figure 5.4.6
Pavlov's dog. Also Adler's self asserting individual.

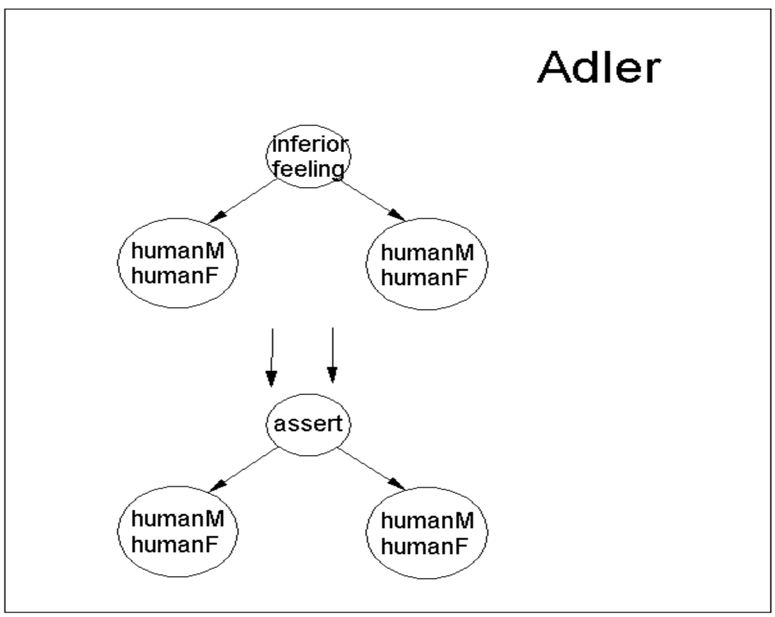


Figure 5.4.7

The general form of a composite move is a transformation whose domain and range are sets of regular thoughts

$$Move : THOUGHT1 \rightarrow THOUGHT2; THOUGHT1, THOUGHT2 \subset MIND \quad (31)$$

together with a probability measure $P_{move}, move \in MOVE$ over the set $THOUGHT1$. The measure P_{move} may be specified in the same way as for the simple moves, although their calculation will be more involved but it can also be modified to account for preferences of thinking. In this way the composite moves contribute to convergence to the equilibriunm measure P just as the simple moves do, but the *trajectories will be different*, the steps $thought(t) \rightarrow thought(t+1)$ will be different, hopefully more realistic in characterizing the fuctioning of a particular mind. This applies to free associations. However, for less passive thinking the probabilities applied to composite moves may be different, influenced by attention to genres as will be discussed in the next section.

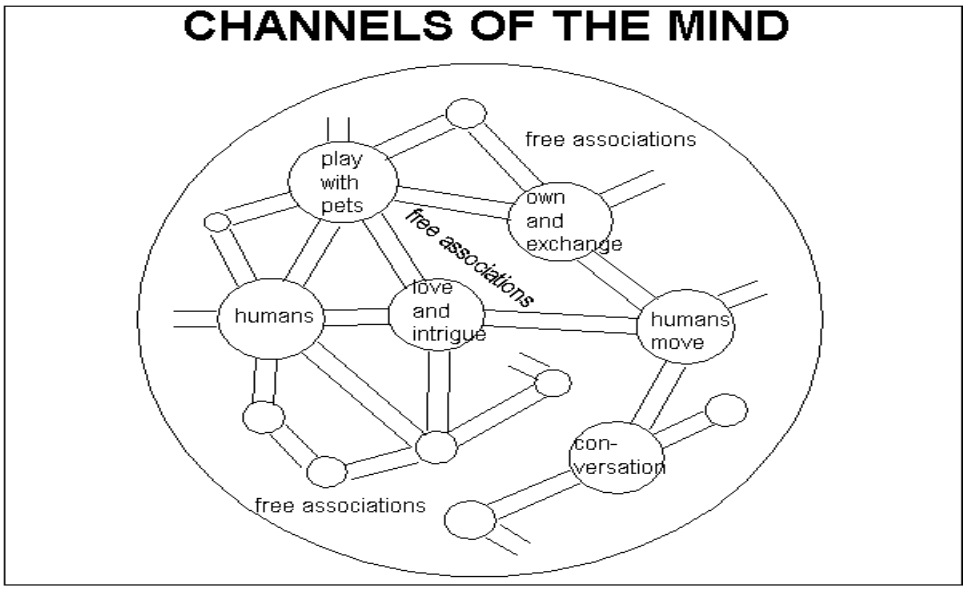
Note that we have implicitly allowed composite moves to apply to patterns of thoughts, not just to single thoughts.

We believe that a realistic mind represention will require many types of composite moves for the mind dynamics in contrast to static mind representation.

5.5 Mental Dynamics with Themes of Attention: Genres

Up till now we have operated with a fixed equilibrium measure, P, but what happens when the mental genre changes? For example, when the domain of discourse concerns wealth and distribution of wealth. Or when the emphasis is on the emotional relation to another individual. To deal with such situations we shall let the Q-vector change, say by increasing the values of $Q(money)$, $Q(acquire)$, $Q(buy)$,$Q(sell), \ldots$ or $Q(love), Q(jealousy), Q(sweetheart), \ldots$, so that the mind visits these generators and their combinations more often than for free associations. Then the discourse is weighted toward a specific *genre* with a lower degree of ergodicity since it will take time to exit from these favored thoughts.

In this way we allow $Q = Q(t)$ to change in steps when one genre is replaced by another. We illustrate it in Figure 5.5.1; the circles represent constant Q and arrows indicate steps between mental genres. Different genres are connected via channels through which the mind passes during the thinking trajectory.



Figures 5.5.1

More formally, introduce genres $\Gamma_r \subset G$ not neccesarily disjoint, in terms of a-energies, and the *mental forces* F_r as the gradient vectors of dimension $|\Gamma_r|$ of the energies

$$F_r = (...f_\mu...); f_\mu = -\frac{\partial q}{\partial g_\mu}; g_\mu \in \Gamma_r \tag{32}$$

This corresponds vaguely to the usage of "force" and "energy" (potential) in rational mechanics. This means that a force acts in the mind space to *drive* the mind into respective genres; it influences attention.

5.6 Mental Dynamics of Dreaming

To represent mind trajectories coresponding to dreaming and less conscious thought processes we shall make the binding between elementary thoughts less stringent, as dreams tend to allow strange and unusual transitions and associations. The technical way that we have chosen to do this is by increasing the temperature T appearing in the structure formula (13). A higher value for the temperature lowers the value of the factor $A^{1/T}[b_j(g_i), b_{j'}(g_{i'})]$ so that the elementary thoughts, the generators, become less stochastically dependent. In other word, the thinking becomes less systematic, more chaotic.

6 A Calculus of Thinking

The MIND calculates. Not as a deterministic computer with strict algorithmic rules, but with a certain amount of controlled randomness. Among its algebraic operations, the *mental operations*, we mention especially two (more to follow):

$\boxed{\textbf{mop 1} \quad = \text{SIMILAR: } \textit{thought} \mapsto s \; \textit{thought}}$

as illustrated in Figure 5.6.1a

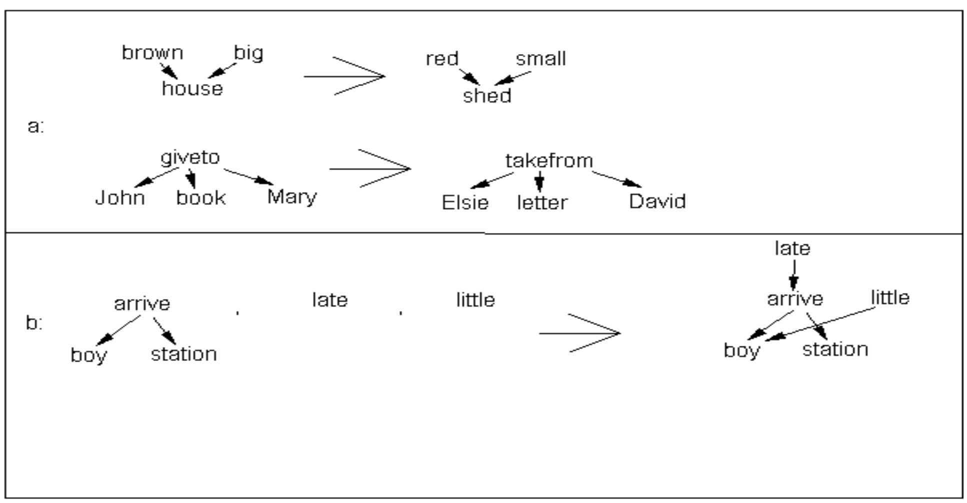


Figure 5.6.1

and
$\boxed{\textbf{mop 2} \;\; = \text{COMPOSE: } thought1, thought2 \mapsto \sigma(thought1, thought2)}$

with some connector σ as illustrated in Figure ???b. We say that $thought1$ contains the $thought2$ if there exista a $thought3$ such that $thought1 = COMPOSE(thought2, thought3)$.

Hence **mop1** changes a thought to one belonging to the sam thought pattern (see Section ?), replacing elementary ideas with similar ones. The **mop2** combines two thoughts into a single one.

Note that this algebra is partial in that compositions of thoughts are only allowed if bondvalues agree in the coupling of the connector σ. The mental operations are formalizations of intuitive concepts of thinking processes. Approximate since the intuitive concepts are vague and not precisely defined. As all mathematical formalizations they increase precision but decrease generality.

With this architectonic approach, *pace* Kant, to the study of the mind, the most fundamental mental states, the elementary ideas, combine to make up the trajectories through the mind space $MIND$, governed by entities like Q, A, drives and so on. Certain regular sub-thoughts can be singled out because of their particular role. But how do we combine and operarate on composite thoughts, how do we hook them together in Poincare's parlance? To do this we shall first consider some special instances.

6.1 Specific Thoughts

6.1.1 Conscious Thoughts

As the trajectory develops many associations are formed, most probably irrelevant. At a particular time t the total mind state $thought = thought(t)$ can be decomposed into connected components w.r.t. the topology induced by the total connector σ. In order that any connected component $subthought \subset thought$ be active enough to reach consciousness will be assumed to depend upon its mental size. We formalize this through the

DEFINITION. *A conscious thought is a maximal component of the current mind state*

6.1.2 Top-thoughts

Another type of (sub)-thought is based on the notion of *top generator*

DEFINITION: *A top-thought in a total thought means a sub-thought (not necessarily a proper subset) that starts from a single generator and contains all its generators under it with respect to the partial order induced by σ. Its level is the level of its top generator. A maximal top-thought has a top generator that is not subordinated to any other generator in thought.*

Let $tops(thought)$ denote the set of all generators in a $thought$ that are not subordinated any other generators. Then we get the decomposition

$$thought = top\_thought(g_1) \oplus top\_thought(g_2) \oplus top - thought(g_3) \ldots ; g_k \in tops \quad (33)$$

where $top\_thought(g)$ stands for the sub-thought extending down fom g. Note that in (27) the terms may ovelap, two top- thoughts may have one or more generators in common as shown in Figure 6.2.1.1

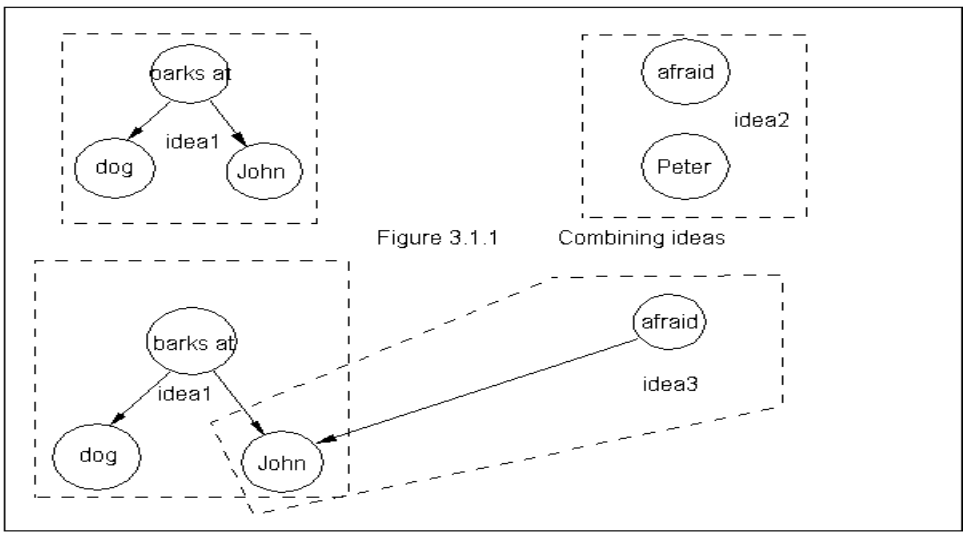


Figure 6.2.1.1

where the two top-thoughts $idea1$ and $idea3$ in the lower part of the figure have the generator $John$ in common but the top-thoughts above in Figure 6.2.2.1 do not: the latter can be said to be *regularly independent*: they are indeed independent as far as their meaning is concerned.

Inversely, given two regular thoughts $thought1$ and $thought2$, we can form the composition

$$thought_{new} = thought_1 \overset{\sigma}{\oplus} thought_2 \tag{34}$$

where we have indicated by $\overset{\sigma}{\oplus}$ what generators, if any, $thought_1$ and $thought_2$

have in common; it can have the form

$$\sigma = \begin{cases} g_{1i_1} = g_{2k_1} \\ g_{2i_2} = g_{2k_2} \\ \dots \end{cases} \tag{35}$$

If *thought* is a top-thought, consider its external bonds

$$ext(thought) = ext_{up}(thought) \cup ext_{down}(thought) \tag{36}$$

consisting of up-bonds and down-bonds; note that all internal (i.e. closed) bonds are excluded. [10].

In section 9, when we start to build a mind, we shall be especially concerned with top-thoughts of level 2 although in general its level can be higher. This will lead to a mind that may be said to be intellectually challenged since its mental combination power is very restricted. We make this assumption only for simplicity; it ought to be removed.

6.2 Generalization/Specialization Operation.

The process of generalization will here be understood in terms of the operator MOD that is first defined on $G \cup \mathcal{M}$ and takes a g into $mod(g)$ and a modality m into itself. In the following it will be assumed that the modality taxonomy is of Linnean form so that MOD is one-valued (it would however be of interest to consider the case of non-Linnean taxonomy in which case the generalization operator can be many-valued). It is then extended in the natural way to $\mathcal{C}(\mathcal{R})$. The operator MOD is distributive over composition, so that $MOD(thought)$ is defined for $thought \in MIND$.

For example,

$$MOD(bark \downarrow Rufus) = (animal\_sound \downarrow animalM) \tag{37}$$

or

$$MOD(color \downarrow house) = (color \downarrow building) \tag{38}$$

The operator MOD *extends the meaning of a thought by suppressing incidental information and hence deserves to be called generalization.* Hence the mind calculus also has access to the operation

mop 3 =GENERALIZATION: MOD TRANSFORM OF THOUGHT

It should be mentioned that the MOD operation can be iterated. For example, we can get the successive generalizations $Rufsan \rightarrow DOG \rightarrow ANIMAL\_canine \rightarrow ANIMAL \rightarrow ANIMATE$. What *generalization is useful* depends upon how often the thoughts contained in it will occur together.

[10] for a discussion of these concepts see GPT, Chapter 1

But this deserves some comments. We have allowed modalities to join in a limited way, combining parts of their contents that have common out-bonds. Thus it makes sense to iterate the generalization operation $\mathcal{G}$, resulting in a *semi-group* $\mathcal{G}^{power}; power \in \mathbf{N}$. Actually, some reservation is needed here to get tree (or forest) structure.In the MATLAB code for GOLEM only Linnean modality structure is allowed. Anyway, this makes it possible to form generalization of *thought* of the first order, $power = 1$, of the second order,$power = 2$, and so on.

The *specialization operation* does the opposite to generalization. In a $thought = \sigma(g_1, g_2, ...g_n)$ one of the g_i is replaced by $MOD(g_i)$. For example:

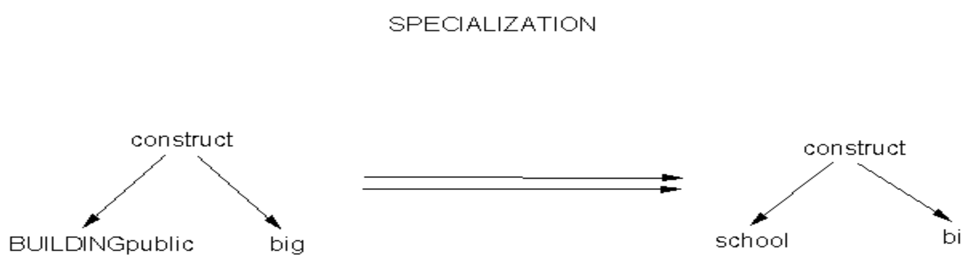


Figure 6.2.1.1a

6.3 Abstraction Operation

Considet a $thought \in MIN)$ with the top generator g_{top} on level l and $mod(g_{top}) = m$ and external down-bonds

$$ext_{down}(thought) = (m_1, \dots m_\omega) \tag{39}$$

If this *thought* occurs more than occasionally the mind may create a new generator, a macro-generator, g_{macro} with the same interpretation as *thought* on level 1, up-bond $IDEA$. This *encapsulation procedure formalizes the mental process of abstraction* and will be spoken of as *ideafication*. Due to it the generator space has increased: the MIND can handle the idea as a unit with no internal structure.

For example

$$thought = (married \downarrow humanMand \downarrow humanF) \tag{40}$$

is abstracted to the macro-generator $g = marriage$ on level 1 with modality $IDEA$. Continuing the abstraction process we can introduce still another macro-generator *divorce* by abstracting the

$$thought = (dissolve \downarrow marriage) \tag{41}$$

as *divorce* of modality $IDEA$. Hence the calculus also includes the operation

mop 4 =ABSTRACTION = ENCAPSULATED THOUGHT

Then we can consider a new thought as a unit [11], a generator in the modality $IDEA$. This means a transformation

$$ENCAPSULATION : thought \rightarrow idea_k \in IDEA \subset G \tag{42}$$

We shall use many such generators in a modality called IDEA, often linked to generators like "say", "ask", "think". The transformation *ENCAPSULATION* plays an important role when representing mental states involving information transfer, for example

$$ENCAPSULATION : say \mapsto (black \downarrow Rufsan) \tag{43}$$

with the right hand side as a generator in $IDEA$ connected to *say*.

It should be mentioned that encapsulation can lead to configurations involving encapsulation again, *nested structures that allow the self thinking about itself* and so on. An iterated encapsulation *idea* will be said to have $power(idea) = p$ if it contains p iterations. Once it is incorporated as a unit in G its power is reset to zero. This will have consequences for the updating of the memory parameters

[11] This has been suggested in GPT, section 7.3

Q, A. More particularly, a new *idea* of size n, *content* $= (g_1, g_2, g_3, ...g_n)$ and connector σ will be given a Q-value

$$Q(idea) = \frac{\kappa_n}{n!} \prod_{i=1}^{n} Q(g_i) \prod_{(k,k\prime)\in\sigma} A^{1/T}[b_j(g_i), b_{j\prime}(g_{i\prime})] \tag{44}$$

and A-values equal to one for those connections that are indicated by the modality transfer function and equal to a small positive number otherwise.

6.4 Completion Operation.

If *thought* has some of its out-bonds left unconnected it will not be meaningful, it is incomplete. It can be made complete by adding elementary ideas so that all out-bond become connected. This multi-valued operation is called COMPLETE, and in the software it is named DEEP THOUGHT since it may require the MIND to search deeply for legal and hence meaningful extensions of *thought*. Or, symbolically,

$\boxed{\textbf{mop 5} = \text{COMPLETE} = \text{DEEP THOUGHT}}$

6.5 Genre Operation.

On the other hand, we can also change the probabilistic parameters that determine the behavior of MIND. Thus we have the GENRE operation

$\boxed{\textbf{mop 6} = \text{genre: } Q \to Q_{genre}; genre \in GENRE}$

6.6 Inference Process

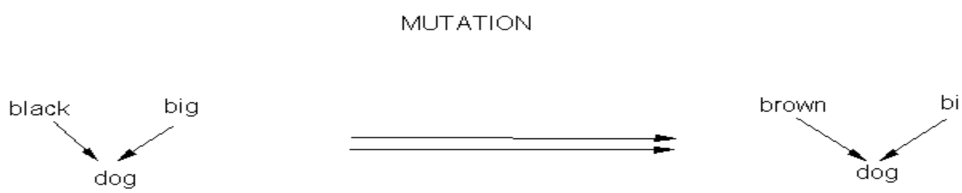
Given the *thought* we can ask what the mind infers from it. This is done by another mental and random operation

$\boxed{\textbf{mop 7} = \text{INFER: } thought \to thought_{infer}}$

where $thought_{infer}$ is a random element sampled from in MIND according to the conditional probability relative to P that the element contains *thought*. Actually, we use the term ”inference” in a wider sense than what is standard. Usually ”inference” means the process by which we try to interpret data in terms of a theoretical super-struture, perhaps using statistical methods. We shall, however, mean the mental process by which we extend a given thought, we *continue* it according to the probability measure P. Thus it is a random and multi-valued process.

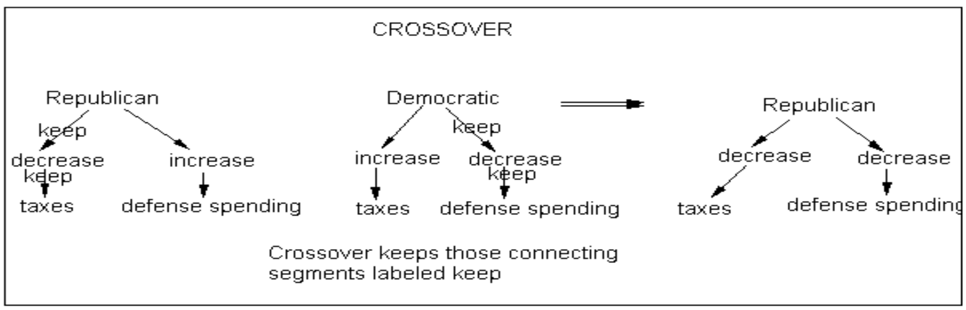
From a given *thought* we can then infer a bigger one that naturally extends $thought- > thought'$. For example, if $A(Rufsan, black)$ is big, we may get the inference $Rufsan- > Rufsan, black$. This will happen if the MIND has seen the sub-thought $Rufsan, black$ many times so that the memory updating (see section 7.3) has taken effect. On the other hand, we may not get the inference $black- > black, Rufsan$, since it is unlikely that the MIND will select that inference from *black* from many others as likely. This lack of symmetry seems natural for human thought.

$\boxed{\textbf{mop 8} = \text{MUTATE: } thought \rightarrow thought_{mutated}}$ The mutation operation in it simplest form changes a generator g_i in $thought = \sigma(g_1, g_2, ...g_n)$ into another g'_i belonging to the same modality, for example:



However, a more general form of mutation would allow a small and random number of simple moves to be applied to the thought.

$\boxed{\textbf{mop 9} = \text{CROSSOVER: } thought1, though2 \rightarrow thought_{crossover}}$ This operation starts with two thoughts $thought1 = \sigma_1(g_{11}, g_{12}, ...g_{1n_1}), thought2 = \sigma_2(g_{21}, g_{22}, ...g_{2n_2})$ and forms a new connector by combining a sub-connector $\sigma'_1 \subset \sigma_1$ with a sub-connector $\sigma'_2 \subset \sigma_2$. Keep generators as they are and form a new *thought* with the connector $\sigma'_1 \cup \sigma'_2$. For example:



The reader will have noticed that we treat thoughts more or less as biological organisms. The crossover operation, in particular, is similar to what occurs in genetics..

$\boxed{\textbf{mop 10}\text{= PERSONALIY CHANGE: } A \to A_{personality}}$ makes changes in the values of $A(self, \cdot)$ so that the MIND's behavior changes probabilistically.

7 Birth and Death of Thoughts

We certainly do not think of a mind as a static system, instead it will develop in time. As already mentioned, under free associations ideas and fragments of ideas will appear according to a probability measure $P = P(t)$ changing with time t but only slowly with time scales as minutes and days rather than seconds and milliseconds. In this view we could claim that what we are constructing is a theory of the *artificial life of thoughts*.

7.1 Competiton among Unconscious Thoughts

Say that the current configuration $thought \in \mathcal{C}(\mathcal{R})$ has been decomposed into the top-thoughts

$$thought = top\_thought(g_1) \oplus top\_thought(g_2) \oplus top\_thought(g_3) \ldots ; g_p \in tops \quad (45)$$

as in section 3.1. Let us calculate the energies

$$E[top\_thought(g_1)] = -log[P\{top\_thought(g_k)\}]; k = 1, 2, ...p \quad (46)$$

determined by the current probability measure and its associated energetics $q(\cdot), a(\cdot, \cdot)$. Hence an energy can be found as

$$E_p = \sum_{i \in \sigma_p} q(g_i) + \sum_{(k,k\prime \in \sigma_p} a(g_i, g_{i\prime}; k = (i, j); k\prime = (i\prime, j\prime) \quad (47)$$

In this random collection of sub-thoughts *they compete with each other for existence on a conscious level*. This may remind a reader of the role of the censor mechanism in Freudian psychoanalysis, but that is not our intention. Instead we consider the thinking process as a *struggle between unconscious thoughts in a thought chatter*. The competition is decided in terms of their energies, but it is not a deterministic decision process. Instead, their associated probabilities

$$\pi_p = exp[-E_p/T] \quad (48)$$

control the choice of the winning one, so that, on the average, low energies are favored.

For a hot mind, $T >> 1$, the mind is a boiling cauldron of competing chaotic thoughts in the unconscious. Eventually the mind crystallizes into connected thoughts, reaching the level of conscious thought. For lower temperature the competing thought are less chaotic and the mind will settle down faster.

7.2 Evolution of the Mind

As time goes on the mind is evolving as the effect of the ideas that have been created and others forgotten. The *long term memory* is represented by the Q and A functions as well as by the evolving generator space G. If a generator g has occurred the effect will be assumed to be updated as

$$Q(g) \rightarrow remember_Q \times Q(g); remember_Q > 1 \quad (49)$$

where the constant $remember_Q$ expresses the strenghtening of memory concerning g. Each time that g does not occur the effect is

$$Q(g) \to forget_Q \times Q(g); forget_Q < 1 \tag{50}$$

with another constant $forget_Q$ for the loss of memory. The acceptor function is modified in a similar way.

Hence we have the MEMORY operation

$$MEMORY : (Q, A) \mapsto (Q_{modified}, A_{modified}); \tag{51}$$

When a new thought *idea* is added to G its Q-value is set proportional to the power $2^{iter(idea)}$ initially and will of course be modified later on due to new experiences and thinking.

It will sometimes happen that some newly created ideas coincide. To avoid misuse of memory we shall remove the copies. Actually, we shall do this as soon as the *content*'s are the same whether the *connector*'s are the same as not. This is done for no other reason than to reduce thinking effort by comparing graphs. Two ideas $idea1$ and $idea2$ will be considered different iff $content(idea1) \neq content(idea2)$. Periodically the memory will be updated by replacing two or more equal ideas by a single one: $\{idea1, idea2, ...idea_k\} \to idea1$, removing its copies and setting $Q(idea1) = \sum_1^\nu Q(idea_\nu)$.

In other words, the ideas behave as organisms: they get born, the grow, compete and change, they die, and the population of ideas in G evolves over time. The mind has a life of its own.

But what happens if GOLEM is not exposed to any inputs, it just lives an isolated life? The Proposition in APPENDIX 5 answers this question. The MIND will degenerate and more and more limit itself to a small subset of elementary ideas, namely those that were favored by the Q-vector at the very beginning of isolation.

8 Some Thoughts in Goethe.

Let us now illustrate the construction by some thoughts appearing in a famous novel by Goethe, "Die Wahlverwandtschaften" (Elective Affinities). This choice is especially appropriate since, when Goethe wrote his work, he was strongly influenced by then current theories of chemistry based on affinities between substances, similar to the bonds between ideas that we have postulated for human thought processes. We shall only look at some simple thoughts and hope that some researcher will pursue this attempt more fully.

A simple example is

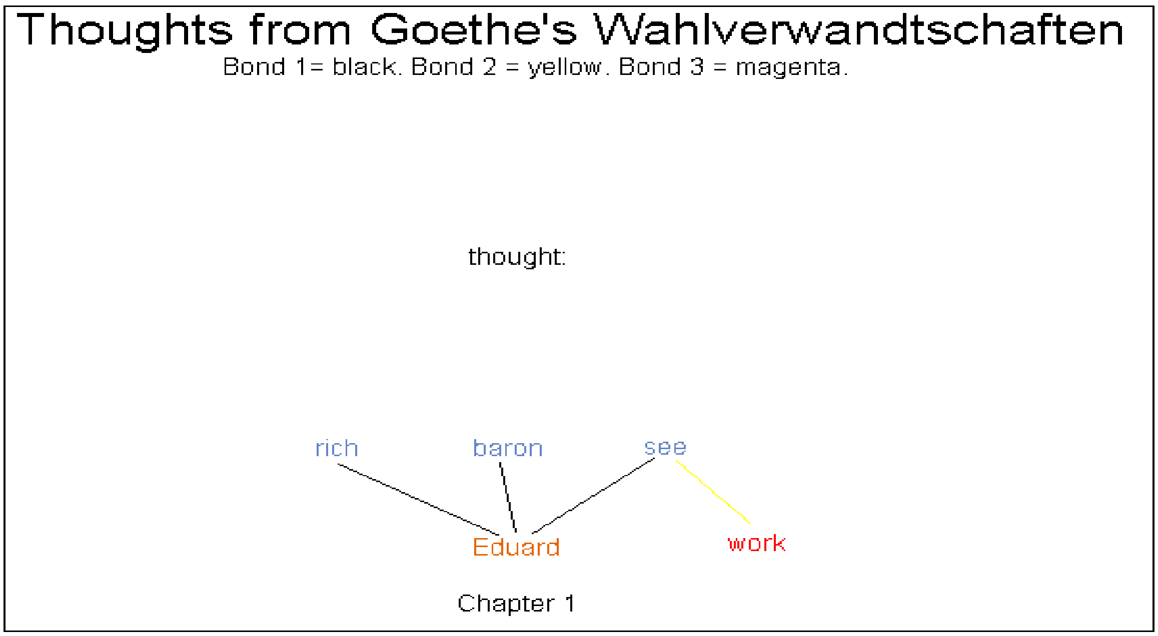


Figure 8.1. Interpretation:"the rich baron Eduard sees work"
and another simple one

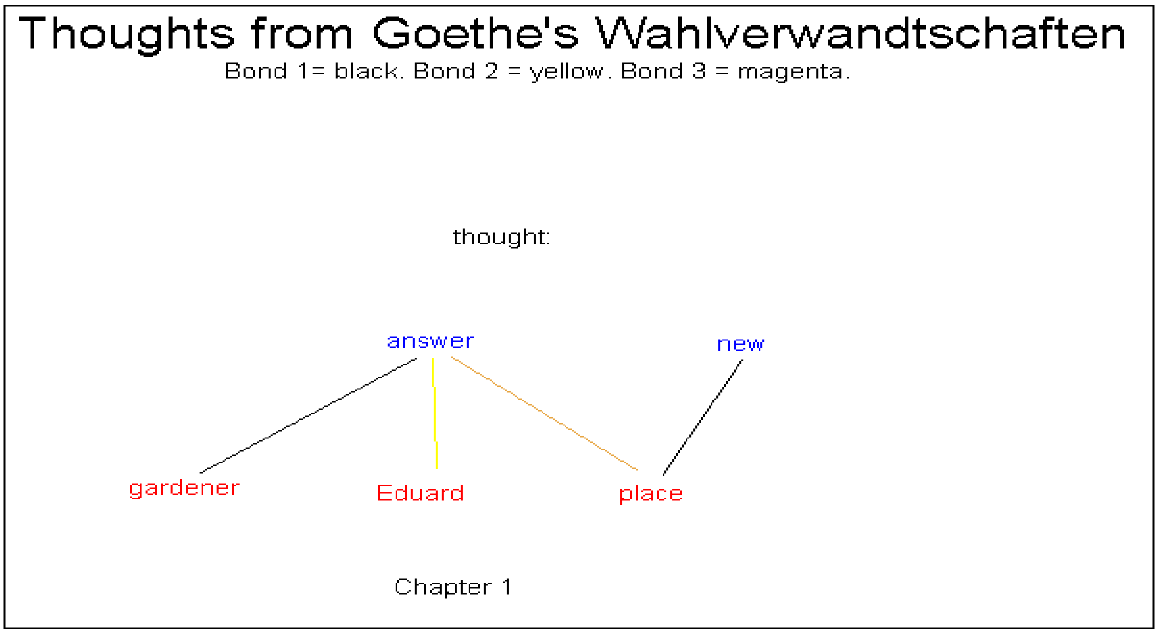


Figure 8.2. Interpretation:"the gardener answers Eduard that the place is new"

The next one involves encapsulation of an idea

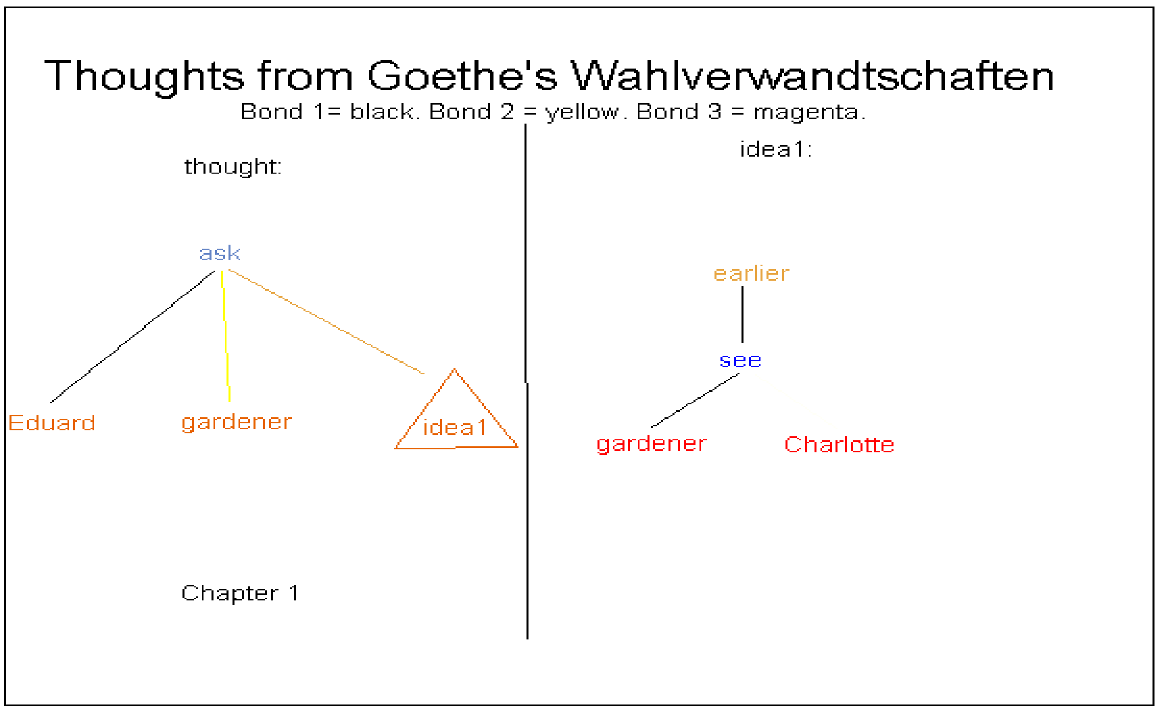


Figure 8.3. Interpretation:"Eduard asks the gardener somthing", something=" gardener has seen (someon) earlier"

Note that *idea*1 is not a complete thought. Recurrent thought with nested encapsulation is seen in Figure 8.4

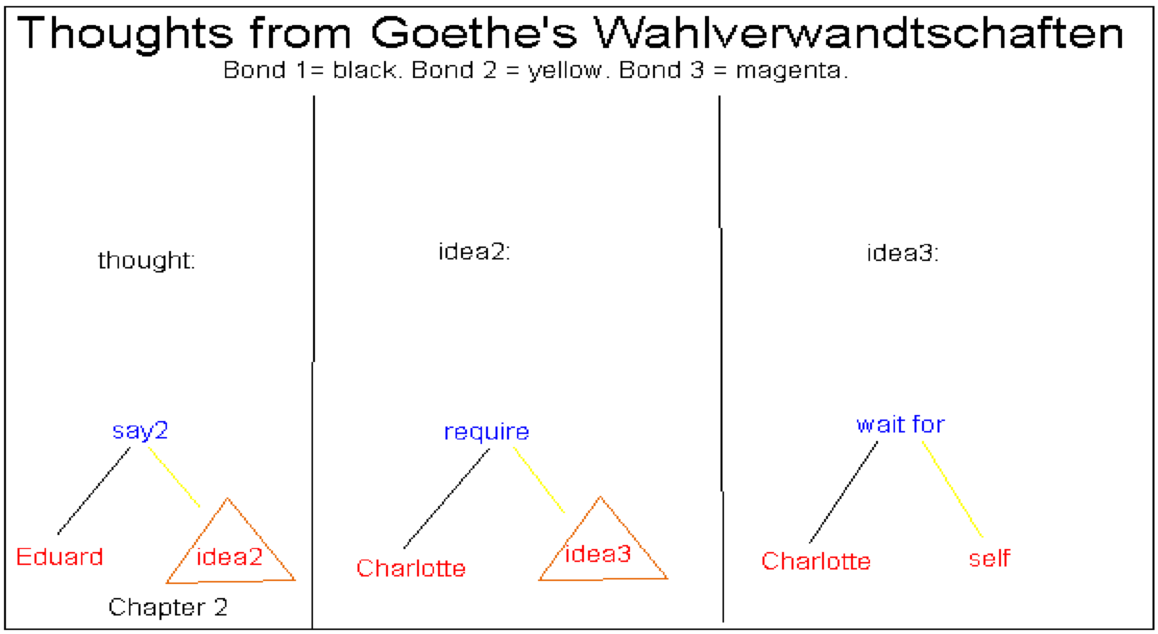


Figure 8.4. Interpretation: "Eduard says that Charlotte requires that she waits for him"

The next three figures show slightly more complicated thoughts.

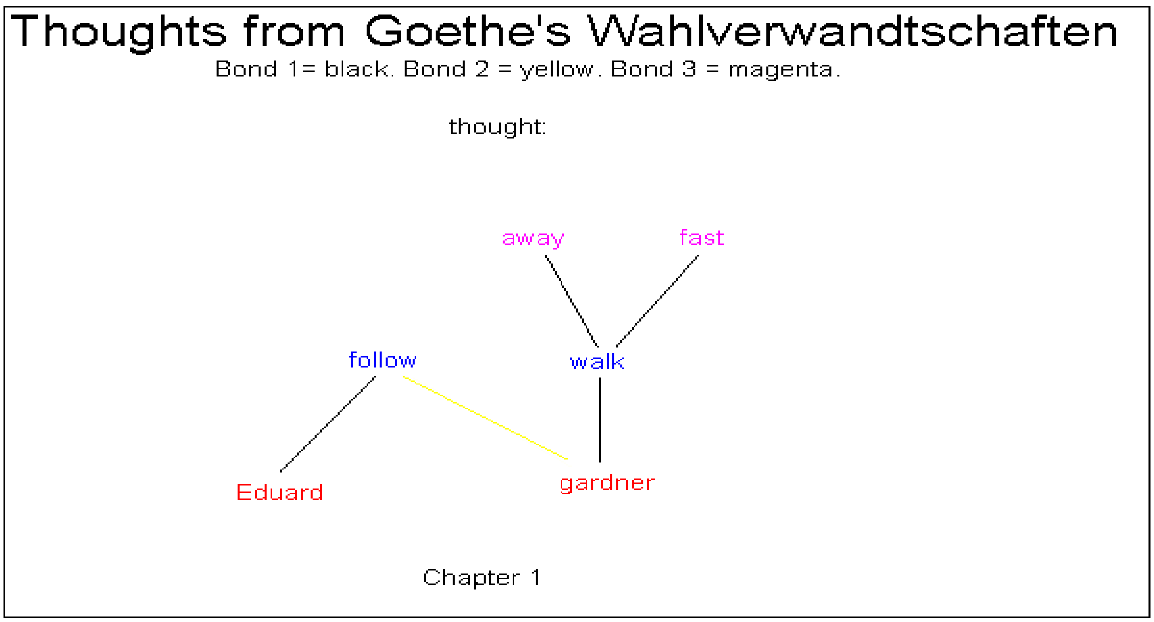


Figure 8.5. Interpretation:" Eduard follows the gardener who walks away fast".

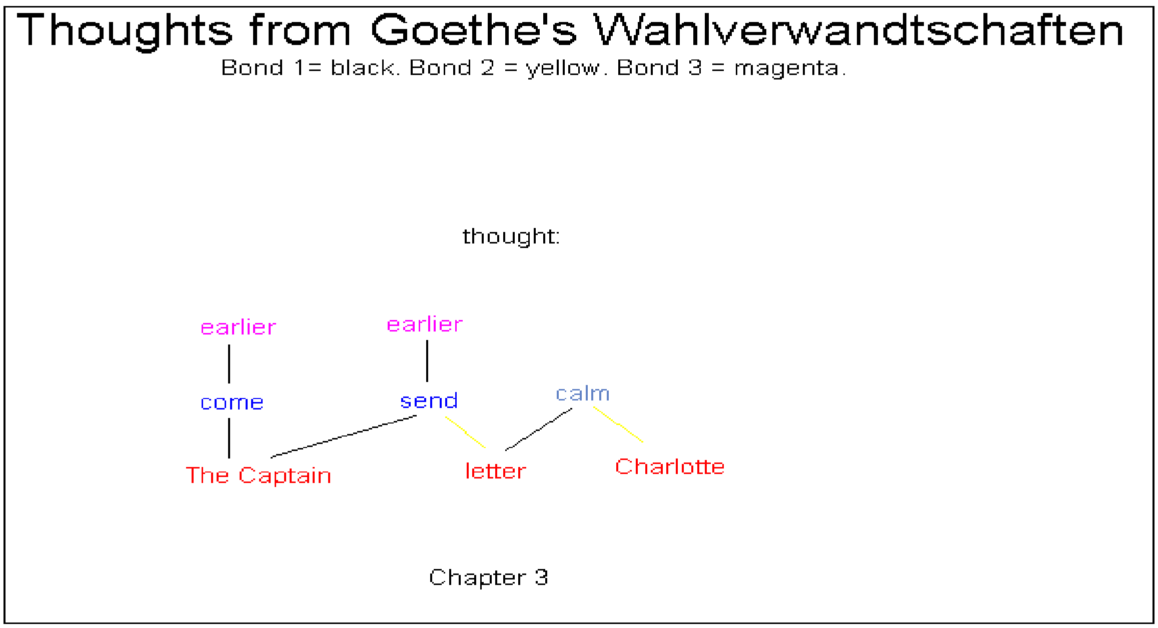


Figure 8.6. Interpretation:" The Captain came earlier and sent earlier a letter to calm Charlotte".

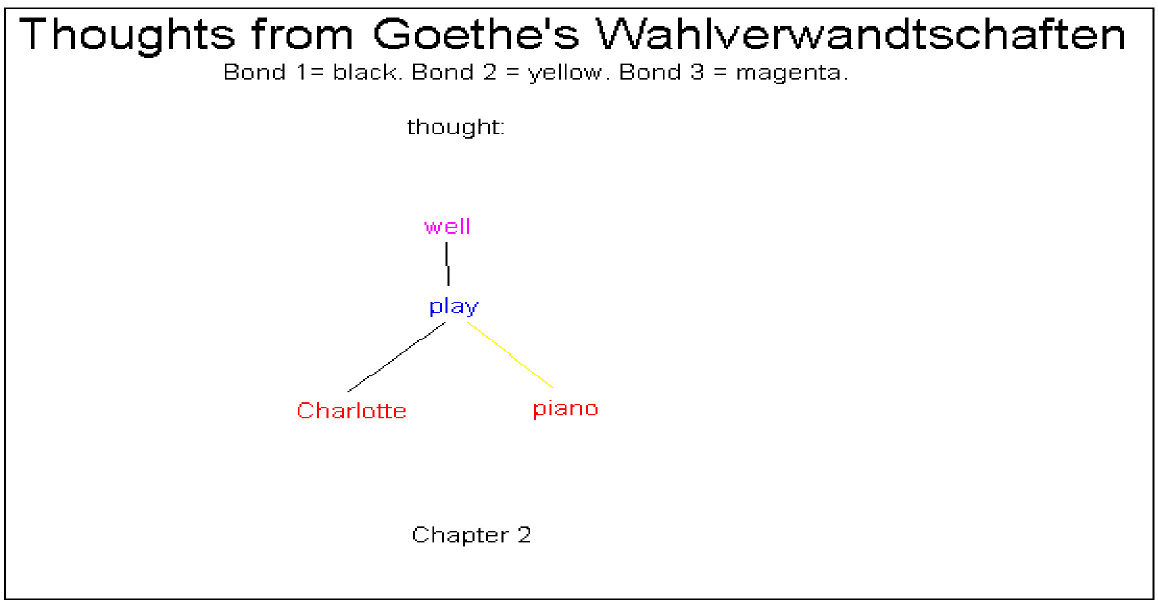


Figure 8.7. Interpretation:" Charlotte plays the piano well".

Some of these examples show connected graphs, or, to use our terminology , they represent conscious thoughts. This is the result of thought chatter, eventually resulting in a dominating thought. Chatter may look like

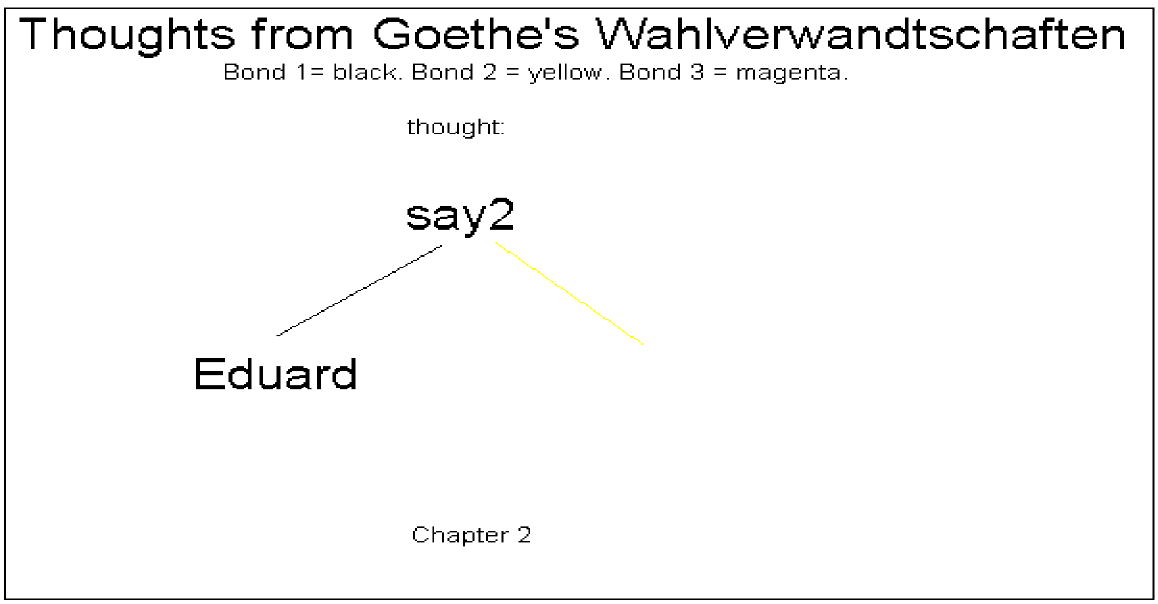


Figure 8.8. Interpretation:" Note that bond No. 2 from "say" (in yellow) is not connected.

If the thought chatter had been allowed to continue it may have led to a complete thought.

Figure 8.9 illustrates how Goethe makes a generalization, using A, B, C, D as modalities:

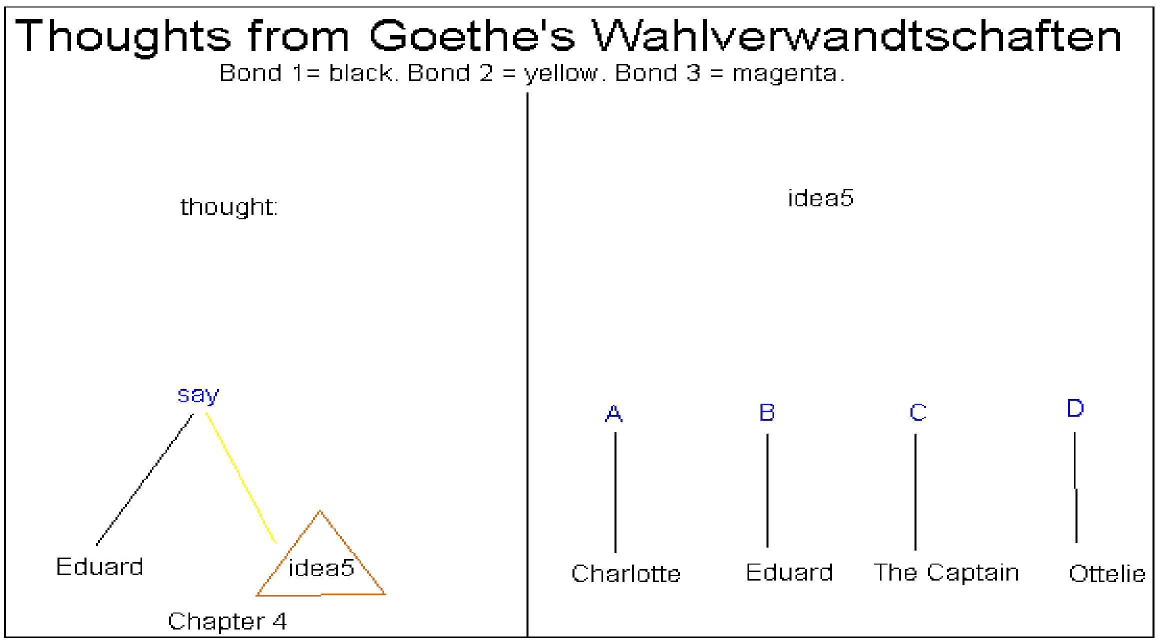
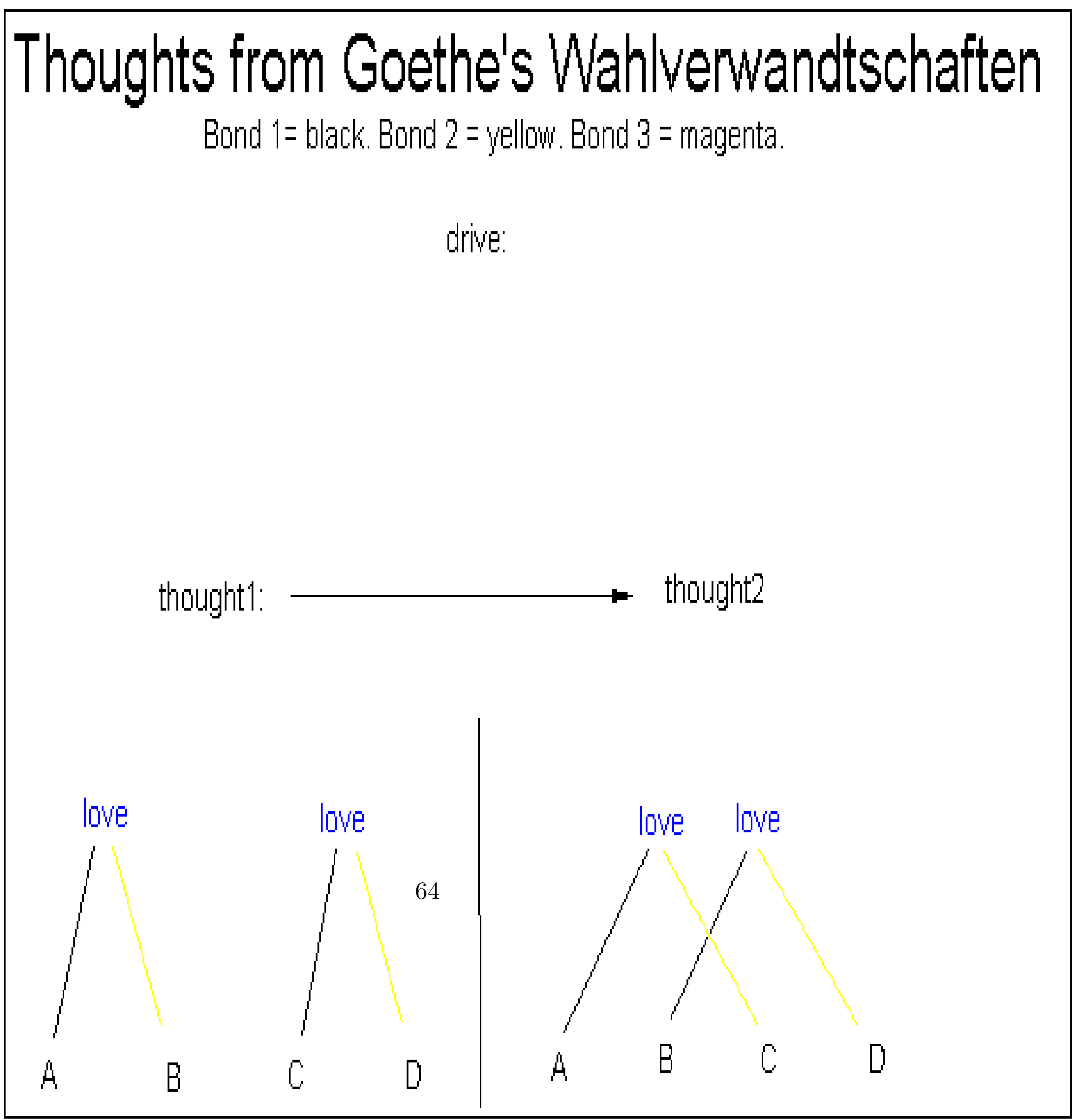


Figure 8.9. Interpretation:" Eduard says idea5", with idea5="let the modality A contain Charlotte, the modality B contain Eduard,..."


Thoughts from Goethe's Wahlverwandtschaften
Bond 1= black. Bond 2 = yellow. Bond 3 = magenta.
drive:
thought1:
thought2
love
love
A
B
C
D
love
love
A
B
C
D

="A loves B and C loves D"; thought2="A loves C and B loves D"

It actually represents a thought transformation with a composite move, a double romantic relation changes into another. Or,

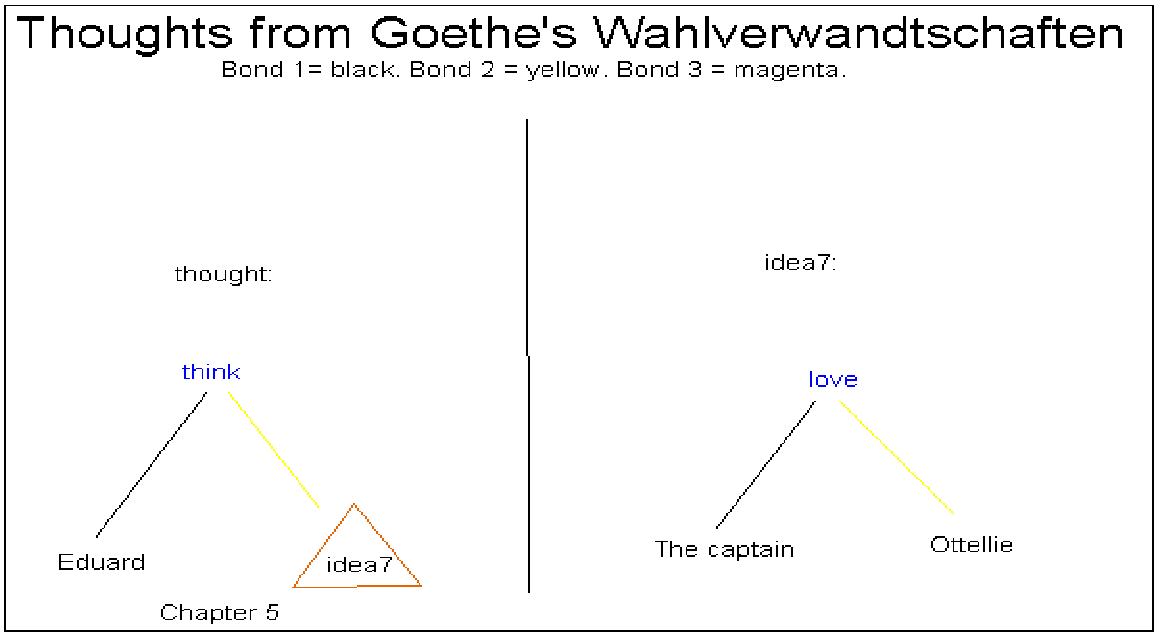


Figure 8.11. Interpretation:" Eduard thinks that idea7" with idea 7=" The Captain loves Ottelie".

Another abstracion with a new idea9 is seen in Figure 12

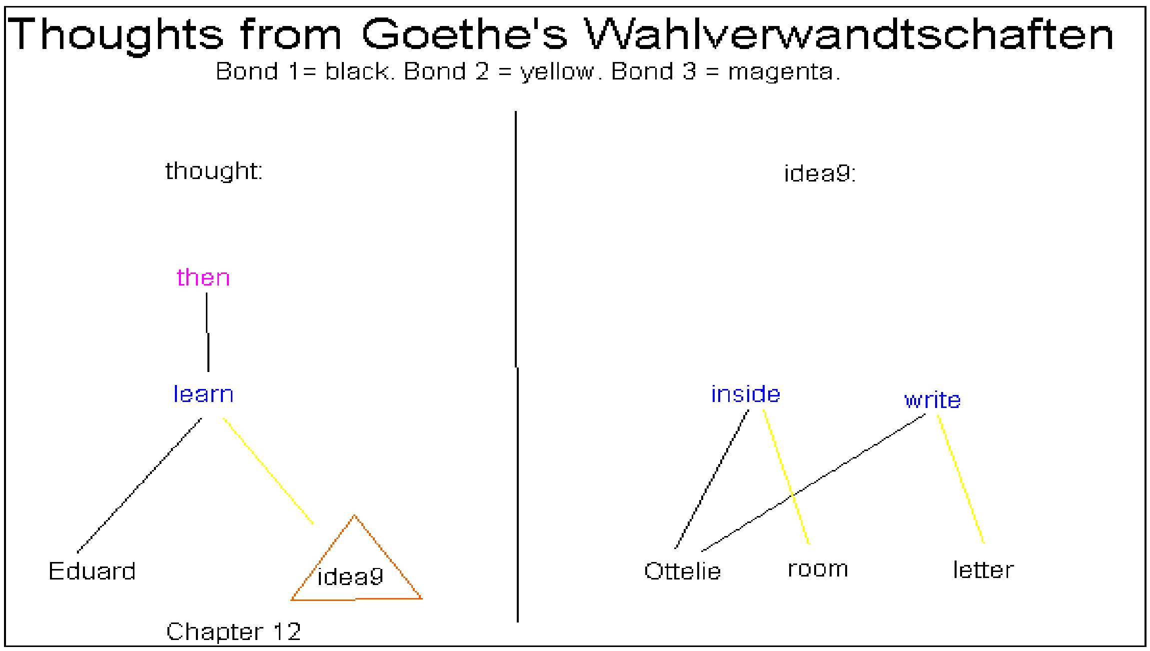


Figure 8.12. Interpretation:" Eduard then learns that Ottelie is inside the room writing a letter".

Enough of Goethe. We hope that other researchers will expose GOLEM to other literary or artistic environments, so that it can develop more rounded personalities. That is for later. But how can we automate such thought processes, how can we write code that realizes the pattern theoretic structures?

9 Building a Golem

"But how can I make a Golem?"
thought Great Rabbi Loew

As described in Section 1.3 we shall judge a mind model through the performance of a software realization of the model. We could say that we shall build a *Golem*, an artificial creature with some thinking abilities. A Golem could be said to belong to sphere of *artificial life*.

But can we build a Golem using the principles announced above? That is, can we present a concrete system in the form of a computer program, that exhibits some of the characteristics we believe characterize the human mind? We shall develop such a system in computational form, a first attempt, admittedly not very successful, but hopefully to be followed by a series of successively more sophisticated systems, perhaps culminating in one with a reasonably anthropoid behavior.

For the sake of programming ease, but at the cost of loss of speed of computation, we select MATLAB as the programming language.

9.1 Data Structures for the Mind

We believe that the choice of data structures is of more than peripheral interest. Indeed, the architecture of a Golem must be expressed in terms of data structures. The data structures we propose in the following are not arbitrary but are the result of careful consideration and likely to be the preferred choice in future realization of Golems even if it is expressed in a different programming language and with more complex implementation. The form of these structures has proven powerful. We recommend that the reader takes takes a careful look at the program code given below.

9.1.1 Data Structures for Elementary Ideas

Generators will have three attributes: name, level and modality. To handle this efficiently we shall let the generator space G be a MATLAB *structure* with the fields 1) name, as a character string, 2) level, as a numeric scalar, and 3) modality, also as a numeric scalar representing names in a variable "modalities". We enumerate G by an index g so that the gth one is

$$G(g) \in G; g = 1, 2, \ldots r \tag{52}$$

with three fields: the name $G(g).name$, the level $G(g).level$, and the modality $G(g).modality$.

To make the following concrete we shall use examples to clarify what we have in mind. The actual software that we shall use is going to be much more extensive but constructed in the same way as indicated by the examples. Some of the 1-level generators could be

G(1)=
name: 'man', level: 1 modality: 1
G(2) =
name: 'boy', level: 1 modality: 1
G(3)=
name: 'self', level: 1 modality: 1
G(4) =
name: 'Peter', level: 1 modality: 1
and some of other modalities:
G(30) =
name: 'chair', level: 1 modality: 8
G(100) =
name: 'jump', level: 2 modality: 28
G(120) =
name: 'today', level: 3 modality: 38

We could use for example the modalities (many more have been added in the MATLAB implementation)

1: humanM , M for male
2: humanF , F for female
3: animalM
4: animalF
5: food
6: vehicle
7: building
8: furniture
9: tool
10: machine
11: body part
12: idea transfer
13: apparel
14: capital
15: social group
16: size
17: color
18: smell
19: taste
20: sound
21: emotion
22: affect
23: hunger
24: greed

25: awareness
26: family relation
27: social-relation
28: movement
29: eat
30: feel
31: likeHA H for human, A for animal
32: likeT T for things
33: activity
34: direction
35: quality
36: quantity
37: where
38: when
39: change hands
40: libidoH
41: libidoA
42: amicus relation
43: active ideas
44: new ideas

and many more. As we have noted before, signifiers like *man*, *likeT* and *change hands* should not be understood as words, but instead as concepts. We can get the modalities

$$humanM = \{man, boy, self, Peter, Paul, ...\} \tag{53}$$

$$likeT = \{likeINAN, dislikeINAN, ...\} \tag{54}$$

$$changehands = \{give, take, ...\} \tag{55}$$

The concept *humanM* means, for this mind, a man in general, a boy in general, the self = the carrier of this MIND, the particular man called Peter, or the particular man called Paul. The concept *LikeINAN* means to like or dislike something inanimate. The concept *changehands* means to give or to take, etc.

The connectivity of MIND will be given by the Matlab *cell* "mod-transfer" consisting of one cell for each modality, each cell with three sub-cells with numerical 3-vectors (possibly empty) as entries. For example cell no. 32 :likeT in this MIND could look like

$$likeT = (1, 2; 5, 6, 7, 8; \emptyset) \tag{56}$$

meaning that the modality is of arity 2 with the first bond extending downwards either to modality 1 or 2, the second bond to either 5,6,7, or 8 and no third bond. For simplicity we have limited the down-arity to three but that could easily be extended; we have not yet needed this. This ordering induces automatically a partial order in the generator space G.

9.1.2 Data Structures for Thoughts

To represent thoughts we use two arrays:

1) an $n \times 2$ matrix "content" witn n =no. of generators

$$content = \begin{pmatrix} h_1 & g_1 \\ h_2 & g_2 \\ ... & ... \\ h_n & g_n \end{pmatrix} \tag{57}$$

where $(h_1, h_2, \ldots h_n)$ means the *set* of generators in the configuration, expressed in h-coordinates and $(g_1, g_2, \ldots g_n)$ the *multiset* of generators expressed in G-coordinates. The h's are assigned to generators as they appear one after another during the mental processes, numbering them consecutively, so that all the h's are distinct in contrast to the g's that can take the same values more than once; an idea can contain reference to for example "man" more than once.

2) an $m \times 3$ matrix "connector", with $m = no.\ of\ connections$

$$connector = \begin{pmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ ... & ... & ... \\ j_{m1} & j_{m2} & j_{m3} \end{pmatrix} \tag{58}$$

This second matrix has three columns for each connection. For the first segment j_{11} is the h-coordinate of the start of the downward segment, j_{12} is the h-coordinate of the end segment, and j_{13} is the j-coordinate of the generator from which the downward segment emanates, and so on for the other connections of this thought. See Figure 10.1.2.1

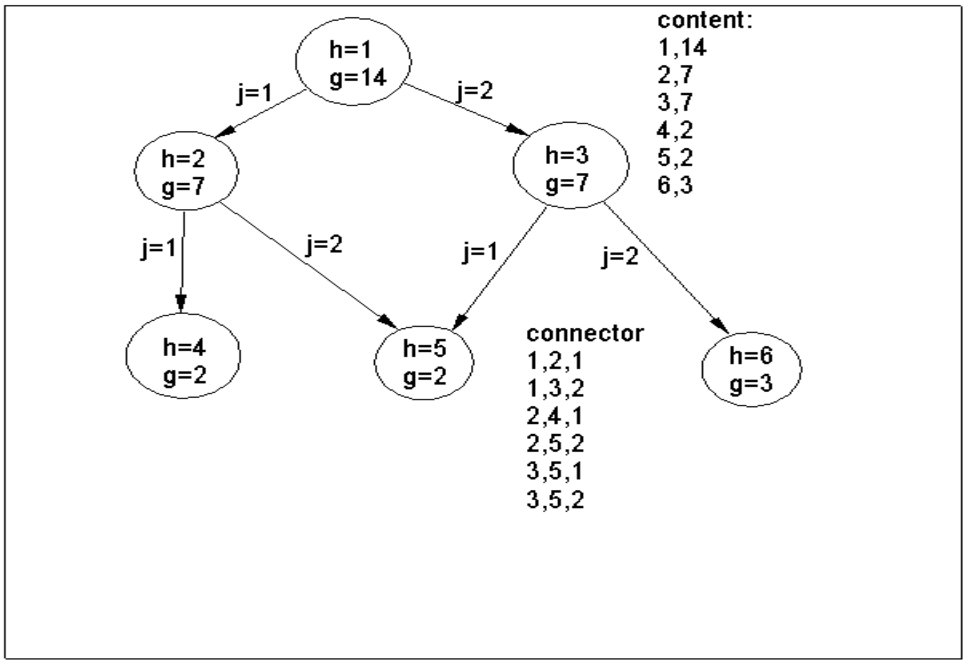


Figure 10.1.2.1.

We shall pay some attention to-top ideas of level 2 including at most 3 generators on level 1; see section 6.1.2 Of course this reduces the intellectual power of the mind to the extent that it is unable to operate with abstractions on higher levels as far as top-ideas are concerned, but it can handle more complex abstractions by other means. We use the following data structures for such thoughts. If the top of a "thought" is $g_{top} = g_0$ and the subordinated generators are g_1, ... g_p expressed in g-coordinates, and with p at most equal to 3, we shall enumerate it with the *Goedel number*

$$goedel(thought) = \sum_{k=0}^{p} r^{g_k}; r = |G| \tag{59}$$

in other words, we use the base r radix representation.

9.1.3 Energies of Thoughts and Genres

It is easier to find suitable data structures for the mental energies. Indeed, we shall let q be a numeric r-vector and a be a numeric $r \times r$ matrix. The same data structures for the weight function $Q(g) = exp[-q(g]); g = 1, 2 \ldots r$ and the acceptor function (matrix) $A(g_1, g_2) = exp[-a(g_1, g_2)]; g_1, g_2 = 1, 2, \ldots r$.

This makes it easy to represent genres. Consider a genre called $genre \subset G$ consisting of the generators that characterize this *genre*. Then we could modify the Q vector to take two values: *max* and *min*

$$Q(g) = max; x \in genre; Q(g) = min; g \notin G \tag{60}$$

Actually we shall use a somewhat more involved modification that will make it possible to account for the development of the mind including changes in genre energies.

As examples of the genres of the mind that we will use we mention the following:

1) *emotional relationHA* between humans & animals
2) *ownership* among humans and property
3) *play pets* for human and pets
4) *work* for humans
5) *relax* for humans
6) *movement* for humans and animals
7) *interior design* for house and home
8) *sports* for humans
9) *reasoning* among humans, not purely logical but also, well, *unreasonable reasoning*
10) *talking* among humans
11) *eating* among humans & animals
12) *objects* about inanimate objects
12) *abstract thinking* with $Q = max$ for those g's for which $MOD(g) = g$
13) *emotionalHH* about emotional relations between humans

We shall also allow Boolean combinations of genres, for example *work* $\vee$ *objects*, meaning to work with some object.

9.1.4 Composite Moves

The data structure of a driver is a bit more complicated. It will consist of four parts:

1) *change-thought* is an $2 \times n_{thought}$ Matlab cell; $n_{thought}$ is the size of the sub-"thought" that the mind is currently thinking about. For each subcell, $k = 1, 2, \ldots n_{thought}$, a choice is made between a) deleting the generator, or b) keeping it unchanged, or c) change to another g-value , or d) choose a random a new g-value from a given set.

2) *ad content* adds a set of new generators

3) *ad connector* adds connections but only inside the "sub-thought"

4) *delet connector* deletes connections but only within the "sub-thought"

We have already seen a number of examples of drivers in Section 2.10.

9.2 Program Hierarchy for the Mind

The GOLEM code is complicated and deserves the reader's attention: it includes many ideas and devices that have not been discussed in the text. Therefore we recommend that a reader who wants to really understand the working of GOLEM to at least glance through the code given in APPENDIX 4.

9.3 Putting it All Together

To build a Golem by successively introducing new entities we can proceed as follows:

a) To introduce a new generator in an existing modality use *set_G*, followed by redefinition of MIND arrays gs_in_mods, get_levels, $get\_mod\_transfer$, $get\_mod\_transfer\_inv$, *set_Qs*, *set_As*, *set_g_mod* and *set_mod_omegas*.

b) To introduce a new modality use *set_modalities* followed by get_levels.

c) Then use *print_G* to print the generator space with numbers and *print_modalities* to print modalities with names.

d) Use *see_modality* to display a single modality graphically and *see_mind* to display the current configuration.

The above family of programs is combined into the *main function* "think" which displays a menu allowing the user to choose between the following alternatives:

1)$ThinkingDrivenbyThemes$. This is the main mode of "think" with several options for the themes.

2)$ContinuousThought$. In this mode the MIND trajectory jumps between different themes and creates new ideas occassionally.

3) $Thinking\ Driven\ by\ ExternalInputs\ of\ the\ Mind$ The user inputs elementary ideas and the MIND makes *inference* from them to build a new thought.

4) $Free\ Associations$ where the trajectory through mind space consists of small steps of simple moves following the probability measure P, not driven by any other outer or inner forces. The result is fairly chaotic, unorganized thinking.

5) $SetPersonalityProfile$ in which the user defines a personality of "self".

6) $SetMindLinkages$ sets the mind parameters Q and A for a given personality profile.

7) $TheVisibleMind$ displays the connectivity of the MIND.

8) $SeeCreatedideas$ displays the new created ideas.

10 A Golem Alive ?

Now let us see what sort of thought patterns are generated by the GOLEM anthropoid. The best way of studying the behavior of the program is of course to download the code and experiment with it oneself; the user is strongly encouraged to do this. Here we only present some snapshots and hope that they give at least some idea of the functioning of this MIND. Let us recall, however, that we do not view ideas and thoughts as words and sentences; instead we consider thinking as a flux of emotions, impressions, vague feelings, etc. The fact that the following diagrams involve words is just an admission that we do not (yet) have access to better representations than the verbal ones. An attempt to do so can be found in

10.1 Free Associations.

To begin with let the GOLEM move freely through its mental space, not influenced by inner or outer constraints. Make the Q and A functions constant and so that the bindings are quite weak: one simple idea that has occurred to the MIND has little influence on the following ones. The partial ordering that we have imposed via the modality lattice prevents the resulting thoughts from being wildly meaningless, but the semantics is far from consistent; how to improve this will be seen later on.

As the program executes it shows a sequence of snapshots of the mind, one mind state is followed by another struggling to reach the level of consciousness. Here we can only show a few of the snapshots; executing the software gives a better idea of how the MIND is working in this mode. In Figures 10.1.1 - 10.1.4 we se some mind states under (very) free associations.

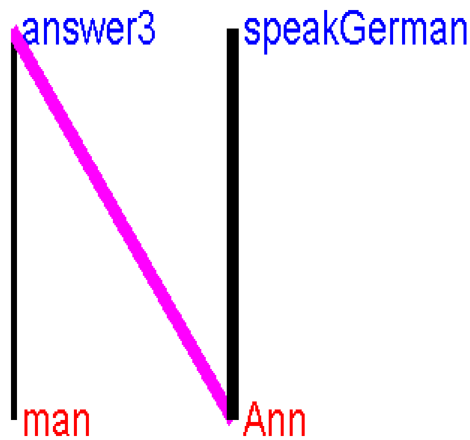


Figure 10.1.1
Man answers Ann who speaks German.

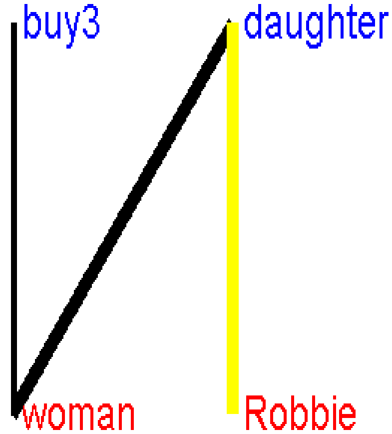


Figure 10.1.2

A woman is the daughter of Robbie, but what does she buy and from whom? An incomplete thought.

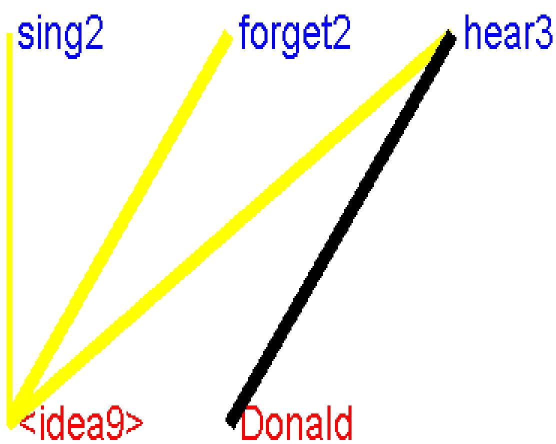


Figure 10.1.3

Donald hears an idea, but who sings and who forgets? Not clear due to the incompleteness of the thought!

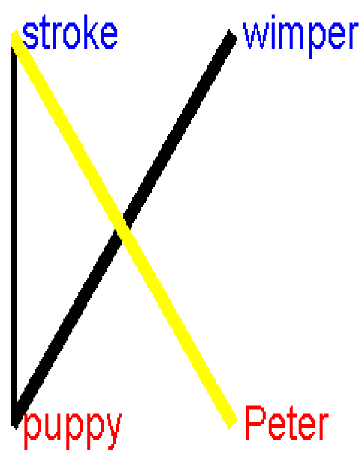
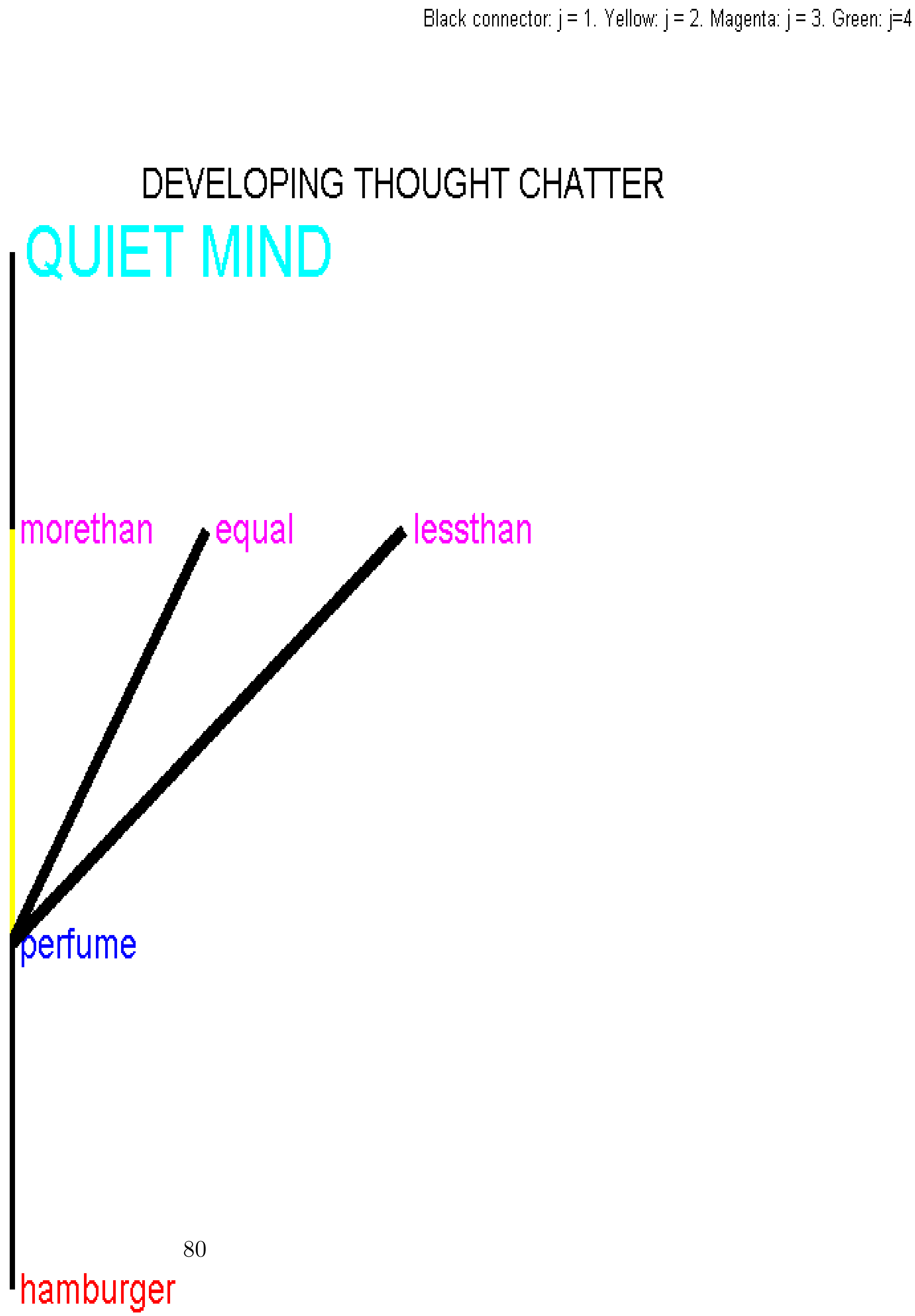


Figure 10.1.4 Peter strokes the puppy who wimpers - a complete thought.


Black connector: j = 1. Yellow: j = 2. Magenta: j = 3. Green: j=4
DEVELOPING THOUGHT CHATTER
QUIET MIND
morethan
equal
lessthan
perfume
hamburger

Here the thinking is disorganised, perhaps the GOLEM is dreaming about the smell of a hamburger. The ideas on the third level seem unrelated. However, the user can instruct the GOLEM to concentrate its thinking, *try to connect sub-thoughts* that appeared disjoint and independent. The way to do this is to choose the option "Concentrated Thought". The resulting idea will appear concentrated with its sub-ideas connected to the extent that regularity and the structure formula allows. This option can be applied in some other modes of thinking too. It will have a noticeable effect only when the connector is disjoint.

10.2 Inferential Thinking.

Now we force the Golem to start from given external inputs and continue it further by the inference process described in Section 3.8. Say that GOLEM starts with the MIND's iput being "cash", *genre*= BUSINESS, Black connector: j = 1. Yellow: j = 2. Magenta: j = 3. Green: j=4

DEVELOPING THOUGHT CHATTER

cash

Figure 10.2.1a
with the inference in Figure 10.2.1b: a visitor gives cash to Carin

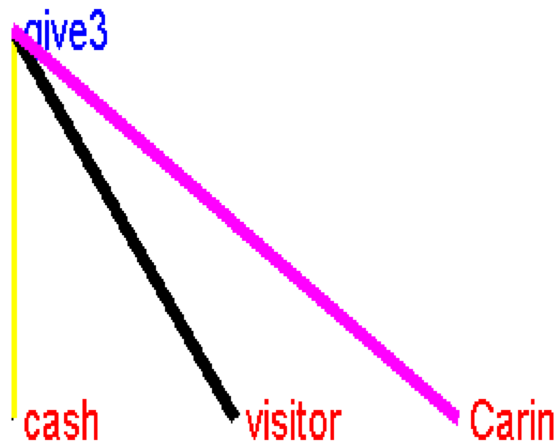


Figure 10.2.1b
or with the inputf "aspirin"

Black connector: j = 1. Yellow: j = 2. Magenta: j = 3. Green: j=4

DEVELOPING THOUGHT CHATTER

aspirin

Figure 10.2.2a
an inference is

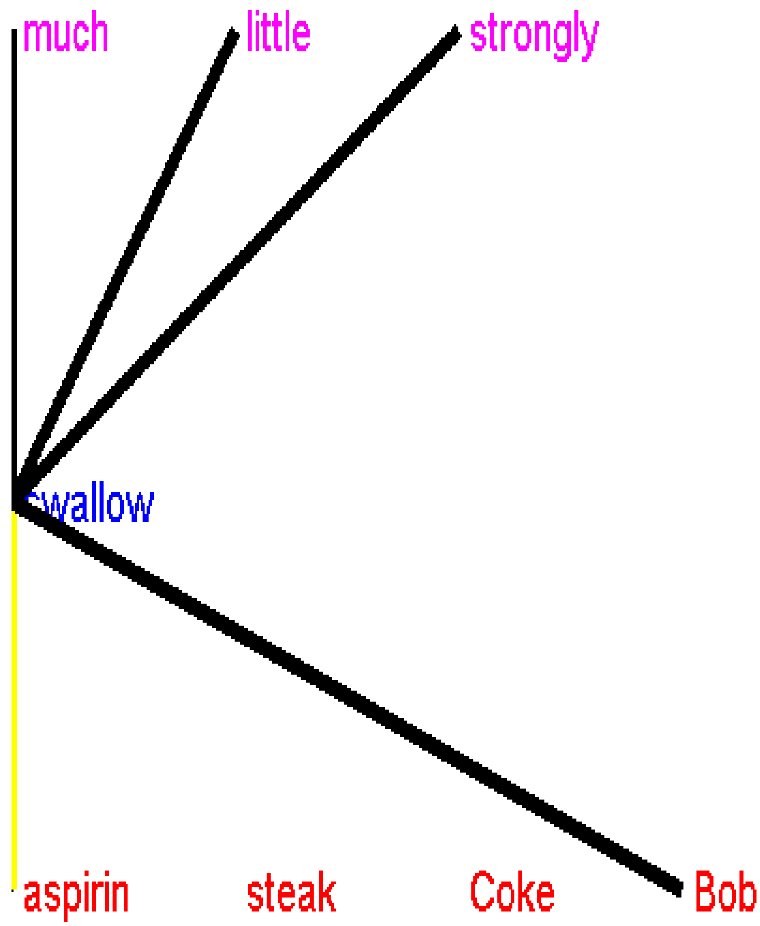


Figure 10.2.2b

with the inference in Figure 10.2.2b that Bob swallows aspirin but with some additional thought chatter.

Starting with the idea of Republican

Figure 10.2.3a
the inference is in Figure 10.2.3b

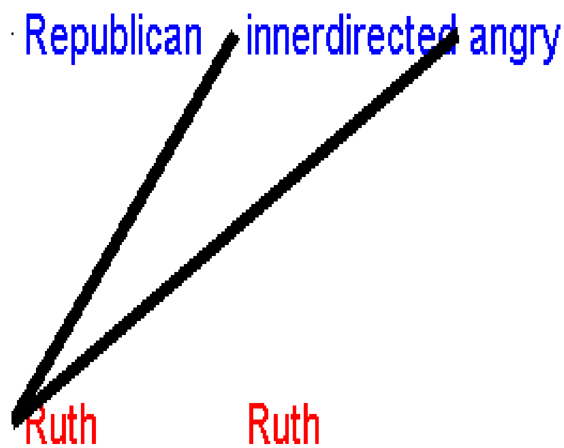


Figure 10.2.3b

which is more or less meaningless, free associations. But human thought can develop in strange ways!

10.3 Associations Driven by Themes .

Golem can carry out thematic thinking. The sub-thoughts are connected internallly, to the extent that regularity and randomness allows, but disconnected externally. Once the inputs are defined, Golem can start thinking, influenced by the inputs. Here is one thought from the theme Sports (with Linda plays)

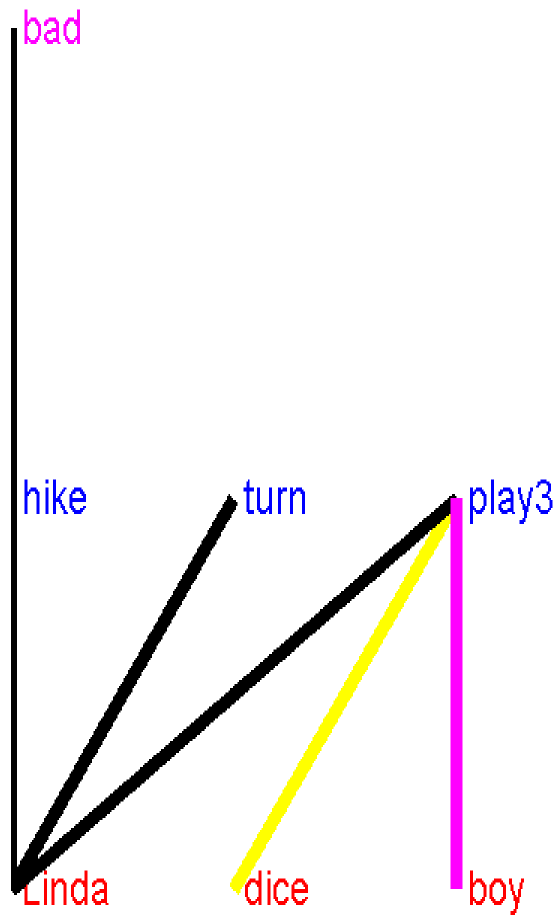


Figure 10.3.1
Linda plays dice with a boy. She also turns and hikes badly.
Another thematic thought from the theme Business

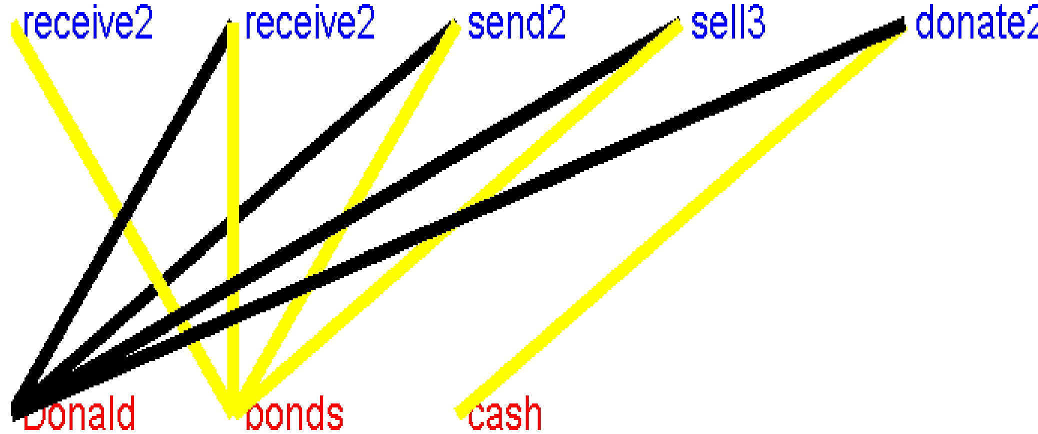


Figure 10.3.2

Donald carries out complicated transaction with belongings changing hands.

For the theme Pets we get

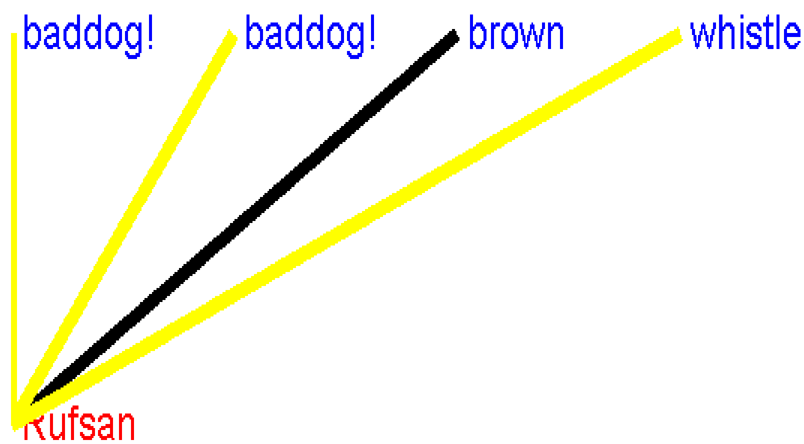


Figure 10.3.3

The thought is highly incomplete. The only completed sub-thought is that Rufsan is brown, but it is not clear who whistles at her and tells her she is a bad dog (repeatedly). We believe that such incompleteness is typical for some human thinking.

And the theme Business again:

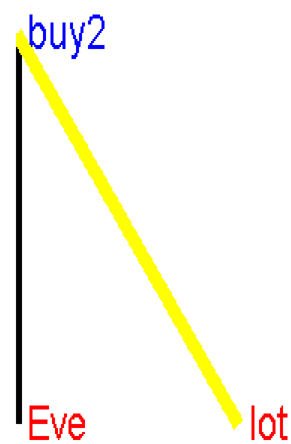


Figure 10.3.4

Eve buys a lot. In these figures we have not shown the thought chatter that induces the resulting thought; that can be seen by running the software and is quite instructuive.

10.4 Continuous Thought.

This is an important option and deserves our attention. Among all the sub-ideas, complete or incomplete, that exist in the mind at any given mind, only some reach the level of consciousness as was discussed earlier. To see how this happens execute option " Continuous Thinking" that shows thought chatter and later the resulting thought. It moves via a Markov chain throught the themes, see section 5.5. The user is asked for the duration of thinking, choose a low number. During the thinking the direction of the mind trajectory may change, if this happens it is announced on the screen. Also, if a new idea is created and added to the generator space that is announcec. New ideas can be displayed using the option " See New Created Ideas" in GOLEM. For example

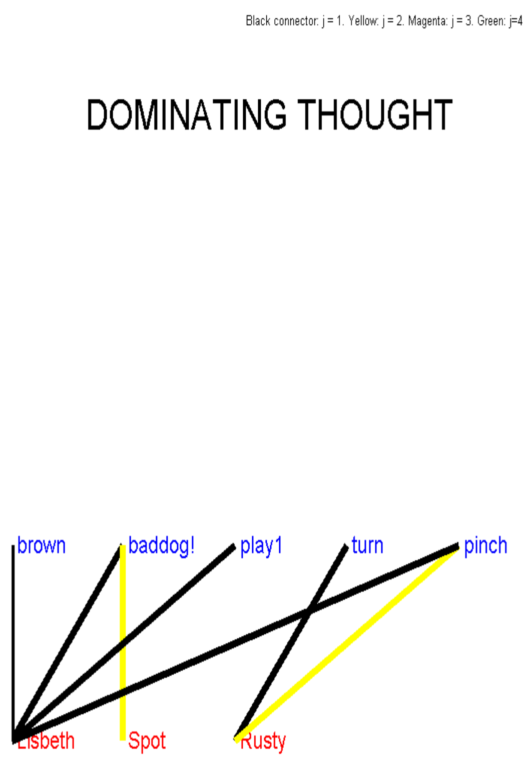


Figure 10.4.1

in which Lisbeth tells Spot he is a bad dog and also pinches Rusty who turns. Lisbeth is tanned. A thought chatter:

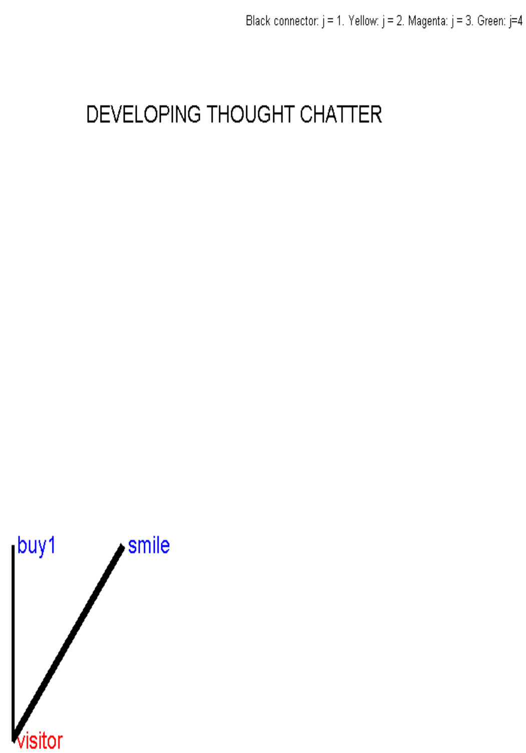


Figure 10.4.2
the visitor is smiling while buying. Or,

EMPTY MIND

Figure 10.4.3

with no resulting thought, the mind is at rest! Again continuous thinking:

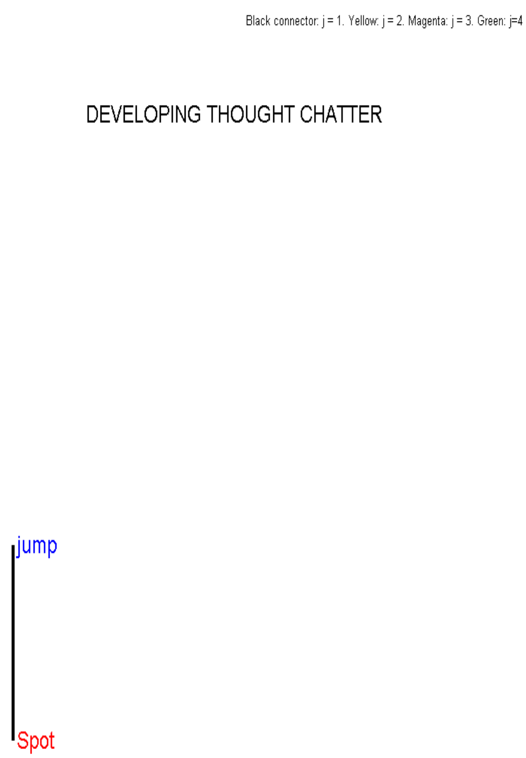


Figure 10.4.4
Spot is jumping.

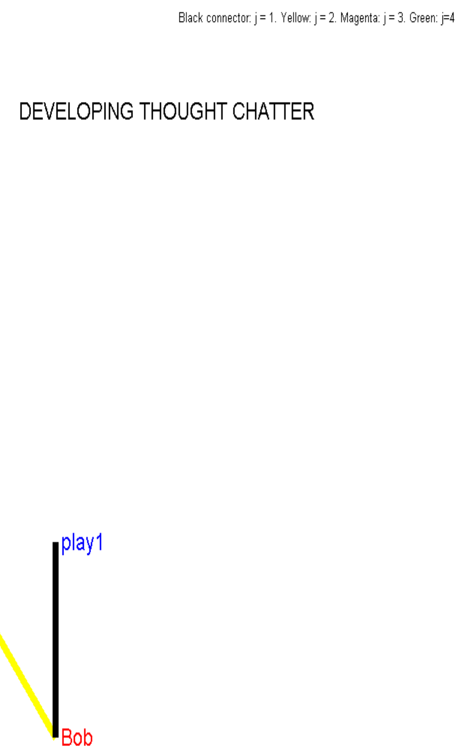


Figure 10.4.5
Helen strokes Bob who plays.

10.5 See Created Ideas.

To display ideas that have been created and added to the generator space choose the option "See Created Ideas". For example

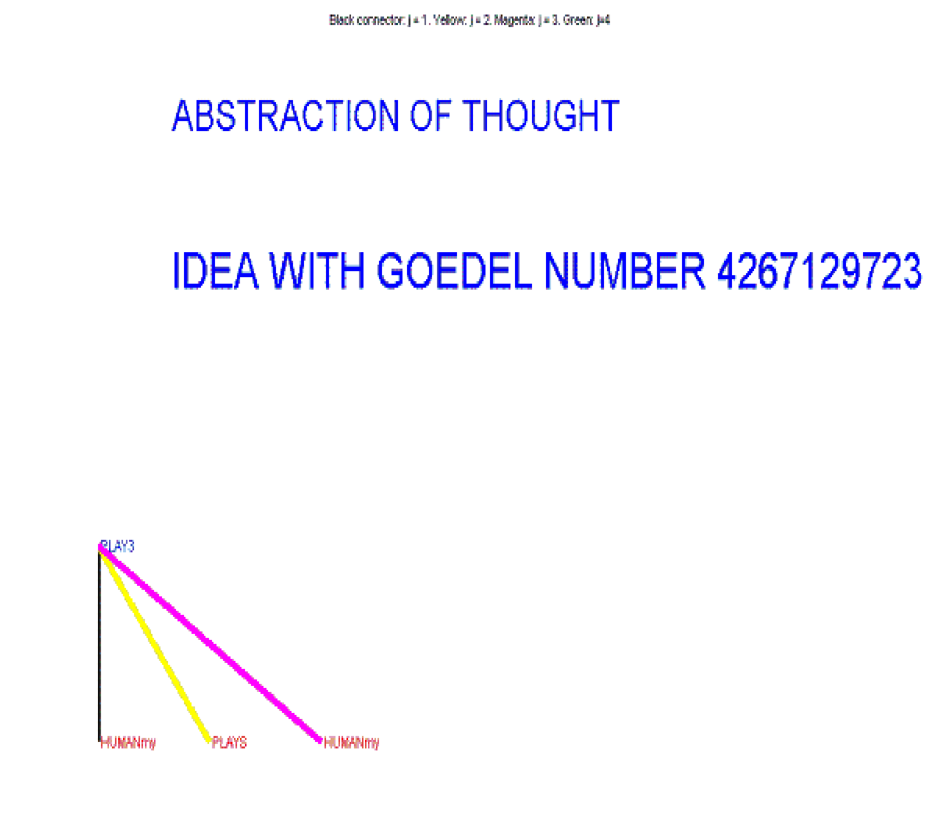


Figure 10.5.1

Two young males play a game with each other.

We have only experienced with a few drivers. One of them is $love\_driver\_1$; in Matlab form as a "cell(6,1)" with the first sub-cell

$$\begin{pmatrix} change & 247 \\ same & [] \\ same & [] \end{pmatrix},$$

the three next sub-cells empty (no generators or connections added), the fourth one .8 (activation probabability, and the sixth one the domain of the driver (246, humanM,humanF). This driver searches the configuration for top-2ideas that belong to the driver. If it finds one, it replaces generator g=246, meaning "love", with generator=247, meaning "desire". We use the program "build-driver" for constructing drivers and"execute-driver" for executing them. We get for example starting with the idea "Jim loves Joanie"

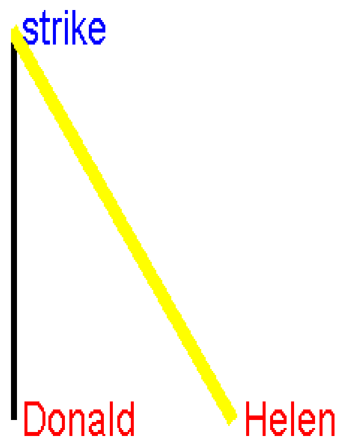


Figure 10.5.2

driven into the new idea "Jim desires Joanie"

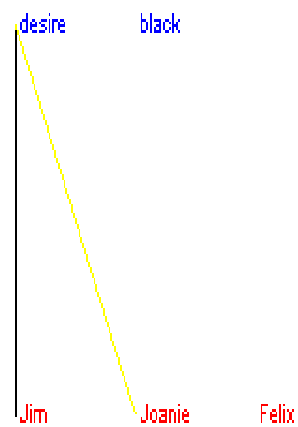


Figure 10.5.3

10.6 Generalizing Top-ideas.

One of the options for GOLEM is to determine the top-2ideas currently in consciousness, and then generalizes them (first order) into the modality lattice to get a thought pattern (see Section ?). We get for example

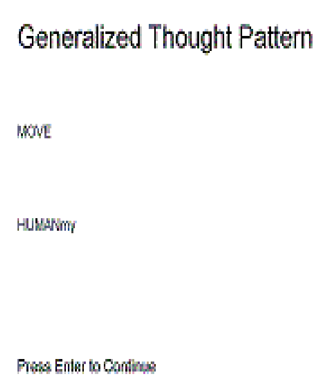


Figure 10.6.1
signifying the concept of a moving young male. And

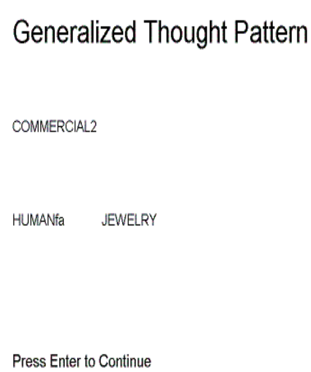


Figure 10.6.2

which shows the thought pattern when a capital transactions involving jewelry takes place to a female adult.

10.6.1 A Developing Personality

After running GOLEM for a substatial time the MIND has changed: its linkage structure has been modified due to internal and external activities. To illustrate this look at Figure 10.7.1

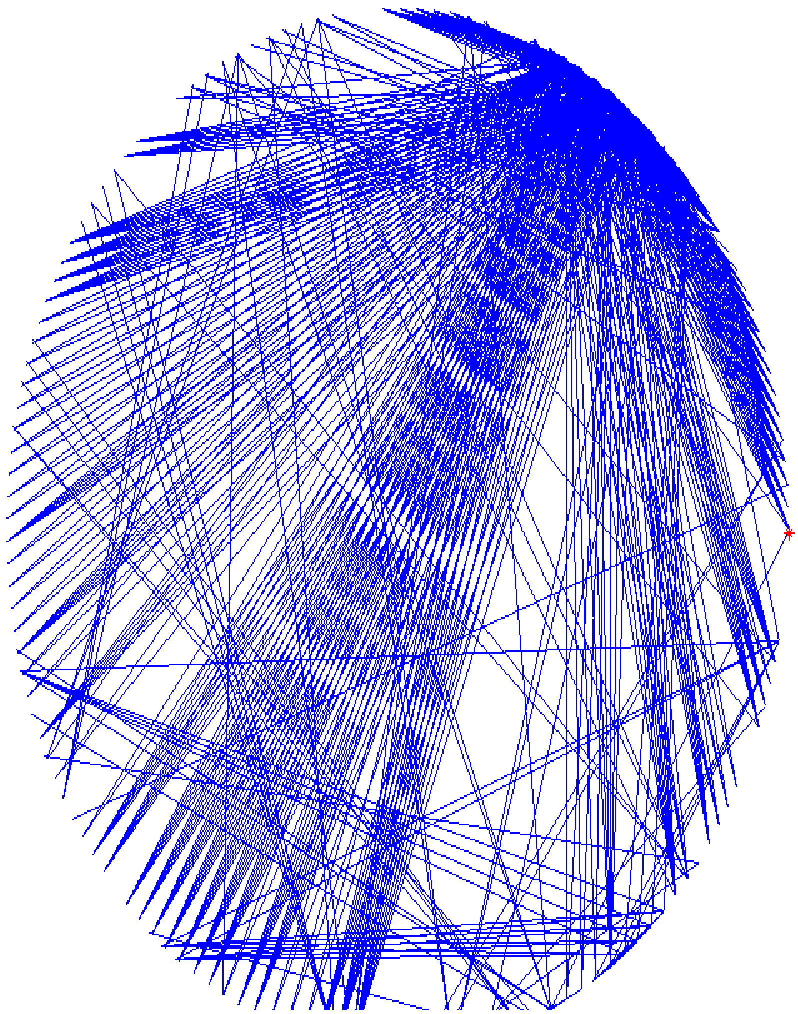


Figure 10.7.1
that exhibits the linkages at an eraly stage of development, and Figure 10.7.2

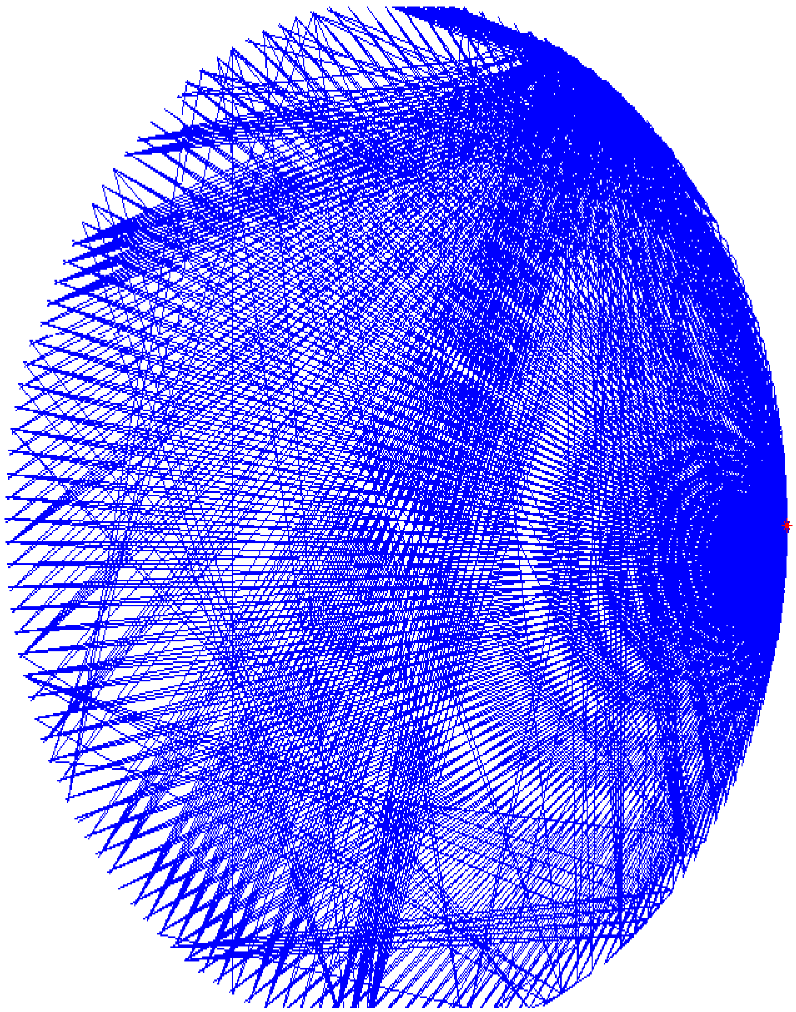


Figure 10.7.2

where we see many more links established. Note in particula the increased activity close to the elementary idea "self", indicated by a small red star to the right in the diagram.

WARNING: This will take a long time - first the development cycles, hours, and then display, minutes.

This inspires to more experiment studying the mental development of MIND under different external environments and themes. How does the linkage structure change if GOLEM is run without external inputs? Or, if it is exposed to a single theme. And, if "self" has developed very aggresive - what sort of inputs should one apply to MIND in order to improve the behavior: another option THERAPY. The reader is encouraged to play with the DEVELOP option in "think".

10.7 Judging the Behavior.

How well does Golem imitate human thinking? Not very well, but it clearly attempts to do so. Under Free Associations the thinking ought to be somewhat chaotic but Golem's thoughts appear *very* disconnected. The connections between sub-thoughts are *too* random, they should be more strongly coupled to each other. The performance is much better under Continuous Thought ant Thinking Driven by Themes, and this gives a hint for improvment. The set of themes ought to be refined into many more and more specific, narrower, ones. As one theme is followed by another the direction of the GOLEM trajectory changes, but in between jumps the probabilistic dependence seems adequate.

To improve the semantics the generator space must also be extended. In the current version we have used
$r = 726$ generators organized into
$M = 180$ modalities. This is clearly insuffient. Perhaps $r = 5000 - 10000$ and $M \approx 1000$ would be adequate. To implement this would require more manpower than what the author has has available. It should be mentioned, however, that a substantial research effort in AI has been directed to defining a large (6000?) set of concepts and relations betweeen concepts; see www.opencyc.org. Perhaps this could be used to extend GOLEM. Also, the modalities should take into account a *taxonomy of ideas*, expressing how human knowledge can be organized into fine categories. This will require more levels representing different degrees of abstraction.

Perhaps Golem should also produce outputs: movement, speech, external reactions, limbic response and so on. We do not see how this can be attained and how to express such outputs. Possibly by using avatars. This will be neccessary to allow for interactions between Golems to be discussed below.

Although GOLEM's performance in imitating the human mind is not impressive, it indicates that a degree of verisimilitude can be achieved by a probabilistic algorithm. When de La Mettrie opened a discussion on the theme *L'Homme machine* it began a discourse that would have delighted the School Men. We shall certainly avoid getting involved in this morass of vague philosophizing. Instead of the metaphor of a machine, with its image of cog wheels and levers, or transistors on silicon, we shall only claim that *the mind can be viewed as an entity that is subject to laws, probabilistic to be sure, but nevertheless regulated by definite rules.* Our main task is therefore to formulate and verify/falsify hypothetical laws of the mind.

Our conclusion is: *The human mind can be understood.*

11 Analysis of a Virtual MIND

Say that we observe the output of a virtual MIND without knowing its inner workings, and that we want to understand it. Here the term "understand" means knowing, at least partly, the parameters that characterize the mind:$G, \mathcal{M}, Q, A$ and possibly others. One could say that we want to perform *psychoanalysis without Freud*. It is known in general pattern theory how to estimate e.g. the acceptor function A. See GPT Chapter 20 and also Besag (1974), Osborn (1986), where however the connector graph σ is supposed fixed and not random as in GOLEM.

It will be more appealing to the intuition to use other parameters for the analysis. Indeed, Q and A do not contain probabilities as elements as may have been thought at first glance. For example, the entries in the Q-vector can be greater than one. Q and A are needed for the probabilistic generation of thoughts but are not simply related to probabilities of simple events. Instead we shall introduce parameters that have a direct interpretation but are not simply related to the Q and A.

For any positive content size n and any generator $g \in G$, consider the average of the conditional probabilities

$$f(g|n) = \frac{1}{|\sigma|} \sum_{i=1}^{n} P(g_i = g : |\sigma| = n) \tag{61}$$

and

$$f(g) = \sum_{n=1}^{\infty} p(n) f(g|n) \tag{62}$$

so that $f(g)$ measures the possibility of MIND thinking the elementary thought g. Further, the expression

$$F(genre) = \sum_{g \in genre \subset GENRE} f(g) \tag{63}$$

measures the *propensity of a particular genre*.

Then we can estimate these parameters in a straight forward way. We simply replace the probabilities $P(g_i = g : |\sigma| = n)$ and $p(n)$ by the respective observed frequencies. But we can reach deeper into the structure of MIND. Indeed, let us fix two thought patterns $PATTERN \in \mathcal{P}$ and $PATTERN'$, and consider two (random) consecutive thoughts, $thought(t)$ and $thought(t+1)$ occurring to MIND at time points t and $t+1$. Introduce the conditional probability

$$Prob = P\{PATTERN' \in thought(t+1) | PATTERN \in thought(t)\} \tag{64}$$

measuring the likelihood that $PATTERN$ is followed by $PATTERN'$. We do not insist on any cause-effect relation, just temporal sequentiality.

For example, if $PATTERN$ is a pattern representing one person, the self, challenging another, and $PATTERN'$ represents violent action, then $Prob$ is a mind parameter with a rather clear interpretation as aggressiveness. Or, if $PATTERN$ stands for self and $PATTERN'$ for sadness, then $Prob$ could be understood as a tendency to depression.

It should be remarked that $PATTERN'$ corresponds to a sub-graph with many inputs, this can imply that this pattern is likely to be activated. This statement should be qualified by pointing out that the likelihood depends upon how the A-values for these inbonds have been modified by MIND's experiences during its development.

12 Where Do We Go From Here?

In spite of its less than impressive performance the Golem points the way to the development of more powerful artificial minds. The improvements suggested in the previous section will require much work and the development of auxiliary programs but nothing new in principle. However, we have started to see some more challenging extensions.

The notion of driver discussed above seems essential. We defined just a few but could easily add to them in the spirit of the composite moves in Section 5.4 using the program "build-driver. But this does not seem the right way to go. Instead the creation of new drives ought to be wholly or partly automated, maybe through extremum principles (energetic ones?). As Golem is experiencing new inputs from the external world, and perhaps from interactions from other individuals, it ought to solidify its experiences into drivers. This should happen over long intervals of time. It is not clear how to arrange this.

The Golem should live in a world inhabited by other Golems, similar but not identical to it. They should exchange ideas and modify themselves as a result of such symbiosis - *a mind game*. For this it is neccessary that all the Golems have their out-inputs in the same format: compatibility.

Once in- and output are defined it seems natural to *analyze the mind in terms of conventional personality types*; we have used some crude types in 'THINK". See C. Brand (2002) for a catalogue of personality categorizations suggested in the psychological literature.

In Section 3.7 we discussed the decisive role of randomness in the study of human thinking. Actually, a more radical approach would be to think of ideas as *clouds of uncertainties* described by probability densities in a high dimensional feature space. The calculus of thoughts that we have discussed would then operate on probability densities, a bit similar to the role of wave functions in quantum mechanics. At the moment it is far from clear how to make this precise; some adventurous colleague may be tempted to look more closely into this problem.

13 Connect to the Brain?

So far we have avoided any reference to a neural substrate for thought, to wit, the brain. But since we have already started down the slippery slope of speculation, we can just as well continue with some unbaked ideas of how to relate GOLEM to actual human thinking.

Using fMRI, say that we equip a patient in the magnet with special glasses for visual inputs and with ear phones for auditory inputs. The sensory inputs should be chosen so that they can be represented as "thoughts" in GOLEM. We then obtain a series of low resolution scans $I^{\mathcal{D}} = \{I^{\mathcal{D}}(1)I^{\mathcal{D}}(2), ...I^{\mathcal{D}}(T)\}$ for the sensory inputs $thought(1), thought(2), ...thought(T)$. Using deformable template techniques, see Grenander (1993), we can relate the observed blobs that have lighted up to the various components of the brain. Thus we get mappings

$$I^{\mathcal{D}}(t) \to \gamma(t) \tag{65}$$

with the γ's representing collections of brain components; $\gamma(t) \in \Gamma$.

Then we are confronted with a statistical estimation problem of general regression type: Find approximate relations

$$thought(t) \approx \gamma()t) \tag{66}$$

To find such relations construct, for each t and i an arrow

$$g_i(t) \to \gamma(t) \tag{67}$$

for

$$thought(t) = \sigma(t)(g_1(t), g_2(t), ...g_i(t), ...) \tag{68}$$

one arrow for each brain component in $\gamma(t)$. This results in a *statistical map* $mind \to brain$. This map tells us how primitive ideas are related to activities in the various brain components, and if we find broad channels in it we have established a mind/brain relation.

To get better statistical stability we could replace the map $G \to \Gamma$ by $\mathcal{M} \to \Gamma$.

Can this experiment actually be caried out? We leave that question to researchers more familiar with brain research than the author.

14 How to Use the code

The MATLAB code is made available for download to computers with Windows operating system and the MATLAB system installed; see www.dam.brown.edu/ptg/publications. Once "mind05" has been downloaded to c:, the user should change directory to "mind05", and execute "think" to start thinking from scratch. Then menus are appearing for choosing options for GOLEM's thinking. The programs have been thoroughly debugged but cannot be guaranteed to be perfect. It was developed on MATLAB 14 but runs also under MATLAB 13.

On the same WEB site executable code can also be found for users without MATLAB in the form of a zipped archive. The the MCRInstaller should be run first, once is enough. Then change directory to "mind05" on the command prompt (DOS window) followed by "think.exe".

15 Not Yet Implemented

The following additions to GOLEM seem natural but have not yet been implemented.

1) One should allow a generator in a thought to be dominated by at most one generator for each modality. This is to avoid thoughts like $(small, big, house)$. An earlier version of GOLEM had this constraint realized but was later excluded.

2) The mind operations MUTATE, SPECIALIZE and CROSSOVER have not been included in the code. The two first ones can easily be implemented with minor changes in the existing cod, but CROSSOVER would require a little effort.

3) GOLEM does not (at present) delete new ideas when they are not reinforced by repeated occurrence. They should be deleted if ideas with the same *content* are not replicated enough.

4) GOLEM can perform *link analysis*. For a given set of generators (concepts) running the GOLEM as an interpolator it will discover links amd attach weights to them if desired. This could be of considerable practical use, to "connect the dots" to use a standard cliché.

5) The thinking simulated by GOLEM is fairly slow, in particular if the spped of the computer is less than 2 GHz. If one had access to parallel hardware it should be possible to achieve much better speed if each level in the configuration for building "thought" was treated at once. May we suggest that this is reminiscent to the columnar organization of the brain?

16 Acknowledgment.

I have benefited from innumerable discussions with Y. Tarnopolsky whose incisive comments and constructive criticism contributed much to this work. I have also listened carefully to David Mumfords skeptical remarks. Contributions to G have been made by Paj, Alexander, Ariana, Nikolas and Tatiana. Anuj Srivastave helped with the construction of GUI's.

REFERENCES

There is an enormous literature on mind theories, especially general, informal ones but also many mathematical/computational formalizations. Below we list only a small number of references that are directly related to the approach of this work.

J. Besag: Spatial interaction and the statistical analysis of lattice systems, J.R.S.S., 1974

C. Brand: www.cycad.com/cgi-bin/Brand/quotes/q03.html

D. E. Brown: Human Universals, McGraw-Hill, 1991

N. Chomsky: Syntactic Structures, Mouton, The Hague, 1957

A. R. Damasio: The Feeling of What Happens : Body and Emotion in the Making of Consciousness. Harcourt Brace and Comp., 1999

J.-L. Faulon: Automorphism Partitioning, and Canonical Labeling Can Be Solved in Polynomial- Time for Molecular Graphs, J. Chem. Inf. Comput. Sci., 1998

W. Feller: An Introduction to Probability Theory and its Applications, Volume I, 2nd ed., Wiley, 1957.

U. Grenander: Lectures on Pattern Theory. Regular Structures Vol. III, (1981), Springer.

U. Grenander: Gemeral Pattern Theory , Oxford University Press, 1993.

U. Grenander: Windows on the World, CD-Rom, 2001.

J. M. Hammersley and P. Clifford : Markov Fields on Finite Graphs and Lattices, preprint, University of California, Berkeley,1968.

I. Kant: Kritik der reinen Vernunft, Konigsberg, 1781.

G. Mack: Interdisziplinare Systemtheorie, Lecture University of Hamburg, 1998.

E. Mally: Grundgesetze des Sollens,1926.

B. Osborn: Parameter Estimation in Pattern Theory, Ph.D. thesis, Div. Appl. Math., Brown University, 1986

J. Pearl: Probabilistic Reasoning in Intelligent Systems,Morgan Kauffman, 1988

C.S. Peirce: On the Algebra of Logic; A Contribution to the Philosophy of Notation, American Journal of Mathematics, 1885.

M.R. Quillian: Semantic memory. Minsky, M., Ed. Semantic Information Processing. pp.216-270. Cambridge, Massachusetts, MIT Press, 1968.

R.C. Schank: Conceptual Information Processing, North-Holland, 1975

M. Tominaga, S.Miike,H.Uchida,T. Yokoi: Development of the EDR Concept Dictionary,Second Workshop on Japan-Unitged Kingdom Bilateral Cooperative Research Programme on Computational Linguistics, UMIST, 1991

L.S. Vygotskij: Thought and Language, Cambridge, MA, MIT, Press, 1962

J. B. Watson: Behavior: An Introduction to Comparative Psychology, 1914.

J.Weizenbaum: ELIZA - a computer program for the study of natural language communicationbetween man and machine. Communications of the ACM 9. 1966.

L. Wittgenstein: Tractatus Logicus-Philosophicus, Sixth Edition, London, 1955.

B. L. Whorf: Language, Thought, and Reality. Selected Writings of Benjamin Lee Whorf. Ed. J. B. Carroll. New York: MIT Press; London: John Wiley, 1956

R. Wille: Formal concept analysis. Electronic Notes in Discrete Mathematics, 2, 1999

G.H. von Wright: An essay in deontic logic, MIND, 1968

Y. Tarnopolsky(2003): Molecules and Thoughts: Pattern Complexity and Evolution in Chemical Systems and the Mind. Rep. Pattern Theory Group at www.dam.brown.edu/ptg

Y. Tarnopolsky (2004): users.ids.net/ yuri

APPENDIX 1: Some Famous Mind Theories

Let us take a brief look at a few of the innumerable earlier attempts and see how the ideas are related to the above discussion.

16.1 A Sample of Mind Theories

L.R.Goldberg: We need to develop a structural model, some kind of an overarching taxonomy to link individual differences so that we're not all speaking idiosyncratic tongues.

BUT

Paul Kline: The history of the psychology of personality, from Hippocrates onwards, is littered with the fragments of shattered typologies.

Here is a list of some attempts to represent human thought. It is of course highly incomplete and the items are included only as pointers to what we will discuss in the following sections. In spite of their different appearence they have elements in common with the rearch attidtude presented in this work. The analogies may not be very strong. A more convincing parallel is to chemistry, something that Tarnopolsky has pointed out in a very convincing way; the reader may wish to consult Tarnopolsky (2003). The belief propagating systems in Pearl (1988) uses similar probabilistic concepts but with a different aim.

16.1.1 Syllogisms.

Aristotle suggested syllogisms as guides for reasonong. Today it is difficult to see why they came to be considered to be so fundamental for thinking, but they were for a couple of thousand years, and innocent school children (including this author) were forced to memorize the possible syllogisms. Here is one of them

If all B's are A,
and all C's are B's,
then all C's are A.

Note the occurence of the *variables* A,B, and C. They make the statement more general than would be a single instance of it, for example
all humans are mortal
all Greeks are human
then all Greeks are mortal
which is the special instance with A= "mortal", B= "human", C= "Greek".

16.1.2 Formal Logics.

Of greater interest is Boolean logic like $x \vee (y \wedge z)$, or in words "x or both y and z". Again, this is a generalization of $big \vee (little \wedge red)$. Another is predicate calculus, for example $\forall x(Ax \supset Bx)$, or in words "for all x it is true that if x is an A then x is a B". We want to mention that C.S. Peirce (1885), always original, actually used what is essentially graphs to represent some human thoughts; he called them existential graphs.

Predicate calculus presumes Aristotelian syllogisms but is more powerful. Still more powerful logical systems of this type exist, but they have in common that they represent *exact thoughts*: the statements are true or false (at least this is the intention but caution is needed here) but less exact thinking is not represented by these systems. For example emotional thinking is not dealt with although this may actually be of greater human relevance for everyday use than exact reasoning. However, some philosophers have gone outside the classical domain of logical thought; as examples we mention Mally(1926) and von Wright (1968) and their studies of deontic logic

16.1.3 Psychoanalysis.

Emotional thinking is described by psychoanlysis as introduced by Siegmund Freud. Less formal than the above systems, this theory tries to understand the human mind in terms of elements: id, ego, superego, censor, libido, castration fear, child sexuality, transfer, repression, Oidipus complex... They are combined to form the nucleus of the mind of the patient, or at least the subconscious part of it, and are supposed to be discovered by the analyst through examination of dreams, slips, free associations and other expressions of the subconscious.

Among the many deviant practitioners of the psychoanalytic faith, Alfred Adler is one of the less exotic ones, actually representing more common sense than the other apostles. His "individual psychology" rejects Freud's original theories that mental disturbances were caused by sexual trauma, often in childhood, and he opposed the generalizations when dreams were interpreted, in most instances, as sexual wish fulfillment. Instead he used as his basic elements of mind feelings of inferiority, striving for power and domination, and wanted to understand mental activities as goal driven.

Posterity has not been kind to Freudian psychoanalytic theory, but it constitutes at least an audacious and admirable attempt to understand the human mind by representing them in terms of simple constituents. We also share this goal, but shall use more elemental units for building flexible models of thought.

16.1.4 Semantic Networks

. The idea of semantic networks has been very popular in the AI community since its introduction in Quillian (1968). Such schemes are knowledge representation with nodes and directed connections between nodes. The nodes represent objects or concepts and the connections mean relations between nodes. A special case is the Petri net that has been suggested as a model of computation. Among other graph based attempts we mention conceptual analysis, Wille (1999), and concept classification, Schanks (1975), Tominaga, Miike,Uchida, Yokoi (1991). A very ambitious attempt using objects and arrows can be found in Mack (1998).

We shall also use digraphs in our knowledge representations, but augmented in pattern theoretic terms, with not only generators and connectors, but also bondvalues, connection types, prior probability measures as well as algebraic operations on "thoughts". The semantic network was certainly a promising idea but interest in it seems to have waned in recent years. This may be due to the lack of specific structure in some of the work on semantic networks.

16.1.5 Formal Grammars

Following Chomsky (1957) many formal grammars have been suggested as models for human languages, for example context free grammars. They also use graphs, for example TREES, to generate the linguistic structures, but were intended to explicate language rather than thought. Among the systems mentioned here this one is closest in nature if not in details to the approach of this work and this applies also to the current linguistic program Principles and Parameters.

16.1.6 Associations.

Behaviorism claims that human behavior can be explained in terms of stimulus-response associations, and that they are controlled by reinforcement. J. B. Watson described this approach in an influential book 1914 about human behavior. Mental terms like goal, desire, and will were excluded. Instead it used as building blocks the associations formed by repeated stimulated actions introducing couplings between input and output.

We shall also apply a compositional view, *but with many and very simple mental building blocks that represent extremely simple ideas*. They will be chosen as what seems to be natural and common sense entities in human thought, close to everyday life. Our choice of units is admittedly subjective but not wholly so. Indeed, we have been encouraged by the discussion of *human universals* in Brown (1991, who advocates the existence of universals organized into specific lists.

APPENDIX 1: Consistency of Probability Measure

For the definiton (13) to make sense as probabilities (normalized) we must have

$$Z(T) = \sum_{c \in \mathcal{C}(\mathcal{R})} \kappa_n \frac{1}{n!} \prod_{i=1}^{n} Q(g_i) \prod_{(k,k\prime) \in \sigma} A^{1/T}[b_j(g_i), b_{j\prime}(g_{i\prime})] < \infty \tag{69}$$

This is similar to the condition for the probability measure over a stochastic CF language to be non-defective, see GPT 8.1.2. The above sum can be written as

$$\sum_{n=1}^{\infty} \kappa_n \sum_{c \in \mathcal{C}_n(\mathcal{R})} \frac{1}{n!} \prod_{i=1}^{n} Q(g_i) \prod_{(k,k\prime) \in \sigma} A^{1/T}[b_j(g_i), b_{j\prime}(g_{i\prime})] \tag{70}$$

where $\mathcal{C}(\mathcal{R})$ consists of all regular configurations of the mind. If the maximum arity is ω_{max}, the cardinality of σ_n is bounded by

$$|\sigma| \leq (n\omega_{max})^n \tag{71}$$

so that the above sum is bounded by

$$\sum_{n=1}^{\infty} \kappa_n \sum_{c \in \mathcal{C}_n(\mathcal{R})} \frac{1}{n!} \prod_{i=1}^{n} Q(g_i) \prod_{(k,k\prime)\in\sigma} A^{1/T}[b_j(g_i), b_{j\prime}(g_{i\prime})] \leq \sum_{n=1}^{\infty} \kappa_n (n\omega_{max})^n \frac{1}{n!} Q_{max}^n A_{max}^{n\omega_{max}} \tag{72}$$

In order that this series converge it is sufficient to ask that

$$\kappa_n = O(\rho^n); \rho < \frac{1}{e\omega_{max} Q_{max} A_{max}^{\omega_{max}}} \tag{73}$$

Indeed, this follows from the classical Stirling formula

$$n! \asymp \sqrt{2\pi n}(\frac{n}{e})^n \tag{74}$$

which implies that the terms in the sum are dominated by those of a geometric series with ratio less than one if (19) is satisfied.

This means that we have the

PRPOSITION. *The probability measure is well defined if the combinatorial complexity of the mind is bounded by (66): the probability of large configurations representing complicated mental modes most be small enough.*

Otherwise the mind would expand indefinitely, taking on more and more complicated states, leading to a mental explosion.

We shall use the notation $\pi_n = \kappa_n/n!$ which describes the probabilities of the size of content(c). It should be noticed that (19) is satisfied with $\pi_n = Poisson_n(\mu)$, a Poisson distribution with mean $\rho = \mu$. It is not clear if this can be motivated by an underlying Poisson process in the MIND.

NOTE: In terms of Gibbsian thermodynamics the above is not the canonical ensemble. Indeed, the number of interacting elements is not fixed but random and variable. Thus we are dealing with Gibbs' *grand* canonical ensemble.

APPENDIX 3: A MODALITY LATTICE

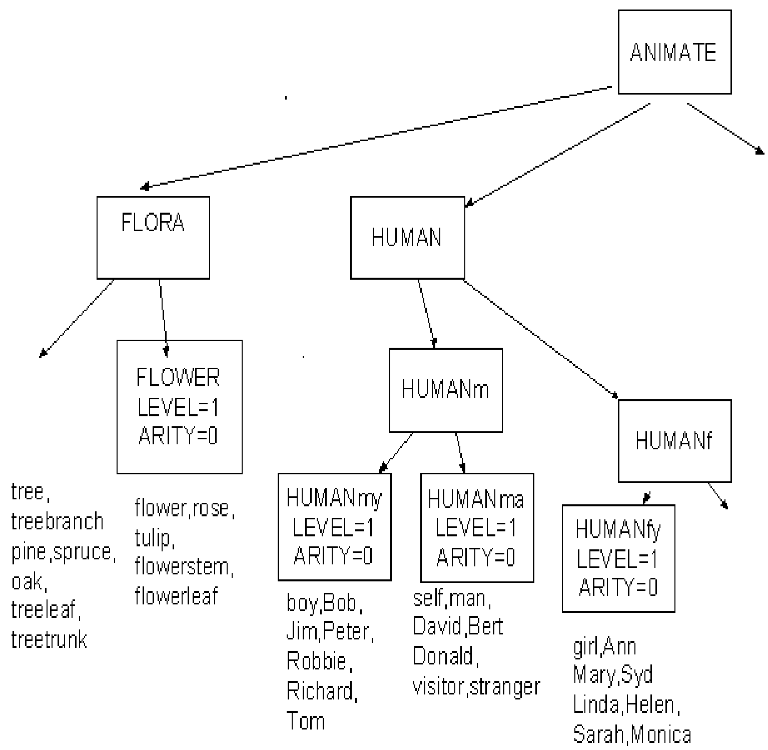
To display the modality lattice $\mathcal{M}$ we have split it up into 30 parts, one each on a separate pages. The pages are organized as in

COORDINATE SYSTEM FOR MODALITY LATTICE

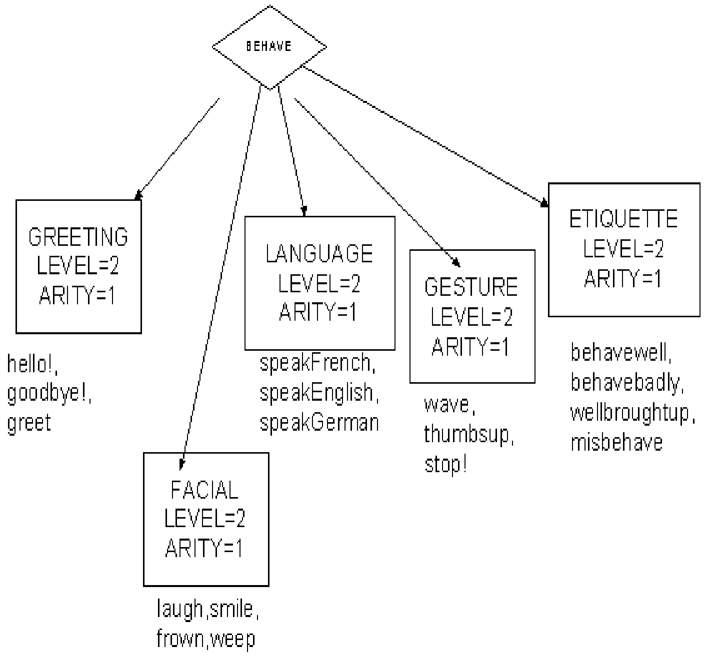
(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)

The coordinates in this diagram correspond to the ones on the top left of each page. Rectangles stand for modalities and diamond shapes to unions of mdalities that do not form modalities themselves. Primitive ideas are shown under the rectangles.

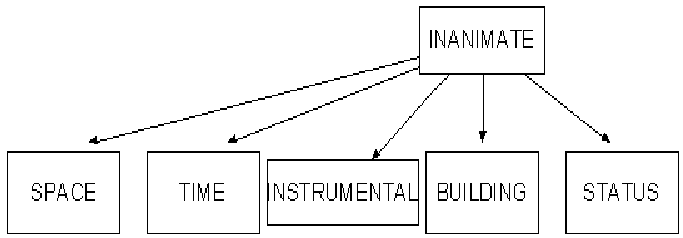
The modlaity lattice is too big to show in its entirety. Instead we show parts of it. The modality ANIMATE


ANIMATE
FLORA
HUMAN
FLOWER
LEVEL=1
ARITY=0
HUMANm
HUMANf
tree,
treebranch
pine,spruce,
oak,
treeleaf,
treetrunk
flower,rose,
tulip,
flowerstem,
flowerleaf
HUMANmy
LEVEL=1
ARITY=0
HUMANma
LEVEL=1
ARITY=0
HUMANfy
LEVEL=1
ARITY=0
boy,Bob,
Jim,Peter,
Robbie,
Richard,
Tom
self,man,
David,Bert
Donald,
visitor,stranger
girl,Ann
Mary,Syd
Linda,Helen,
Sarah,Monica

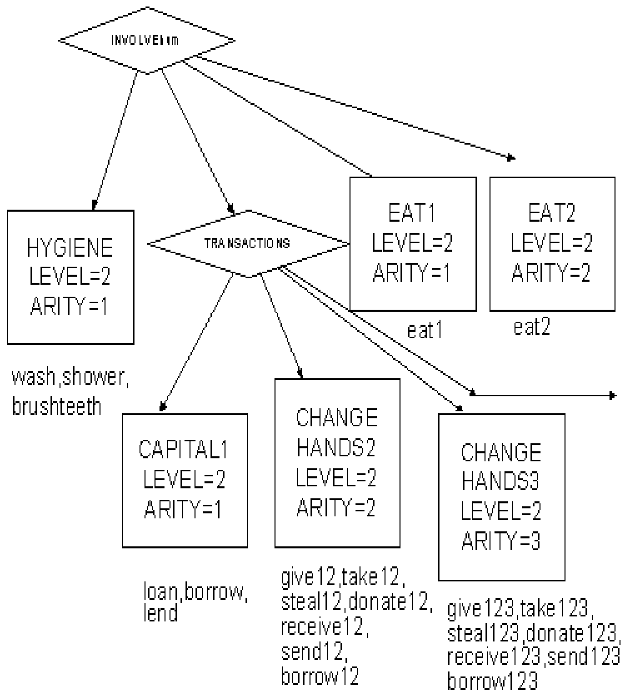
and BEHAVE



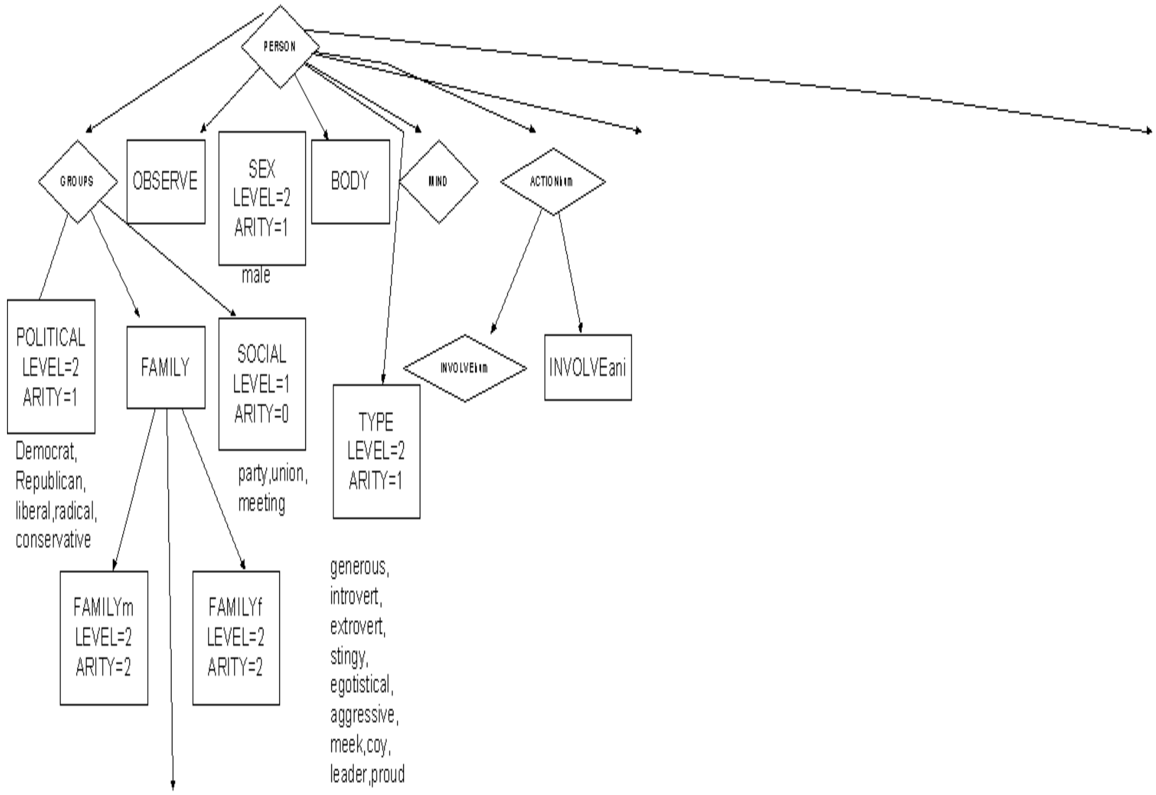
Note that BEHAVIOR is not a modality but is broken up into modalities. And INANIMATE



and the non-modality INVOLVEhum



Finally PERSON is shown only in part



APPENDIX 4: MATLAB Code

Executing GOLEM calls a number of functions, first of all the main function "think":

MAIN FUNCTION

The oputput to "think" is of the form [content,connector]. To start it from scratch execute The function loads a file "mind \ data" containing a generator space, the modality lattice and much else; it should be placed in c:\ mind _data. The code is complicated, but the reader is recommended to glance, at least briefly, on it to see what programming strategy has been applied.

```
 function think
%creates complete "thought" and displays 2-idea if there is one in thought
%set seed forrandomness
rand('state',sum(100*clock));
c=menu('CHOOSE A MIND OPERATION','THINKING DRIVEN BY THEME','CONTINUOUS THOUGHT','THINKING D
    'FREE ASSOCIATIONS','SET PERSONALITY PROFILE','SET MIND LINKAGES','THE VISIBLE MIND','SE
switch c
```

The first case implements thinking in *themes*; it is one of the most important options:

```
    case 1
[content,connector]=think1;
hold on
load c:\mind_data
%is there a 2-idea?
cont=content(:,2);
'X'
mods=g_mod(cont)
gs= ismember(mods,180);
if any(gs)
    see_mind(content,connector)
    hold on
    blinktxt(.6,.7,'NOTE ABSTRACT IDEA')
    hold on
    pause(4)
    figure('Units','Normalized','Position',[0 0 1 1])
    axis off
    a=menu('ANALYZE IDEA ?','YES','NO')
    if a==1
        close all
        ind=find(gs);idea_generator=cont(ind(1));idea_generator=G(idea_generator);
        idea_name=idea_generator.name;
        number=name_2_number(idea_name);
        idea_content=CREATION{1,number,1};idea_connector=CREATION{1,number,2};
        see_mind(idea_content,idea_connector)
        N=radix2num(idea_content(:,2),r)
        text(.1,.7,['IDEA WITH GOEDEL NUMBER ',num2str(N)],'FontSize',30,'Color','b')
        pause
    end
    close all
    b=menu('APPLY ABSTRACTION OPERATOR TO IDEA ?','YES','NO')
    if b==1
```

```
        see_mind_mod(idea_content,idea_connector)
        pause
    end
end
c=clock;c=rem(c(5),5);
if c ==0
    [Q,A]=memory(content,connector);
    close all
    clf
    'Y'
figure('Units','Normalized','Position',[0 0 1 1])
axis off
text(.2, .2, ['STRENGTH OF MIND LINKAGES UPDATED'],'Fontsize',20','Color','b')
pause(1)
end
close all
```

The next case is more complicated. It deals with thinking where the trajectory jumps from one theme to another repeatedly ans sometimes creates new ideas:

```
case 2
    load('C:\mind_data');
    %figure('Units','Normalized','Position',[0 0 1 1])
    %axis off
    clf
    answer=questdlg('MORE CONTINUOUS THOUGHT ?', 'YES','NO');
    if answer==2
        return
    end
    duration=menu(['HOW MANY SECONDS OF CONTINUOUS THOUHT  ? '],'10','20','30','40');
    duration=duration*10;%duration=str2num(duration)
    t0=clock;genre_old=1;


      while etime(clock,t0)<duration
            genre=select(ones(1,9)./9)
            if ~(genre==genre_old)
                figure('Units','Normalized','Position',[0 0 1 1])
                axis off
                clf
                text(.01, .5,['MIND TRAJECTORY CHANGES DIRECTION'],'FontSize',26,'Color','y
                axis off
                pause(.6)
            else
```

```
     end
      content=[];load c:\mind_data G
      %create thought germ "content,connector"
      [content,connector]=think2(genre);
      [content,connector]=add_generator_up_Q(content,connector,genre);
[content,connector]=add_generator_up_Q(content,connector,genre);
      %[content,connector,Q_theme]=build_thought_mod(genre);
      %see_mind_germ(content,[])
      pause(3)
      close all
      w=[];
      if isempty(content)
          figure('Units','Normalized','Position',[0 0 1 1])
           text(.2,.1 ,['EMPTY MIND'],'Color','r','FontSize',20)
           axis off
           pause(1)
      else
      v=content(:,2);n_v=length(v);
      k=1:n_v
      g=G(v(k));
      w=[w,g.level];
      if all(ismember(w,1))
           figure('Units','Normalized','Position',[0 0 1 1])
           text(.2,.1 ,['STOP THINKING!   NO OPEN BONDS!'],'Color','r','FontSize',20)
           axis off
           pause(1)
      end
      end
      %is any down bond open?
      found=1;
      while found==1
         [i,h,omega,found]=find_open_down_bond(content,connector);
         if found==0
                 see_mind(content,connector)
                 pause(1)
                 close all
                 %return here?
          else
                 [content,connector,found]=connect_down_bond(content,connector, i,h,o
                 see_mind(content,connector)
                 pause(1)
          end

       end
       see_mind(content,connector);
       pause(1.6)
```

```
            close
        [content,connector]=add_generator_up_Q(content,connector,genre);
        [content,connector]=add_generator_up_Q(content,connector,genre);
         see_mind(content,connector);
               pause(1.6)
               close

               [content,connector]=dom_thought(content,connector);
               see_mind_dom(content,connector);
               pause(3)
               genre_old=genre;
               close all
     end
 %now detect top_2ideas
   [top_2ideas_g,top_2ideas_h]=get_top_2ideas(content,connector); %these are the top_2ideas
   n_ideas=length(top_2ideas_g);
   ns=zeros(1,n_ideas);
   if n_ideas ==0
       figure('Units','Normalized','Position',[0 0 1 1])
       axis off
       text(.2,.8,'No Conscious Thought','FontSize',32)
       text(.8,.1,['Press Enter to Continue'],'FontSize',8)
       %pause
       return
   end

   for t=1:n_ideas
        gs=top_2ideas_g{1,t,:}; ns(t)=length(gs);
   end
   [Y,I]=max(ns);
   m=I(1);
   hs=top_2ideas_h{1,m,:};gs=top_2ideas_g{1,m,:};
   content1(:,1)=hs';content1(:,2)=gs';n=length(hs);connector1=[];
   for k1=1:n
       for k2=1:n
           for j=1:3
               h1=hs(k1);h2=hs(k2);g1=gs(k1);g2=gs(k2);
           segment=(connector(:,1)==h1)&(connector(:,2)==h2)&(connector(:,3)==j);
                   if any(segment)&(g1~=g2)
                       connector1=[connector1;[h1,h2,j]];
                       else
                   end
            end
       end
   end
   %add new idea to "G"
```

```
    r=length(G);n_new_ideas=length(gs_in_mod{180});%note numbering of "new ideas " modality
    G(r+1).name=['<idea',num2str(n_new_ideas+1),'>'];
    G(r+1).level=1;
    G(r+1).modality=180;
g_mod=[g_mod,180];x=size(CREATION);
    n_new_idea=x(2);
    CREATION{1,n_new_idea+1,1}=content1;
    CREATION{1,n_new_idea+1,2}=connector1;
    Q=[Q,1];A_new=zeros(r+1);A_new(1:r,1:r)=A;A_new(r+1,:)=ones(1,r+1);A_new(:,r+1)=ones(r+1
    figure('Units','Normalized','Position',[0 0 1 1])
    axis off
    text(.2,.8,'New Idea Created !','FontSize',32)
    text(.5,.1,['Press Enter to Continue'],'FontSize',20)
    %pause
    [L1,L2,L3,L4]=get_levels(G);
    clear content connector omega genre theme
    clear content1 connector1
    save c:\mind_data
```

The third case accepts inputs from the external world and learns from experience by updating "Q" and "A":

```
case 3
    %get input from external world:
%carries out inference from inputted thought
load c:\mind_data
external_world=sensory;
l_external=length(external_world);connector=[];content=[];
%now start to build internal MIND as configuration
content_col2=[];connector1=[];l=0;
for nu=1:l_external
   sub=external_world{nu};
   l_sub=length(sub(:,1));content1=zeros(l_sub,2);connector1=[];
   content1(:,1)=[l+1:l+l_sub]';content1(:,2)=sub(:,2);
   [content1,connector1]=add_connector_new(content1,connector1);
   connector=[connector;connector1];
   content_col2=[content_col2,sub(:,2)'];
   l=l+l_sub;
end
   l_scene=length(content_col2);
   content=zeros(l_scene,2);content(:,1)=[1:l_scene]';content(:,2)=content_col2';
   see_mind(content,connector)
   %print -dbitmap 'c:\mind_figures\mind_fig1'
   pause(3)
```

```
   close
   v=content(:,2);n_v=length(v);w=[];
for k=1:n_v
g=G(v(k));
w=[w,g.level]
end
if all(ismember(w,1))
    figure('Units','Normalized','Position',[0 0 1 1])
    text(.2,.1 ,['STOP THINKING!   NO OPEN BONDS!'],'Color','r','FontSize',32)
    axis off
    pause(1)
    return
end

   figure('Units','Normalized','Position',[0 0 1 1])
   axis off
   text(0,.5,['Input complete. Press Enter to continue and wait...'],'FontSize',22)
   pause
   close all
for iter=1:3
   [content,connector]=add_generator_up(content,connector);
   [content,connector]=add_generator_down(content,connector);
end

%is any down bond open?
found=1;
while found==1
    [i,h,omega,found]=find_open_down_bond(content,connector);
    if found==0
            see_mind(content,connector)
            pause(1)
            close all
            %return
    elseif found==1
       Q(gs)=20;Q_theme=Q;
       [content,connector,found]=connect_down_bond(content,connector, i,h,omega,Q_theme);
       see_mind(content,connector)
       pause(1)
    end
end
'x'
see_mind_infer(content,connector)
%pause
   close all
   [Q,A]=memory(content,connector);
```

```
close all
```

In case 4 the thinking is not controlled by either external inputs nor by thematic restrictions. The result is very chaotic thoughts

```
case 4
    %free associations
figure('Units','Normalized','Position',[0 0 1 1])
   load('C:\mind_data');
   text(.2,.5,['WAIT...'],'FontSize',32)
   axis off
   pause(1);content=[],connector=[];
   n_input=0;
   sto=1;
   while sto==1
   for iter=1:3
      [content,connector]=add_generator_new(content,connector);
   end
   see_mind(content,connector)
   text(.1,.98,'CHAOTIC THINKING...','Fontsize',20,'Color','y')
pause(1)
close
for iter=1:4
   [content,connector]=add_generator_up(content,connector);
   [content,connector]=add_generator_down(content,connector);
see_mind(content,connector)
text(.1,.98,'CHAOTIC THINKING...','Fontsize',20,'Color','y')
pause(1)
end
for iter=1:1
[content,connector]=delete_generator_connections(content,connector);
end
pause(1)
close
[content,connector]=see_mind_dom(content,connector)
text(.1,.98,'CHAOTIC THINKING...','Fontsize',20,'Color','y')
%print -dbitmap 'c:\mind_figures\mind_fig1'
hold on
text(.2,.05,'Press ENTER to continue', 'FontSize',12)
hold off
pause
close all
   figure('Units','Normalized','Position',[0 0 1 1])
   axis off
```

```
   q=menu('CONCENTRATED THOUGHT ? HARD THINKING, TAKES TIME...WAIT...', 'YES','NO');
if q==1
  [content,connector]=add_connector_new(content,connector) %note;
   see_mind(content,connector)
   %print -dbitmap 'c:\mind_figures\mind_fig2'
   hold on
text(.2,.05,'Press ENTER to continue', 'FontSize',12)
pause
   close
end
   figure('Units','Normalized','Position',[0 0 1 1])
   axis off
   p=menu('CONTINUE WITH FREE ASSOCIATIONS ?', 'YES','NO');
   if p==2
       sto=2;
  see_mind(content,connector)
  hold on
text(.2,.05,'Press ENTER to continue', 'FontSize',12)
hold off
pause
close all
    end
end
[Q,A]=memory(content,connector);
figure('Units','Normalized','Position',[0 0 1 1])
axis off
text(.1, .5, ['MIND LINKAGES UPDATED: FORGET AND REMEMBER'],'Fontsize',20','Color','b')
pause(1)
close all
```

Next case lets the user define a personality profile for "self":

```
case 5
    set_personality
```

Case 6 implements the personality profile by changing "Q" and "A":

```
case 6
    load c:\new
    load c:\mind_data
    %personality_behavior are sets of g's
    %first set Q's
   % Q(greedy)=(1-val1)*3;
```

```
   % Q(generous)=val1*3;
   %  Q(scholastic)=(1-val2)*3;
   % Q(athletic)=val2*3;
   %  Q(aggressive)=(1-val3)*3;
   % Q(mild)=val3*3;
   %  Q(selfish)=(1-val4)*3;
   % Q(altruistic)=val4*3;
    %then set A's
    r=length(G);
for g=1:r
    if strcmp(G(g).name,'self')
        sel=g;
    end
end
 A(greedy,sel)=(1-val1)*3;
    A(generous,sel)=val1*3;
     A(scholastic,sel)=(1-val2)*3;
    A(athletic,sel)=val2*3;
     A(aggressive,sel)=(1-val3)*3;
    A(mild,sel)=val3*3;
     A(selfish,sel)=(1-val4)*3;
    A(altruistic,sel)=val4*3;
    %symmetrize
    A=(A+A')./2;
    save c:\mind_data
    figure('Units','Normalized','Position',[0 0 1 1])
    axis off
    text(.1,.9,'STRENGTH OF MINd LINKAGES SET TO: ','Color','y','Fontsize',28)
    text(.1,.8,['greedy: ',num2str(1-val1)],'Color','y','Fontsize',28)
     text(.1,.7,['generous: ',num2str(val1)],'Color','y','Fontsize',28)
     text(.1,.6,['scholastic: ',num2str(1-val2)],'Color','y','Fontsize',28)
      text(.1,.5,['athletic: ',num2str(val2)],'Color','y','Fontsize',28)
      text(.1,.4,['aggressive: ',num2str(1-val3)],'Color','y','Fontsize',28)
      text(.1,.3,['mild: ',num2str(val3)],'Color','y','Fontsize',28)
      text(.1,.2,['selfish: ',num2str(1-val4)],'Color','y','Fontsize',28)
      text(.1,.1,['altruistic: ',num2str(val4)],'Color','y','Fontsize',28)
    pause
```

In case 7 the MIND is displayed as connections between elementary ideas situated on the circumference of a circle. Note the idea "self":

```
    case 7
        %display "A" linkages
        load c:\mind_data G A r
```

```
angles=2*pi.*[0:r-1]./r;
xs=cos(angles);ys=sin(angles);
figure('Units','Normalized','Position',[0 0 1 1])
axis off
text(.3,.8, 'VISIBLE MIND','Fontsize', 25,'Color','r')
text(.3,.6, 'LOCATION OF "SELF" INDICATED BY *','Fontsize', 25,'Color','r')
text(.3,.4, 'WAIT !','Fontsize', 25,'Color','r')
text(.3,.2, 'TAKES A WHILE...','Fontsize', 25,'Color','r')
pause(2)
close all
clf
figure('Units','Normalized','Position',[0 0 1 1])
text(-1.5,1.1,'SITES OF ELEMENTARY IDEAS ON THE CIRCUMFERENCE','Fontsize', 25,'Color
hold on
for g1=1:5:r-1
   for g2=g1+1:5:r
    if (A(g1,g2)>.5)
        plot([xs(g1),xs(g2)],[ys(g1),ys(g2)])
        axis off
        axis equal
        hold on
    end
   end
   %find "self"
end
for g=1:r
   if strcmp(G(g).name,'self')
      sel=g;
end
end
hold on
plot(xs(g),ys(g),'*r')
%pause
%end of "think":
```

Case 8 lets the user display the configuration diagrams of created ideas:

```
case 8
    load c:\mind_data
 figure('Units','Normalized','Position',[0 0 1 1])
 axis off
 clf
 text(.1,.9,'NUMBRER OF CREATED IDEAS :','FontSize',26)
 siz=size(CREATION);
```

```
    axis off
    text(.1, .8,num2str(siz(2)),'FontSize',26)
    %text(.1,.6,'Select <idea> number','FontSize',26)
    axis off
   hold on
    number=inputdlg('Enter <idea> number ')
    number=str2double(number)
    hold off
    content2=CREATION{1,number,1};connector2=CREATION{1,number,2};
    %content=content2
    %connector=connector2
    %see_mind_new(content2,connector2,number)
    hold on
    idea_content=CREATION{1,number,1};idea_connector=CREATION{1,number,2};
        see_mind_mod(idea_content,idea_connector)
        N=radix2num(idea_content(:,2),r)
        text(.1,.7,['IDEA WITH GOEDEL NUMBER ',num2str(N)],'FontSize',30,'Color','b')
        pause
end
```

The DEVELOP option takes a long time to execute.

```
 case 9
        load('C:\mind_data');
        A_old=A;
        close all
    clf
    duration=menu(['HOW MANY HOURS OF DEVELOPMENT  ? '],'1','2','3','4');
    duration=duration*12;%change 12 to 3600
    t0=clock;genre_old=1;
      while etime(clock,t0)<duration
            genre=select(ones(1,9)./9);

             content=[];
             %create thought germ "content,connector"
             [content,connector]=think2(genre);
             [content,connector]=add_generator_up_Q(content,connector,genre);
       [content,connector]=add_generator_up_Q(content,connector,genre);
             w=[];
             if isempty(content)

             else
             v=content(:,2);n_v=length(v);
             k=1:n_v;
             g=G(v(k));
```

```
          w=[w,g.level];
          if all(ismember(w,1))

          end
          end
          %is any down bond open?
          found=1;
          while found==1
             [i,h,omega,found]=find_open_down_bond(content,connector);
             if found==0
              else
                     [content,connector,found]=connect_down_bond(content,connector, i,h,o
              end

          end
           close
     [content,connector]=add_generator_up_Q(content,connector,genre);
     [content,connector]=add_generator_up_Q(content,connector,genre);
        genre_old=genre;
    end

clear content connector omega genre theme
clear content1 connector1
save c:\mind_data
A_new=A;
 angles=2*pi.*[0:r-1]./r;
    xs=cos(angles);ys=sin(angles);
     figure('Units','Normalized','Position',[0 0 1 1])
subplot(1,2,1),text(-1.5,1.1,'BEFORE...','Fontsize', 25,'Color','r')


    hold on
    for g1=1:5:r-1
       for g2=g1+1:5:r
        if (A_old(g1,g2)>.5)
            plot([xs(g1),xs(g2)],[ys(g1),ys(g2)])
            axis off
            axis equal
            hold on
        end
       end
       %find "self"
    end
    for g=1:r
       if strcmp(G(g).name,'self')
          sel=g;
```

```
        end
        end
        hold on
        plot(xs(g),ys(g),'*r')

         subplot(1,2,2),text(-1.5,1.1,'...AND AFTER','Fontsize', 25,'Color','r')
         %figure('Units','Normalized','Position',[0 0 1 1])

        hold on
        for g1=1:5:r-1
           for g2=g1+1:5:r
            if (A_new(g1,g2)>.5)
                plot([xs(g1),xs(g2)],[ys(g1),ys(g2)])
                axis off
                axis equal
                hold on
            end
           end
           %find "self"
        end
        for g=1:r
           if strcmp(G(g).name,'self')
              sel=g;
        end
        end
        hold on
        plot(xs(g),ys(g),'*r')

        pause


close all
```

The primary function "think" calls a secondary function "think1" that grows a mind germ and then applies out the COMPLETION operation to it:

```
function [content,connector]=think1
%simulates GOLEM for given theme of thoughts
content=[];load c:\mind_data
%create thought germ "content,connector"
[content,connector,Q_theme]=build_thought;
see_mind_germ(content,[])
pause(3)
```

```
close all
w=[];
v=content(:,2);n_v=length(v);
k=1:n_v
g=G(v(k));
w=[w,g.level];
ismember(w,1);
if all(ismember(w,1))
    figure('Units','Normalized','Position',[0 0 1 1])
    text(.2,.1 ,['STOP THINKING!   NO OPEN BONDS!'],'Color','r','FontSize',20)
    axis off
    pause(1)
    return
end

%is any down bond open?
found=1;
while found==1
    [i,h,omega,found]=find_open_down_bond(content,connector);%_mod?

    if found==0
        'not found'
            see_mind(content,connector)

            pause(1)
            close all
            return
    elseif found==1
        'found'
       [content,connector,found]=connect_down_bond(content,connector, i,h,omega,Q_theme);
       see_mind(content,connector)
       pause(1)
    end

end
```

....

SIMPLE MOVES

Amomg the simple moves isadding a connector

```
function [content,connector]=add_connector_new(content,connector)
%differs from "add_g" in that conntent is not changed
load('C:\mind_data');
if isempty(content)
```

```
  return
   else
n=length(content(:,1));
for i1=1:n
   for i2=1:n
   if isempty(connector)
      connector=[1,1,1];%this cludge to avoid error
   else
       h1=content(i1,1);h2=content(i2,1);g1=content(i1,2);g2=content(i2,2);
      level1=G(g1).level;level2=G(g2).level;
      if level1==level2+1
        for j=1:3
            is_old=any((connector(:,1)==h1)&(connector(:,2)==h2));
            is_old=is_old|any((connector(:,1)==h1)&(connector(:,3)==j));
            reg=connection_regular_new(i1,i2,j,content,connector,g_mod,mod_transfer);
            answer=(~is_old)&(g1~=g2)&(h1~=h2)&reg;
                if answer
                    connector=[connector;[h1,h2,j]];
                end
        end
        end

    end
end
end

end
```

Similarly the functions *add_generator_down* and *add_generator_down_Q* add new generators downwards. The qualifier "Q" here indicates that the theme driven "Q" vector should be used.

```
function [content,connector]=add_generator_down_Q(content,connector,theme)
%executes theme driven associations, downwards ideas
%NOTE: "connection_regular_new" has not yet been included
load('C:\mind_data');
gs=set_gs_in_mods(theme,gs_in_mod);

Q(gs)=20;
if isempty(content)
   Q=Q./sum(Q);g=select(Q);
   content=[1,g];
   return
else
```

```
   %select one of the gens in "content"
   n=length(content(:,1));i=select(ones(1,n)./n);
   g=content(i,2);h=content(i,1);
   mod=g_mod(g);
   to_g_downs=[gs_in_mod{mod_transfer{mod,1}},gs_in_mod{mod_transfer{mod,2}},...
         gs_in_mod{mod_transfer{mod,3}}];

   %now try to connect down to each of these gens
   probs=[];
   if isempty(to_g_downs)
      return
   else
   end

      n_to_g_downs=length(to_g_downs);
   for nu=1:n_to_g_downs
      prob=Q(to_g_downs(nu))*mu/(n+1);prob= prob*A(g,to_g_downs(nu))^(1/T);probs=[probs,prob
   end
   probs=[probs,1];
   probs=probs./sum(probs);
   nu=select(probs);
   %n_to_g_downs;
   if nu==n_to_g_downs+1
      return
      end
      g_to=to_g_downs(nu);
      new_h=max(content(:,1))+1;

  content=[content;[new_h,g_to]];

mod1=g_mod(g_to);
if ~isempty(connector)
    for j=1:3
        is_old=any((connector(:,1)==h)&(connector(:,2)==new_h));
            is_old=is_old|any((connector(:,1)==h)&(connector(:,3)==j));
     if (~is_old)&ismember(mod1,mod_transfer{mod,j});
       connector=[connector;[h,new_h,j]];

   else
   end
    end
else

end
end
```

```
function [content,connector]=add_generator_up_Q(content,connector,theme)
%executes theme driven thinking upwards ideas
load('C:\mind_data');
gs=set_gs_in_mods(theme,gs_in_mod);
Q(gs)=20;
if isempty(content)
   Q=Q./sum(Q);g=select(Q);
   content=[1,g];
else
   %select one of the gens in "content"
   n=length(content(:,1));i=select([1:n]./n);h=content(i,1);g=content(i,2);...
      mod=g_mod(g);
   mod_ups=mod_transfer_inv{mod};
   n_mod_ups=length(mod_ups);
   to_g_ups=[];
   %find generators up from which connection may be created
   for m=1:n_mod_ups
      to_g_ups=[to_g_ups,gs_in_mod{mod_ups(m)}];
   end
   %now try to connect up to each of these gens
   n_to_g_ups=length(to_g_ups);
   probs=[];
   if isempty(to_g_ups)
      return
   else
   end

   for nu=1:n_to_g_ups
      prob=Q(to_g_ups(nu))*mu/(n+1);prob= prob*A(g,to_g_ups(nu))^(1/T);probs=[probs,prob];
   end
   probs=probs./sum(probs);probs=[probs,1];
   nu=select(probs);
   if nu==n_to_g_ups+1
      return
   end
   new_h=max(content(:,1))+1;
      g_to=to_g_ups(nu);
      mod1=g_mod(g_to);
         for j=1:3
            h=content(i,1);
            if isempty(connector)
               connector=[connector;[new_h,h,j]]
            else
            is_old=any((connector(:,1)==new_h)&(connector(:,2)==h));
            is_old=is_old|any((connector(:,1)==new_h)&(connector(:,3)==j));
             if (~is_old)&ismember(mod,mod_transfer{mod1,j});
```

```
                 connector=[connector;[new_h,h,j]];
               end
            end
          end
   content=[content;[new_h,g_to]];
end
```

Thought germ is created by "build_thought"

```
function [content,connector,Q_theme]=build_thought
% computes new thought from scratch (enpty "content") according to PRINIPLES
%executes theme driven associations
%NOTE: "connection_regular_new" has not yet been included
load C:\mind_data ;

%find gnerators in various levels
[L1,L2,L3,L4]=get_levels(G);

%select theme
number=menu('Select Theme of Mind','To Have and Have Not','Love and Hate',...
        'Sport','Business','Study','Health','Pets','Conversation','Politics');
    theme=THEMES{1,number,:};
%find generators in "theme"
gs=set_gs_in_mods(theme,gs_in_mod);content=[];connector=[];
Q(gs)=20;Q_theme=Q;

%thinking power defined in terms of size of "thought_germ"
prob_germ1=1./[1:4];prob_germ1=prob_germ1./sum(prob_germ1);
n_germ1=select(prob_germ1);


 %form sample of size "n_germ" on level 1
 level = 1;
 gs1=intersect(gs,L1);
 sample1=[];Q1=Q(gs1);sampl1=[];
 if ~isempty(gs1)
 for k=1:n_germ1
     sample1=[sample1,select(Q1./sum(Q1))];
 end
 sampl1=gs1(sample1);
end

 %now level 2
 prob_germ2=1./[1:4];prob_germ2=prob_germ2./sum(prob_germ2);
 n_germ2=select(prob_germ2)-1;
 gs2=intersect(gs,L2);
```

```
 sample2=[];Q2=Q(gs2);sapl2=[];
 if ~isempty(gs2)
 for k=1:n_germ2
     sample2=[sample2,select(Q2./sum(Q2))];
 end
 sampl2=gs2(sample2);
end

 %now level 3
 prob_germ3=3./[1:2];prob_germ3=prob_germ3./sum(prob_germ3);
 n_germ3=select(prob_germ3)-1;
 gs3=intersect(gs,L3);
 sample3=[];Q3=Q(gs3);sampl3=[];
 if ~isempty(gs3)
 for k=1:n_germ3
     sample3=[sample3,select(Q3./sum(Q3))];
 end
 sampl3=gs3(sample3);
end

 %now level 4
 prob_germ4=1./[1:1];prob_germ4=prob_germ4./sum(prob_germ4);
 n_germ4=select(prob_germ4)-1;
 gs4=intersect(gs,L4);
 sample4=[];Q4=Q(gs4);sampl4=[];
 if ~isempty(gs4)
 for k=1:n_germ4
     sample4=[sample4,select(Q4./sum(Q4))];
 end
 sampl4=gs4(sample4);
end


 n=length(sampl1)+length(sampl2)+length(sampl3)+length(sampl4);
 content(:,1)=[1:n]';
 if ~isempty(content)
 content(:,2)=[sampl1,sampl2,sampl3,sampl4]'
end
```

Finds connected components in configuration"

```
function [c,v] = conn_comp(a,tol)
warning off
%       Finds the strongly connected sets of vertices
```

```
%                in the DI-rected G-raph of A
%          c = 0-1 matrix displaying accessibility
%          v = displays the equivalent classes

%make symmetric
a=(a+a')/2;

[m,n] = size(a);
if m~=n 'Not a Square Matrix', return, end
b=abs(a); o=ones(size(a)); x=zeros(1,n);
%msg='The Matrix is Irreducible !';
%v='Connected Directed Graph !';
v=zeros(1,m);v(1,:)=1:m; %?????????????????????????????????????????????????
if (nargin==1) tol=n*eps*norm(a,'inf'); end

% Create a companion matrix
b>tol*o; c=ans; if (c==o)  return, end
% Compute accessibility in at most n-step paths
for k=1:n
      for j=1:n
            for i=1:n
                  % If index i accesses j, where can you go ?
                  if c(i,j) > 0  c(i,:) = c(i,:)+c(j,:); end
            end
      end
end
% Create a 0-1 matrix with the above information
c>zeros(size(a)); c=ans; if (c==o) return, end

% Identify equivalence classes
d=c.*c'+eye(size(a)); d>zeros(size(a)); d=ans;
v=zeros(size(a));
for i=1:n find(d(i,:)); ans(n)=0; v(i,:)=ans; end

% Eliminate displaying of identical rows
i=1;
while(i<n)
      for k=i+1:n
            if v(k,1) == v(i,1)
                  v(k,:)=x;
            end
      end
      i=i+1;
end
j=1;
for i=1:n
```

```
        if v(i,1)>0
           h(j,:)=v(i,:);
           j=j+1;
        end
end
v=h;
%end
```

Connects bonds down:

```
function [content,connector,found]=connect_down_bond(content,connector, i,h,omega,Q_theme)
%finds generator to connect to open down bond (i,h,omega)
load c:\mind_data G mod_transfer gs_in_mod Q A T
g=content(i,2);n=length(content(:,1));
if ~isempty(connector)
    m=length(connector(:,1));
else m=0;
end


%connect generator to what? Set of "to_gs" =v;
s=G(g);
mod=s.modality;
to_mods=mod_transfer{mod,omega};to_gs=gs_in_mod(to_mods);n_to_gs=length(to_gs);
%connect to g's?
v=[];
for nu=1:n_to_gs
    v=[v,to_gs{nu}];
end
to_gs=v;
old_gs= ismember(content(:,2),to_gs);
    if any(old_gs)
        u=content(:,1);v=content(:,2);
        to_h=u(logical(old_gs));
        to_g=v(logical(old_gs));n_to_h=length(to_h)

        %random selection
        probs=[];
        for nu=1:n_to_h
      prob=Q(v(nu))*n/(n+1);prob= prob*A(g,v(nu))^(1/T);probs=[probs,prob];
        end
   probs=probs./sum(probs);
   nu=select(probs);
```

```
        to_h=to_h(nu);
        t=isempty(connector);
        if t==1
            connector=[h,to_h,omega];
            found=1;
            return
        end
        already_connected=(connector(:,1)==h)&(connector(:,2)==to_h);%error?
        if ~any(already_connected)
               connector=[connector;[h,to_h,omega]];
               found=1;
               return
         end
        %else find new g to connect to
    end
         %sample from probs over set "to_gs"
         probs=[];
         for mu=1:n_to_gs
             prob=Q_theme(to_gs(mu))*mu/(n+1);prob= prob*A(g,to_gs(mu))^(1/T);probs=[probs,p
             probs=[probs,prob];
         end
         probs=probs./sum(probs);
         new_g=select(probs);new_g=to_gs(new_g);
         %connect this "new_g" to old content, connector
         content=[content;[max(content(:,1))+1,new_g]];r=1:3;

            connector=[connector;[h,max(content(:,1)),omega]];%note that "content"already be
            %[h,max(content(:,1)),omega]
            found=1;
```

Verifies that down connection is regular:

```
function answer=connection_regular_new(i1,i2,j,content,connector,g_mod,mod_transfer)
%finds whether proposed connection i1->i2  for "j"th down bond is regular
answer=0;
if i1==i2
    %same generator?
    return
end

%first check whether modalities satisfy regularity
h1=content(i1,1);h2=content(i2,2);
g1=content(i1,2);g2=content(i2,2);
mod1=g_mod(g1);mod2=g_mod(g2);
```

```
mod=mod_transfer{mod1,j};
if ismember(mod2,mod)
                answer=1;
                return
end
```

Creates new idea:

```
    function class_idea = create_idea
    %Use local coordinates for idea. Only 2_top_idea allowed
    omega=input(' Down arity = \n');idea_class=cell(1,omega);
    load('C:\mind_data');
    r=length(G);Q=ones(1,r);
    for l=1:omega+1
       svar= input(['for bond no. ', num2str(l),' modality (1) or generators (2) ? \n'])
    if svar==1
        mod=input('modality = ? \n');
        idea_class{1,l}=gs_in_mod(mod)
    elseif svar ==2
        gs=input('give vector of generators \n')
       idea_class{1,l}=gs;
    end
end
class_idea=idea_class;
%store this in driver ws
```

Deletes connection:

```
function [content,connector]=delete_connector(content,connector)
load('C:\mind_data');
%differs from "delete_g"  in that content is not changed
load('C:\mind_data')
%r=length(G);Q=ones(1,r);

n=length(content(:,1));
if isempty(connector)
   return
else
   m=length(connector(:,1));
   j_del=select(ones(1,m)./m)
   h1=connector(j_del,1);h2=connector(j_del,2);
   i1=find(content(:,1)==h1);i2=find(content(:,1)==h2);
```

```
   g1=content(i1,2);g2=content(i2,2);
   prob_del=(A(g1,g2)^(-1/T));
   prob_del=prob_del/(1+prob_del);
   answer =select([prob_del,1-prob_del]);
   if answer==1
      connector=connector([1:j_del-1,j_del+1:m],:);
   else
   end
end
```

Deletes generator from G, use with caution:

```
function delete_g(g,G)
%deletes single generator "g" in "G"
r=length(G);
v=[[1:g-1],[g+1:r]];
G(v);
```

Deletes generator with its connections:

```
function [content,connector]=delete_generator_connections(content,connector)
%this program deletes generator and associated connections
load('c:\mind_data');
if isempty(content)
   return
   else
      n=length(content(:,1));
      %select generator

     i_del=select(ones(1,n)./(n));%in i-coordiantes
      g=content(i_del,2);
if i_del>=n
   return
end

if isempty(connector)
   prob_del=(n/mu)/Q(g);  %check this!
   prob_del=prob_del/(1+prob_del);
   if select([prob_del,1-prob_del])
      content=content([1:i_del-1,i_del+1],:);
      return
   end
else
```

```
   m=length(connector(:,1));

   %bonds down to this generator from others above
   h=content(i_del,1);
j_above=find(connector(:,2)==h);%in j-coordinates
l_above=length(j_above);
product=n/(mu*Q(g));
for j=1:l_above
   j=j_above(j);h1=connector(j,1);
   i1=find(content(:,1)==h1);i2=find(content(:,1)==h);
   g1=content(i1,2);g2=content(i2,2);
   product=product*(A(g1,g2))^(-1/T);
end

%bonds up to this generator from others below
j_down=find(connector(:,1)==h);%in j-doordinates
l_down=length(j_down);
for j=1:l_down
   j=j_down(j);h2=connector(j,2);
   i1=find(content(:,1)==h);i2=find(content(:,1)==h2);
   g1=content(i1,2);g2=content(i2,2);
   product=product*(A(g1,g2))^(-1/T);
end

prob_del=product;%check this!
prob_del=prob_del/(1+prob_del);
answer=select([prob_del,1-prob_del]);
if answer==1
   content=content([1:i_del-1,i_del+1:n],:);
   connector=connector(setdiff([1:m],[j_above',j_down']),:);

else
end
end
end
```

Deletes generators but keeps external inputs:

```
function [content,connector]=delete_generator_keep_input(content,connector)
%this program has been written so that a simple modification (defining "n_input)
% will make the inputted "content" stay unchanged
load c:\matlabr12\golem2\mind_data2 A G Q T g_mod mod_transfer mu;
if isempty(content)
   return
```

```
   else
      n=length(content(:,1));
      %select generator, not input
n_input=0;
     i_del=n_input+select(ones(1,n-n_input)./(n-n_input));%in i-coordiantes
      g=content(i_del,2);
if i_del>n
   return
end

if isempty(connector)
   prob_del=(n/mu)/Q(g);%check this!
   prob_del=prob_del/(1+prob_del);
   if select([prob_del,1-prob_del])
      content=content([1:i_del-1,i_del+1],:);
      return
   end
else
   m=length(connector(:,1));

   %bonds down to this generator from others above
   h=content(i_del,1);
j_above=find(connector(:,2)==h);%in j-coordinates
l_above=length(j_above);
product=n/(mu*Q(g));
for j=1:l_above
   j=j_above(j);h1=connector(j,1);
   i1=find(content(:,1)==h1);i2=find(content(:,1)==h);
   g1=content(i1,2);g2=content(i2,2);
   product=product*(A(g1,g2))^(-1/T);
end

%bonds up to this generator from others below
j_down=find(connector(:,1)==h);%in j-doordinates
l_down=length(j_down);
for j=1:l_down
   j=j_down(j);h2=connector(j,2);
   i1=find(content(:,1)==h);i2=find(content(:,1)==h2);
   g1=content(i1,2);g2=content(i2,2);
   product=product*(A(g1,g2))^(-1/T);
end

prob_del=product;%check this!
prob_del=prob_del/(1+prob_del)
answer=select([prob_del,1-prob_del]);
if answer==1
```

```
   content=content([1:i_del-1,i_del+1:n],:);
   connector=connector(setdiff([1:m],[j_above',j_down']),:);

else
end
end
end
```

Finds idea in ”thought”:

```
function [idea_content,idea_connector]=get_idea_thought(content,connector)
%displays one of the "ideas" in "thought"
[top_2ideas_g,top_2ideas_h]=get_top_2ideas(content,connector);
[idea_content,idea_connector]=single_idea(content,connector,top_2ideas_g{1},top_2ideas_h{1})
```

Finds dominating thought:

```
function [content1,connector1]=dom_thought(content,connector)
%computes connected components in thought chatter and finds largest
%component
if isempty(connector) | isempty(content)
    content1=[];connector1=[];
    return
else
end

n=length(content(:,1));m=length(connector(:,1));
%create DI-graph
graph=zeros(n);
for j=1:m
    h1=connector(j,1);h2=connector(j,2);
    i1=find(content(:,1)==h1);
    i2=find(content(:,1)==h2);
    graph(i1,i2)=1;
end

%find connected components
[c,v]=conn_comp(graph);
    ls=sum((v>0),2);
    [y,i]=max(ls);
    is=v(i,:);is=find(is);is=v(i,is);
    if ischar(is)
        content1=content;connector1=connector;
        return
```

```
    else
    end
 content1=content(is,:);
 %find rows in new connector1
 connector1=[];
 for j=1:m
     if ismember(connector(j,1),content1(:,1))&ismember(connector(j,2),content(:,1))
         connector1=[connector1;connector(j,:)];
     end
 end
```

Gets template for driver:

```
function [content,connector]=driver_template(driver,content,connector,content_idea,connector
%NOTE: DELETE_CONNECTOR HAS BEEN COMMENTED OUT TEMPORARILY TO MAKE SURE THAT
%NO CONNECTIONS ARE LEFT WITHOUT ATTACHED GENERATORS
%transforms mental state with driver expressed as "content_idea"+"connector_idea"
%into new mental state.
% use "name" instead of "driver" in line 0 (as character string)
%include "G" in "driver" workspace!!!!!!!!!!!!!!!!!!!!!
load(['\matlabr12\golem2\',driver])
s=select([activation_probability,1-activation_probability]);
if s==2
    return
end
load \matlabr12\golem2\mind_data2 class_idea
%check if driver is applicable to this drive
x=size(class_idea)

omega_driver=x(1);applicable=1;
for k=1:omega_driver
    if ~ismember(content_idea(k,:),class_idea(k,:))%perhaps cell structures?
        applicable=0;
    end

    if applicable

r=length(G);n=length(content(:,1));m=length(connector(:,1));
%only adds new connections inside idea; use i_ and j_coordinates
%formats:change_idea cell array (2,n_idea) with values in first row
% 'delete' meaning delete this generator
%'same' meaning same generator, unchanged
%'replace' by g
%'random' set of g's, randomly select one from this set
%in second row column 3 g-value; in second row column 4 set of g'values, other columns []
%format of ad_content: 2-column matrix , first column max(content(:,1))+1,
%second column g-values
```

```
%format of ad_connector: 3-column marrix with i-coorinates in first two columns, bond coordi
%format delet_connector: vector of j-coordinates

%keep configuration minus "idea"
keep_h=setdiff(content(:,1),content_idea(:,1));
keep_i=find(ismember(content(:,1),keep_h));
keep_content=content(keep_i,:);
keep_connector=find(ismember(connector(:,1),keep_h)&ismember(connector(:,2),keep_h));
keep_connector=connector(keep_connector,:);
between1=ismember(connector(:,1),keep_h)&ismember(connector(:,2),content_idea(:,1));
between2=ismember(connector(:,2),keep_h)&ismember(connector(:,1),content_idea(:,1));
keep_idea_connector=connector(find(between1'|between2'),:);
n=length(content(:,1));m=length(connector(:,1));
n_idea=length(content_idea(:,1));
n_ad=length(ad_content);
%n_delete=length(delete_content);

m_idea=length(connector_idea);
m_add=length(ad_connector);
m_delet=length(delet_connector);

%begin by changing values (no deletion yet)
del=zeros(1,n);
for i=1:n_idea
    if strcmp(change_idea{i,1},'delete')
        del(i)=1;
    elseif strcmp(change_idea{i,1},'same');
    elseif strcmp(change_idea{i,1},'replace')
        content_idea(i,2)=change_idea{i,2};
    elseif strcmp(change_idea{i,1},'random')
        new_set=change_idea{i,4};n_new_set=length(new_set);
        choose=select([1:n_new_set]./n_new_set);
        content_idea(i,2)=new_set(choose);
    end

end

%then add new generators
content_idea=[content_idea;ad_content];
%then add new connections
if m_add>0
for j=1:m_add
    h1=ad_connector(j,1); h2=ad_connector(j,2);
    %h1=content_idea(1,i1); h2=content(1,i2);
    connector_idea=[connector_idea;[h1,h2,b]];
    end
```

```
end

v=setdiff([1:n_idea],del);
content_idea = content_idea(v,:);

%now delete unneeded connections
%unneeded_i=find(content_idea(:,2));%in i-coordinates for "content_idea"
%unneeded_h=content(unneeded_i,1);
%un=union(ismember(connector_idea(:,1),unneeded_h),ismember(connector_idea(:,2),unneeded_h))
%un=find(un);m_new=length(connector_idea(:,1));
%connector_idea=connector_idea(setdiff([1:m_new],un),:);

%put transformed "idea" back into configuration
new_content=[keep_content;content_idea];

new_connector=keep_connector;
if ~isempty(connector_idea)
new_connector=[keep_connector;connector_idea];
end
if ~isempty(keep_idea_connector)
new_connector=[new_connector;keep_idea_connector];
end
end
end
content=new_content;
connector=new_connector;
```

Executes driver:

```
function [content,connector]=execute_driver(driver,content,connector)
%executes driver named "driver" for (total) idea={content,connector}
load('c:\mind_data')
if isempty(connector)
    return
end
n=length(content(:,1));m=length(connector(:,1));
[top_2ideas_g,top_2ideas_h]=get_top_2ideas(content,connector); %these are the top_2ideas
    n_ideas=length(top_2ideas_g); belongs_to_domain=zeros(1,n_ideas);
    domain=driver{6};
    %find if any of the top_2ideas in idea belongs to "domain" of "driver"
    %check each entry in of top_2idea w.r.t. "domain" of driver
    for k=1:n_ideas
        gs=top_2ideas_g{1,k,:}; n_gs=length(gs);above=gs(1);below=[];hs=top_2ideas_h{1,k,:};
        driv=driver{1};
        belongs_to_domain(k)= ismember(above,domain{1});
        for n=2:n_gs
```

```
        belongs_to_domain(k)=belongs_to_domain(k)&(ismember(gs(k),domain{k}))|isempty(do
    end
    %belongs_to_domain
    if ~belongs_to_domain
        return
    end
    first_idea=min(find(belongs_to_domain));
    gs=top_2ideas_g{1,first_idea,:};hs=top_2ideas_h{1,first_idea,:};n_idea=length(gs);
    %do not execute "driver" for the first idea with probability...
    if rand(1)>driver{5}
        return
    end
end


    %now execute "change_idea" of "driver"
    change_idea=driver{1};dels=[];%i-numbers of deletions
    for i=1:n_idea %enumerates generators in sub-idea
        if strcmp(change_idea{i,1},'delete')
            dels(i)=1;
        else if strcmp(change_idea{i,1},'same')
        elseif  strcmp(change_idea{i,1},'replace')
            i_value=  find(content(:,1)== hs(i));g_new=change_idea{i,2};
            content(i_value,2)=g_new
        elseif strcmp(change_idea{i,1},'random')
            i_value=  find(content(:,1)== hs(i));
            g_set=change_idea{i,2};g_set_n=length(g_set);
            choose=select([1:g_set_n]./g_set_n);
            g_new=g_set(choose);
            content(i_value,2)=g_new;
        end
    end

    %deletes generators with dels==1 (i-numbers in sub-idea)
    del_h=hs(dels);
    if ~isempty(del_h)
        i_dels=[];
        %delete generators
        for k=1:n
            i_dels=[i_dels,find(content(:,1)==del_h)];
        content=content(setdiff([1:n],i_dels),:);
        end
        %delete connections
        j_s=[];
        for j=1:m
            j_s=[j_s,find(ismember(connector(j,1),del_h))|...
```

```
                        find(ismember(connector(j,2),del_h))];
                end
                    connector=connector(setdiff([1:m],j_s),:);
            end

            %add new generators
            ad_content=driver{2};
            content=[content;ad_content]

            %add new connectors
            ad_connector=driver{3};
            connector=[connector;ad_connector];

            %delete connectors in "idea"
            delet_connector=driver{4};
            j=find((connector(:,1)==hs(1))&(connector(:,3)==delet_connector));
            m=length(connector(:,1));
            connector=connector(setdiff([1:m],j),:);

        end
```

Finds element in "G":

```
function find_g
%searches for generator number with given name
name=input( 'specify name  \n','s')
load c:\mind_data
r=length(G);
for g=1:r
    if strcmp(G(g).name,name)
        g
    end
end
```

Finds open bond downwards"

```
function [i,h,omega,found]=find_open_down_bond(content,connector)
%prepares for completing the given thought expressed as content,connectorn
%by searching for open down bond
if isempty(content)
    i=1;h=1;omega=1;found=0;not_found=1;
    'EMPTY THOUGHT'
    return
end

%find"down" open down-bonds
load c:\mind_data
```

```
n=length(content(:,1));found=0;

for i=1:n
    h=content(i,1);g=content(i,2);mod=g_mod(g);
    arity=mod_omegas(mod);
   if (arity >0) & (~isempty(connector))
       m=length(connector(:,1));
        for omega=1:arity
            v=(connector(:,1)==h)&(connector(:,3)==omega);
            if all(v==0)
                found=1;
                return
            end
        end
   end
end
if isempty(connector)
    for i=1:n
        h=content(i,1);g=content(i,2);mod=g_mod(g);
        arity=mod_omegas(mod);
        if arity>0
          found=1;
         omega=1;
        end
        omega=1;
     end
end
```

Computes level sets in "G":

```
function [L1,L2,L3,L4]=get_levels(G);
%computes G-sets for level=1,1...
r=length(G);L1=[];L2=[];L3=[];L4=[];
for g=1:r
   l=G(g).level;
   if l==1
      L1=[L1,g];
   elseif l==2
      L2=[L2,g];
   elseif l==3
      L3=[L3,g];
   elseif l==4
      L4=[L4,g];
   end

end
```

Computes inverse of transformation "mod_transfer":

```
function mod_transfer_inv=get_mod_transfer_inv(mod_transfer)
%computs inverse of "mod_transfer"
n_mods=length(mod_transfer);mod_transfer_inv=cell(1,n_mods);n_mods
for mod=1:n_mods
for k=1:n_mods
    for j=1:3
         if ismember(mod,mod_transfer{k,j})
            mod_transfer_inv{mod}=[ mod_transfer_inv{mod},k];
         else
       end
    end
end
end
```

Finds top-ideas in "thouight":

```
function [top_2ideas_g,top_2ideas_h]=get_top_2ideas(content,connector)
%computes only second level ideas; this MIND is intellectually challenged and
%cannot think about abstractions of level greater than two
%produces only complete ideas
if isempty(connector)
    top_2ideas_g=[];top_2ideas_h=[];
    figure('Units','Normalized','Position',[0 0 1 1])
    axis off
    text(.2,.5,'No top-2ideas','FontSize',32)
    pause(2)
    return
end
load('c:\mind_data')
tops_i=find(ismember(content(:,2),L2));%in i-coordinates
tops_g=content(tops_i,2);
%above in g-coordinates
tops_h=content(tops_i,1);
% above is in h-coordinates
n_tops=length(tops_i); top_2ideas_g=cell(1,n_tops);top_2ideas_h=cell(1,n_tops);
for k=1:n_tops
   top_2ideas_g{1,k,1}=tops_g(k);
   top_2ideas_h{1,k,1}=tops_h(k);
   top_g=tops_g(k);top_h=tops_h(k);mod=G(top_g).modality;omega=mod_omegas(mod);
   f=find((connector(:,1)==top_h)&(connector(:,3)==1));
   if ~isempty(f)
   f1=connector(f,2);i=find(content(:,1)==f1);f=content(i,2);
   top_2ideas_g{1,k,:}=[top_2ideas_g{1,k,:},f];
   top_2ideas_h{1,k,:}=[top_2ideas_h{1,k,:},f1];
   end
```

```
   f=find((connector(:,1)==top_h)&(connector(:,3)==2));
   if ~isempty(f)
   f1=connector(f,2);i=find(content(:,1)==f1);f=content(i,2);
   top_2ideas_g{1,k,:}=[top_2ideas_g{1,k,:},f];
   top_2ideas_h{1,k,:}=[top_2ideas_h{1,k,:},f1];
   end
   f=find((connector(:,1)==top_h)&(connector(:,3)==3));
   if ~isempty(f)
   f1=connector(f,2);i=find(content(:,1)==f1);f=content(i,2);
   top_2ideas_g{1,k,:}=[top_2ideas_g{1,k,:},f];
   top_2ideas_h{1,k,:}=[top_2ideas_h{1,k,:},f1];
   end
end

%find complete ideas
complete=zeros(1,n_tops);
for k=1:n_tops
    v=top_2ideas_g{1,k,:};
    top=v(1);mod=g_mod(top);omega=mod_omegas(mod);
    if (length(v)==1+omega)
        complete(k)=1;
    end
end

%now keep only complete ideas
top_2ideas_g=top_2ideas_g(find(complete));
top_2ideas_h=top_2ideas_h(find(complete));
```

APPENDIX 5. GOLEM Living Alone The memory parameters Q and A are updated as the the mental trajectory evolves in MIND; see Section 7.2. If no external inputs occur and no internal modifications in terms of GENRE changes are made, GOLEM is living in isolation. What happens to its MIND in seclusion?

Let us study the temporal development of $\overline{Q} = \overline{Q}(t,g); t = 1,2,3,..; g \in G$. Let us used normalized $\overline{Q}$; $\sum_g \overline{Q}(t,g) = 1$ where the bar in $\overline{Q}$ indicates averaging. The updating described will result in the recursion

$$\overline{Q}(t+1,g) = \frac{\overline{Q}(t,g)\sum_{h\neq g}\overline{Q}(t,h)(1-\epsilon_{forget}) + \overline{Q}^2(t,g)(1+\epsilon_{remember})}{(\sum \overline{Q}(t,g))^2 + \sum_g \overline{Q}^2(t,g)(1+epsilon_{remember} - \epsilon_{forget})} \quad (75)$$

where the denominator corresponds to normalization of the $\overline{Q}$-vector and $\epsilon_{remember} > \epsilon_{forget}$. Hence

$$\frac{\overline{Q}(t+1,g_1)}{\overline{Q}(t+1,g_2)} = \frac{\overline{Q}(t,g_1)}{\overline{Q}(t,g_2)} \times \frac{1-\epsilon_{forget} + \overline{Q}(t,g_1)(\epsilon_{remember} - \epsilon_{forget})}{1-\epsilon_{forget}\overline{Q}(t,g_2)(\epsilon_{remember} - \epsilon_{forget})} \quad (76)$$

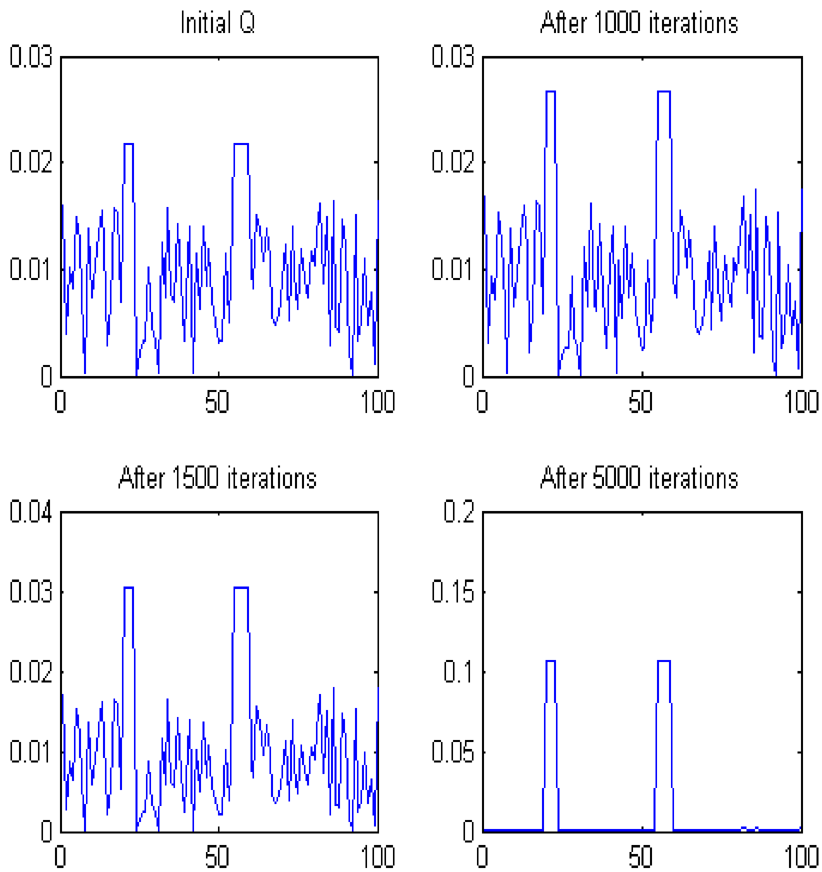
Now let $g_1 \in G_{max}$ with the maximal set $G_{max} = \{g : \overline{Q}(t,g) = max_h \overline{Q}(t,h)\}$. Then the second ratio on the right hand side is greater than one, so that the ratios $\frac{\overline{Q}(t,g_1)}{\overline{Q}(t,g_2)}$are increasing and the maximal set remain maximal. We also get

$$\frac{\overline{Q}(t+1,g)}{\overline{Q}(t,g)} = \frac{1 - \epsilon_{forget} + \overline{Q}(t,g)(\epsilon_{remember} - \epsilon_{forget})}{1 - \epsilon_{forget} + average[\overline{Q}(t,\cdot)](\epsilon_{remember} - \epsilon_{forget})} \tag{77}$$

so that the $\overline{Q}(t,g)$ are increasing and this increse will continue to do so until, in the limit, $\overline{Q}(g_1)/average(Q) \approx 1$, that is we have a flat $\overline{Q}$ for those elementary ideas for which $\overline{Q} \neq 0$. This means that we have the following

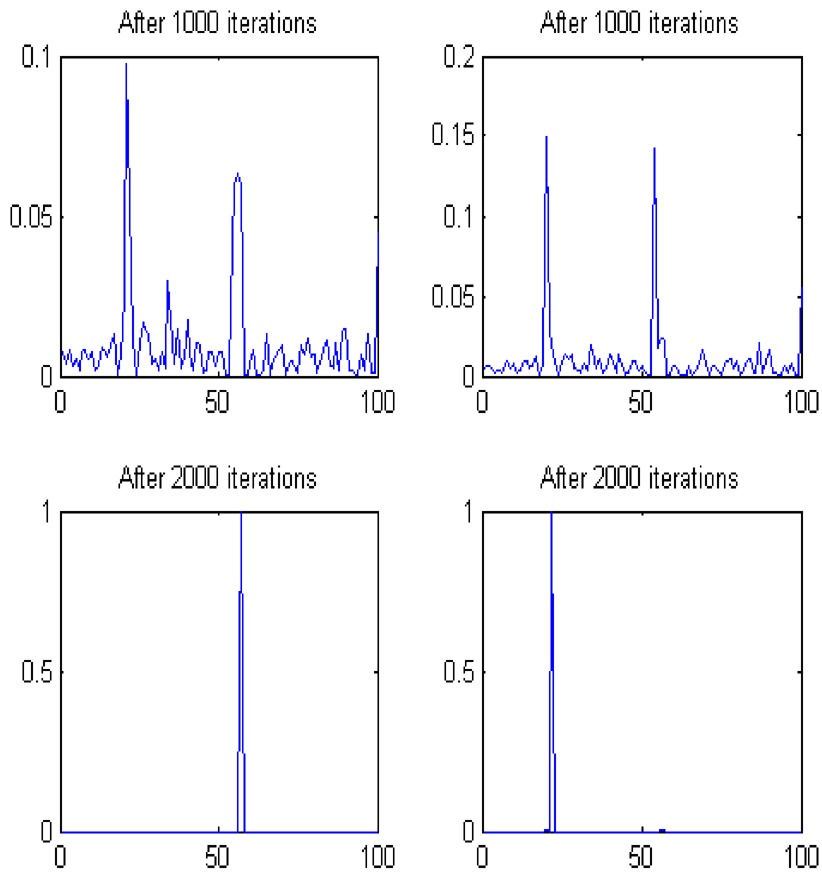
PROPOSITION. *The entries in the* $\overline{Q}$- *vector tend to zero except for the maximal set. In other words, GOLEM's mind tends to degenerate state excluding all primitive thought except for those in the maximal set.*

As an illustration let us start with a small $\overline{Q}$-vector $\overline{Q} = (.1 .1, .1, .15, , .15, .2, .2)$ so that $\overline{Q}_{max} = (6, 7)$. Iterating the recursion euation we get



with the convergence stated in the Proposition.

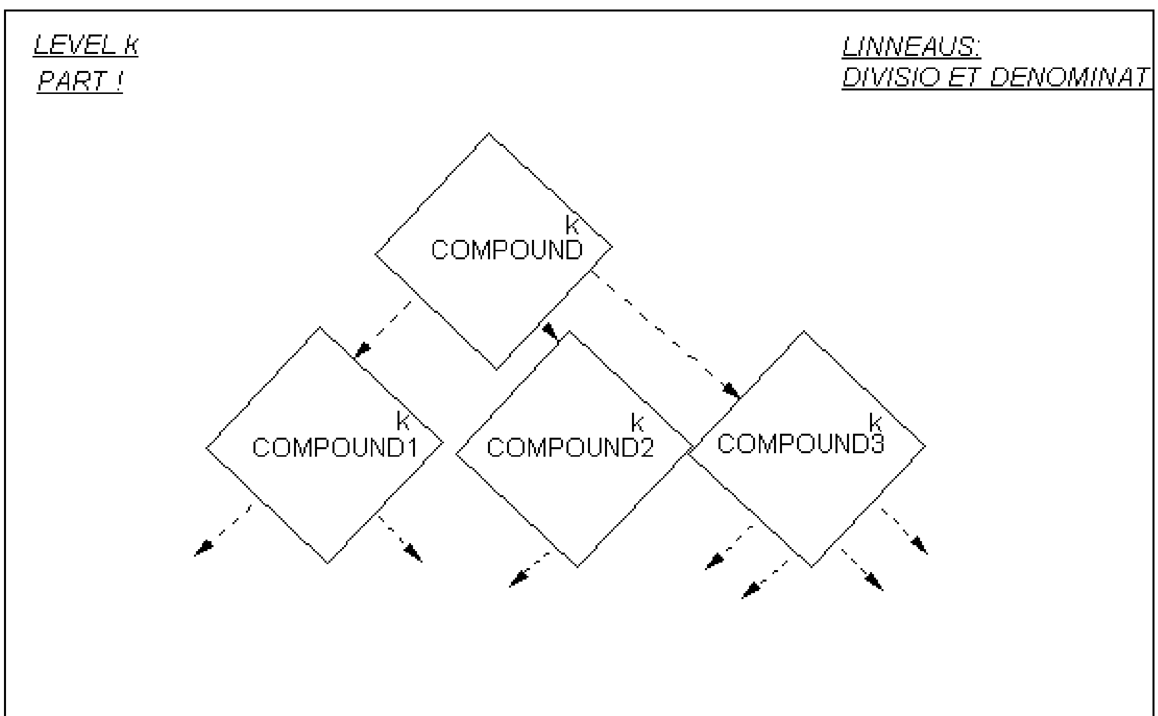
Remember, however, that we have dealt with an averaged version of the Q's while the Q's themselves form a stochastic process whose behavior for $t = 1, 2, 3...$ will influence the later values. We therefore expect the asymptotic values of $Q(t, g)$ to be in the maximal set with large probability but other values can also occur. An example is



APPENDIX 6.

Mental Divisions and Their Thought Patterns. The modality lattice used in the present version of GOLEM is too limited and we shall try to build a more satisfactory one. In order to do this we will have to be more systematic and start from some general principles.

The architecture will still be in the form of levels, modalities and other divisions of the modality lattice $\mathcal{M}$. On each level we shall first divide M according to the arity $\omega = 1, 2, 3$ and then subdivide repeatedly untill a modality is reached. Therefore, at level k the first subdivision will look like Figure 1.



We use the same sort of illustrations as in APPENDIX 3 with diamond shapes for non-modal divisions of M and rectangles for modalities. Arrows with dotted lines relate divisions on the same level. In Figure 1 the division COMPOUND with a superskrpt k indicates that it belongs to level k; the $COMPOUND1^k$ means that it is still on level k but with arity restricted to $\omega = 1$. Then subdivisions continue as in Figures 3,4..., but for level 1 we start differently since its generators have arity $\omega = 0$ so they do not possess down-bonds. Instead we shall use the diagram in Figure 2, where the first division UNITS is divided into MATERIAL and IDEA and so on. We have continued the left branch all the way down to the modality stage but the other branches are not shown in order not to clutter up the picture.

LINNEA
DIVISIO ET DE

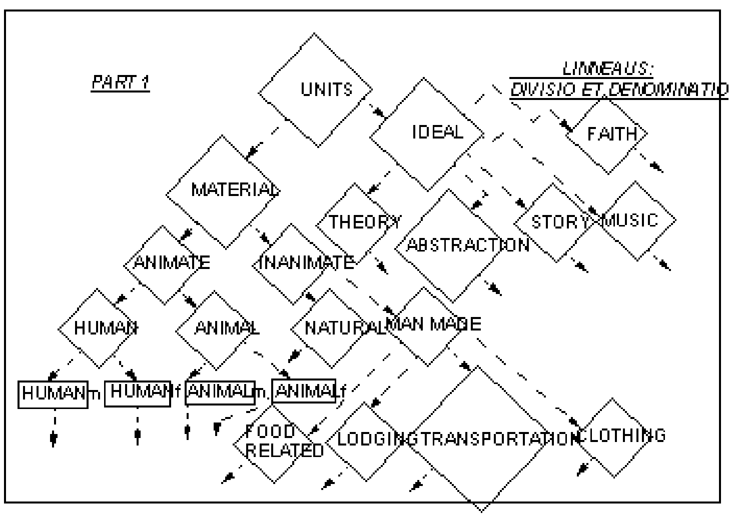


Figure 2

Then back to level 2. In Figure 3 we show how the division $COMPOUND1^2$ has been sub-divided into $\underset{MATERIAL}{MODIFIER}1^2$ and $\underset{IDEAL}{MODIFIER}1^2$

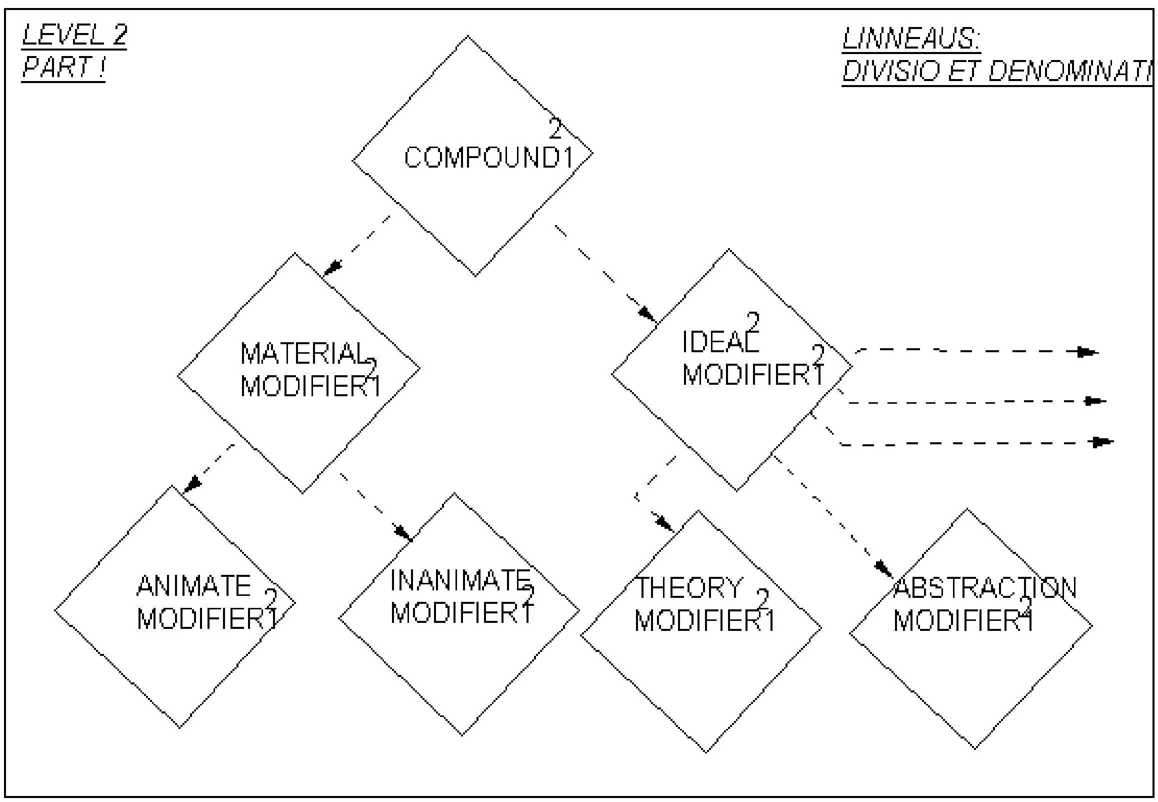

LEVEL 2
PART !
LINNEAUS:
DIVISIO ET DENOMINATI
COMPOUND1 2
MATERIAL MODIFIER1 2
IDEAL 2 MODIFIER1 2
ANIMATE MODIFIER1 2
INANIMATE MODIFIER1 2
THEORY MODIFIER1 2
ABSTRACTION MODIFIER1 2

Figure 3

Again, the superscript 2 refers to level 2 while the "1" in MODIFIER1 refers to arity $\omega = 1$.

Further $\overset{MATERIAL}{MODIFIER1^2}$ is divided into $\overset{ANIMATE}{MODIFIER1^2}$, and into $\overset{INANIMATE}{MODIFIER1^2}$.

The division $\overset{IDEAL^2}{MODIFIER1^2}$ is divided into five subdivisions, the two first of which are $\overset{THEORY}{MODIFIER1^2}$ and $\overset{ABSTRACTION}{MODIFIER1^2}$.

But $COMPOUND2^2$ in Figure 4 is divided into three divisions with arities

$\omega = 2$

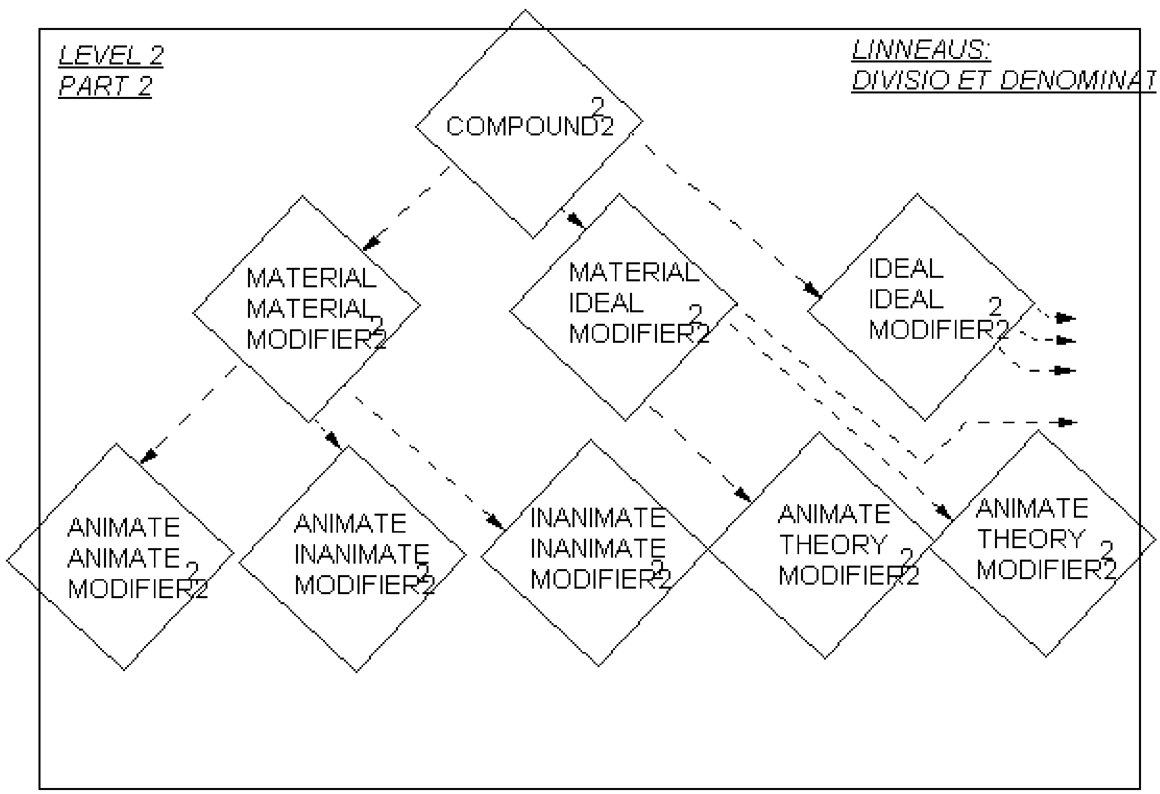


Figure 4

They represent the combinations $\overset{MATERIAL}{MATERIAL}$, $\overset{MATERIAL}{IDEAL}$ and $\overset{IDEAL}{IDEAL}$, each with the $MODIFIER2^2$. These in turn are further divided into combinations; only some are shown in Figure 5.

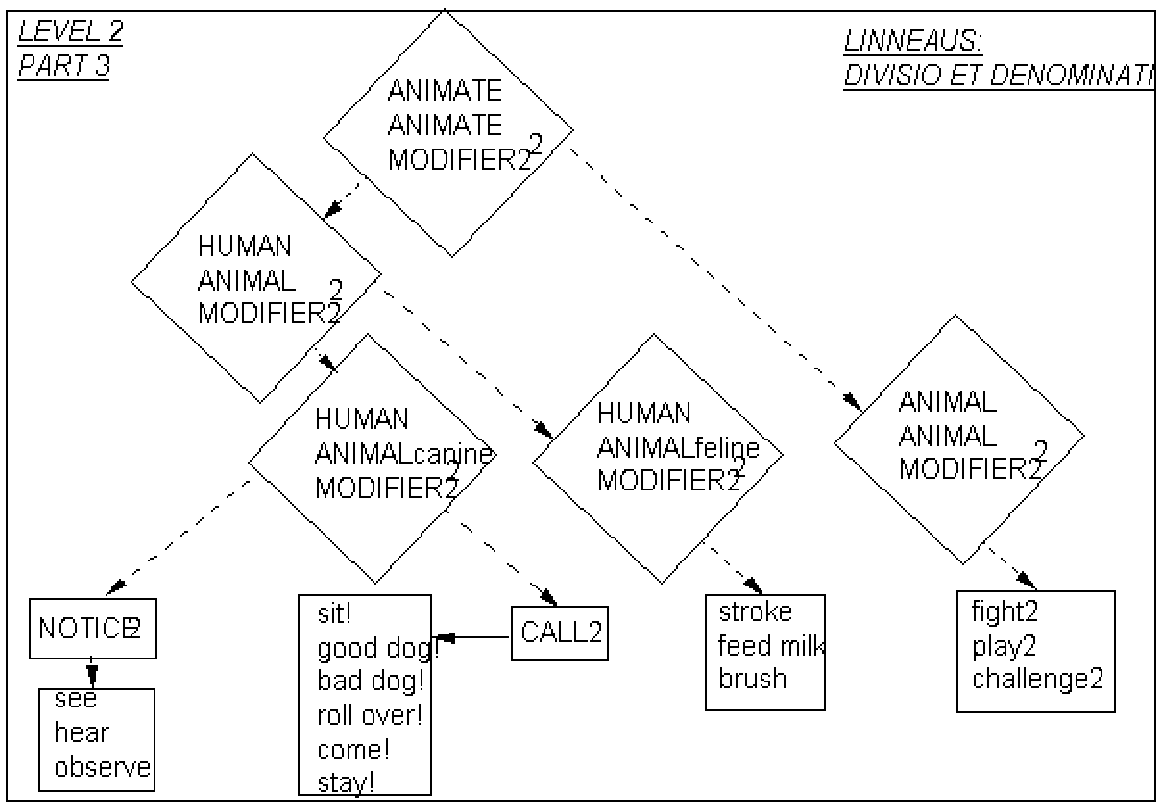


Figure 5

Note that some of the branches in this Linnean tree extend all the way to the ultimate generators on level one.

An important division is the one in Figure 6 for handling information. The modality transfer function points from a division $KNOWLEDGEHANDLING$

to $KNOWLEDGE^2$

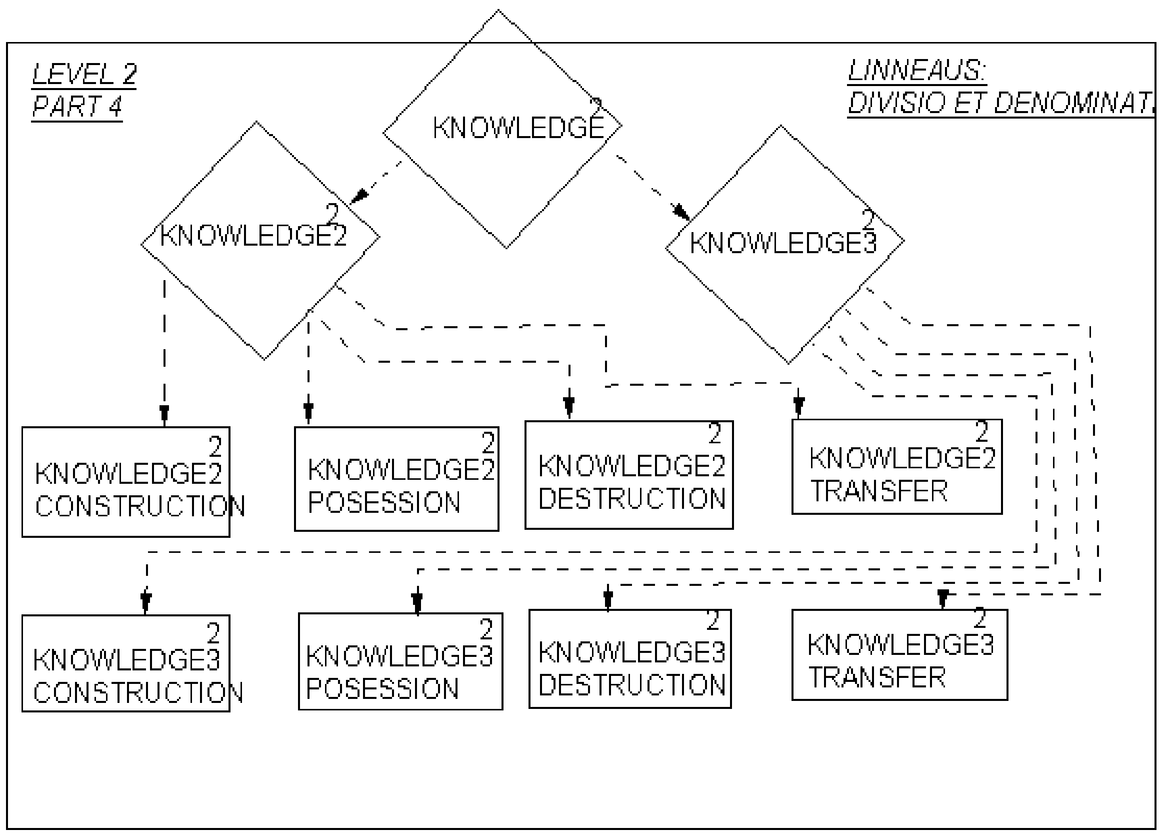


Figure 6

Now level three. In Figure 7 part of this level is shown, leaving out the upper part of the tree of the form in Figure 1

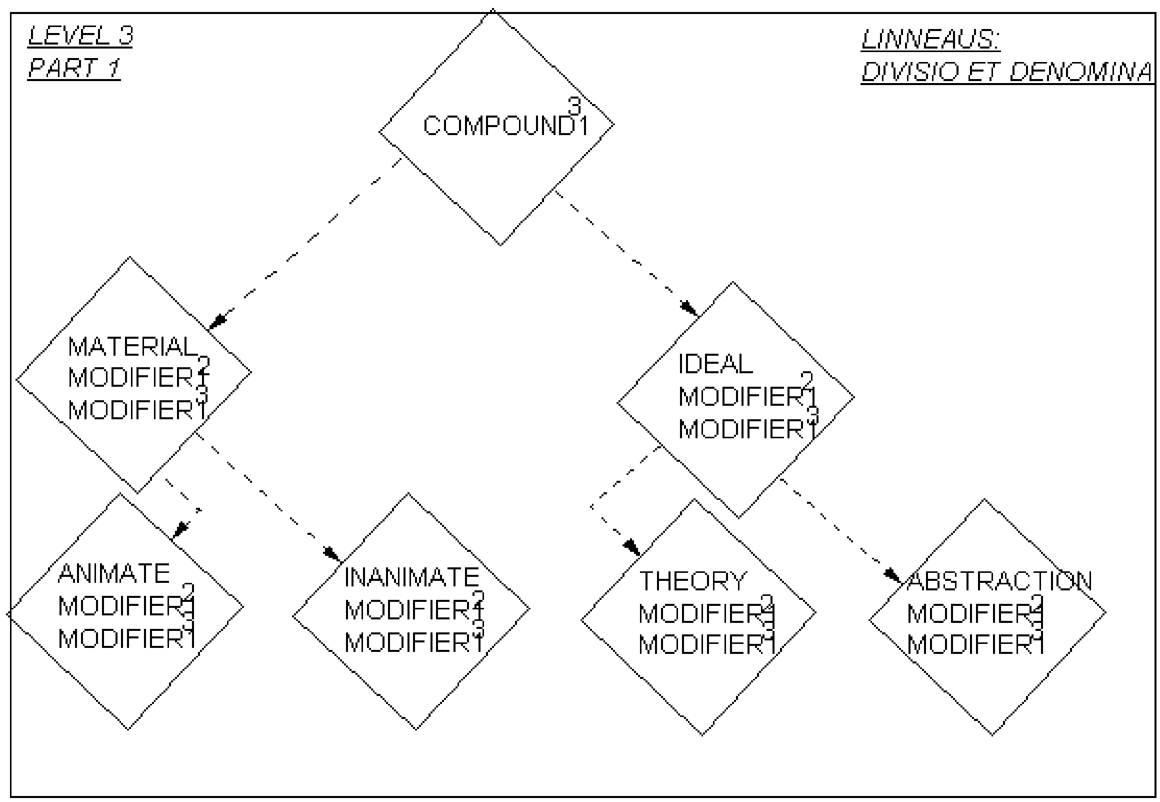


Figure 7

Two modifiers now appear with $MODIFIER1^3$ signifying still another modification to be applied; same for the other divisions most of which are not shown in Figure 6. This corresponds to the increase in specificity for increasing levels. We believe that four levels should be enough for most thought structures; recall that the mental operation ideafication allows for unlimited abstraction levels.

To handle the resulting division structure, including non-modal divisions, we should develop software for the construction of a more satisfactory $\mathcal{M}$..

Let us consider a simple thought pattern in Figure 8

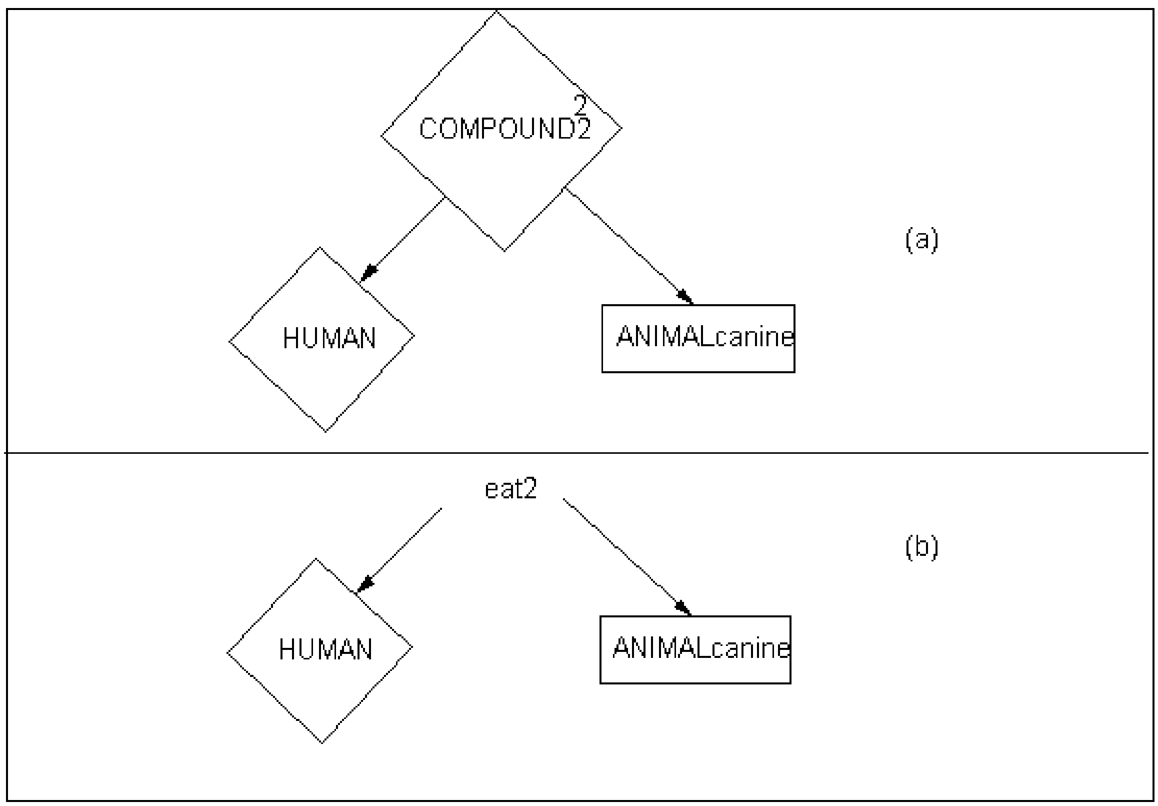


Figure 8

This diagram represents a whole pattern of thoughts [12], all about humans dealing with dogs . It could be the sub-pattern of humans talking to dogs or, another sub-pattern of a human feeding Rufus, and so on. But some of the thoughts may have so high energy that they would seldom reach the conscious level, for example the one depicted in Figure 7b (at least for Western thoughts...). Adopting the term "ground state" from physics we shall speak of the *ground thought(s)* as the set of thoughts with minimum energy in a thought pattern. In Figure 8 the sub-pattern "HUMAN calls dog" may be the ground thought.

[12] See Section 4.6.

A more complicated thought pattern is shown in Figure 9

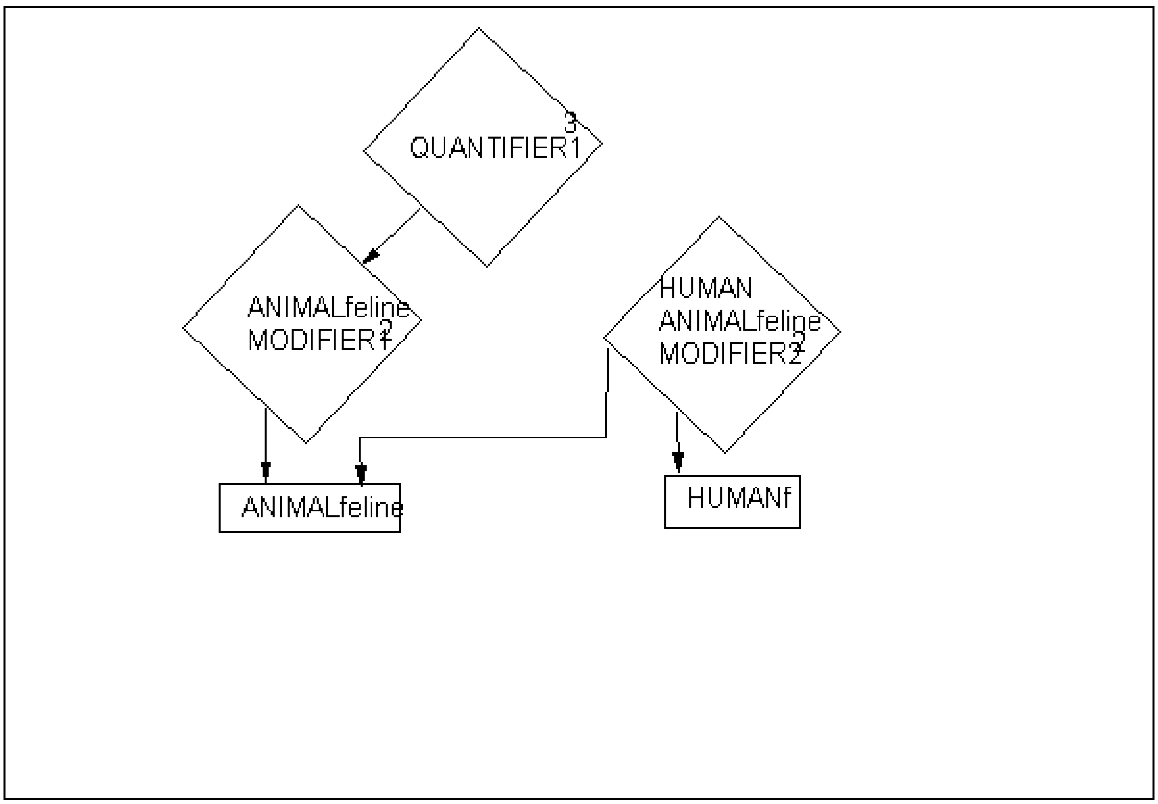


Figure 9

containig for example the thought ”Mary strokes the very happy cat” or ”a woman sees the fairly big Felix”.

Still another example of thought patterns, but one involving abstrctions, is

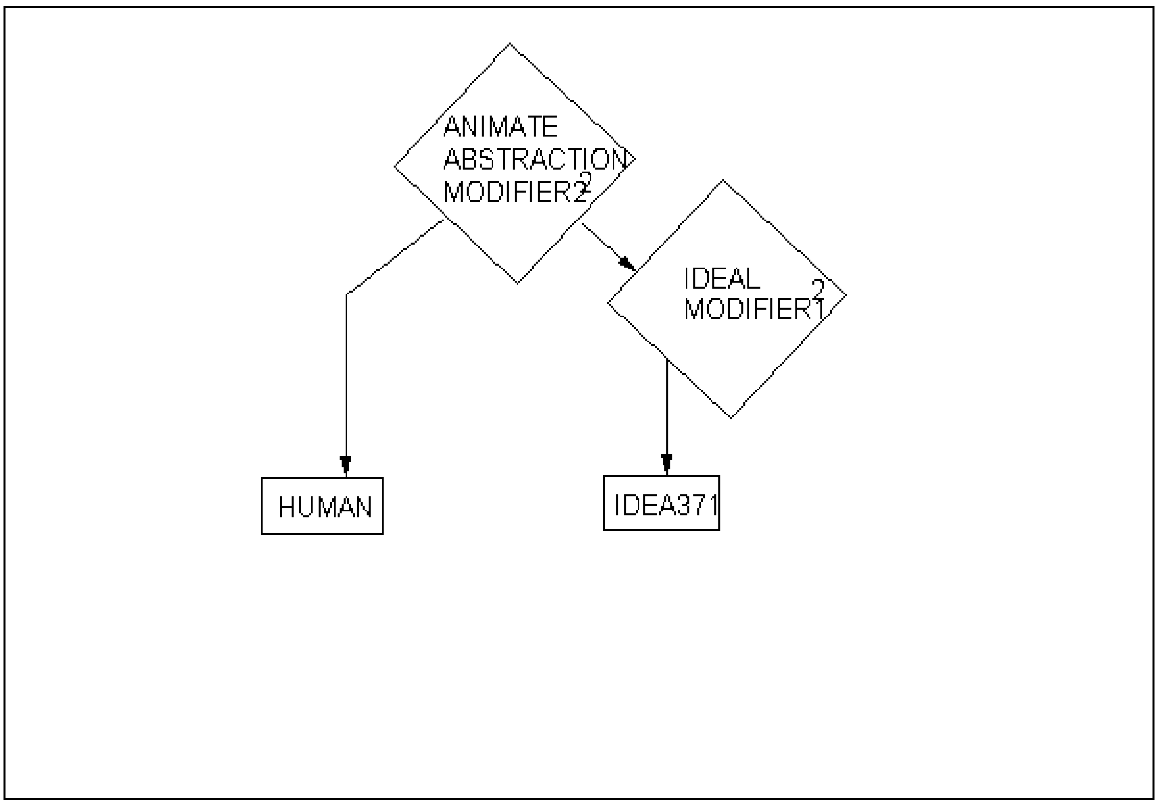


Figure 10

containing for example the thought "Bob believes that it will rain tomorrow" if IDEA371 is the thought "it will rain tomorrow".

Thought patterns can be classified according to their topology (connector diagram). We show some of the simpler ones in Figure 11. The ones in the first row consist just of a single elementary thought, the ones in the second row have a single elementary thought belonging to the second level, and so on.

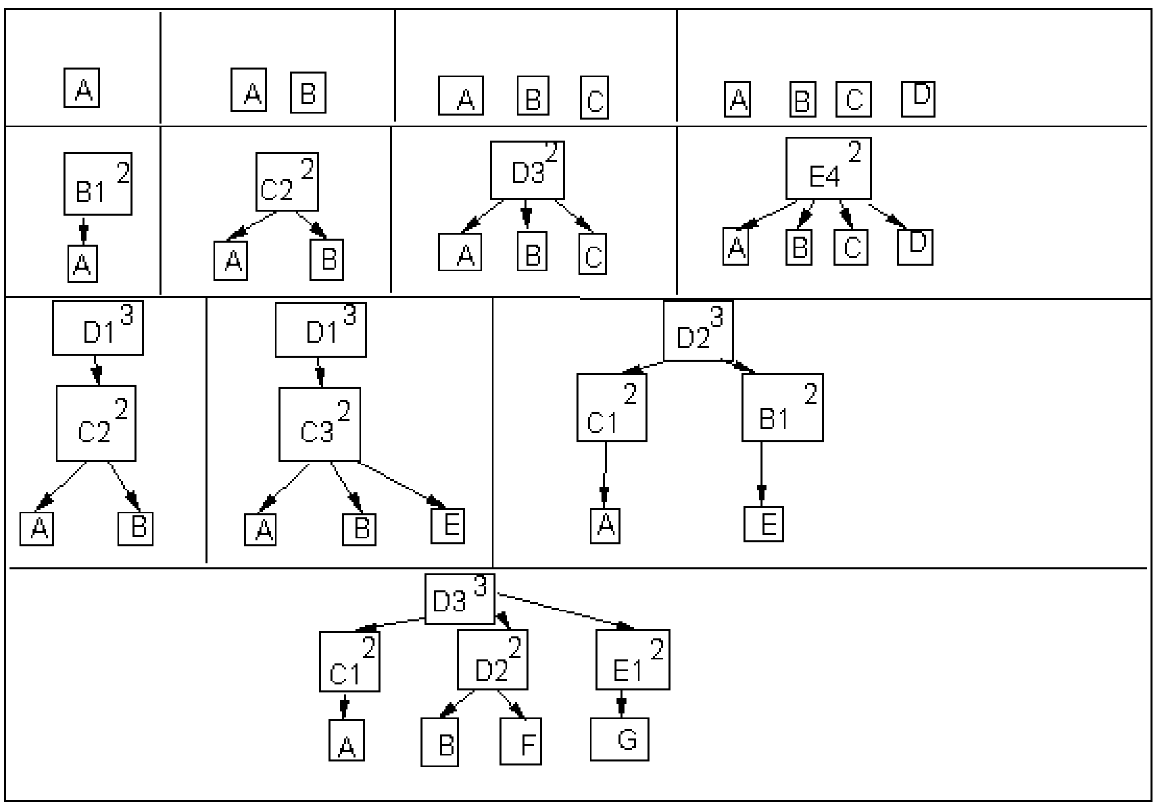


Figure 11

When a more satisfactory division (than the one in GOLEM) has been constructed one should investigate empirically which thought patterns are occurring often, but not just in terms of topology but expressed in modalities and other divisions. This would lead to a taxonomy of thought that could help us better understand human thinking.

Transformations of Thought Patterns. With the help of this taxonomy for $\mathcal{M}$ we can study transformation of thought more generally than we did in Section 5. Let us start with some simple ones.

Size Preserving Transformations. In Figure 12a we see a thought pattern that is being transformed by a *single substitution* $A2^2 -> B2^2$, for example *self is thinking for himself -¿ self is talking to himself*

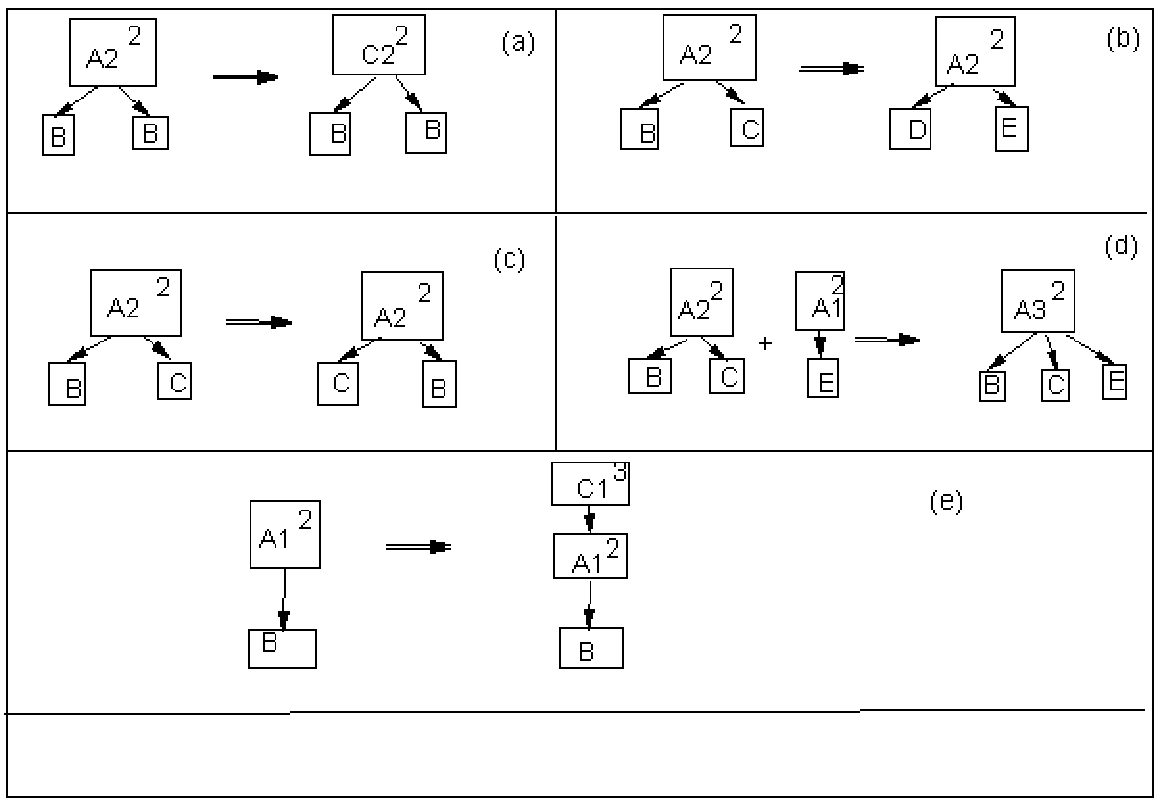


Figure 12

while (b) has a *double substitution* $B- > D, C- > E$, for example *Susan talks to Jim -¿ Mary talks to Bob.* Diagram (c) shows a *switch* $B- > C, C- > B$, for example *the cat chases a dog -¿ the dog chases the cat.*

Size Changing Transformations The transformation in (d) is *contracting* in that size is decreasing, n=5 to n=4. The two thought patterns react with each other, for example *Paul and Tom play with each other and Susan plays alone -¿ Paul, Tom and Susan play together.* Energetically this could be a conscious transformation if *E(Paul and Tom play with each other and Susan plays alone) is greater than E(Paul, Tom and Susan play together).*

Other contracting transformations occur when one or several elementary ideas are deleted with their connections.

On the other hand, (e) shows an *expanding* transformation, from n=2 to n=3. This transformation could represent for example the thoughts *a beautiful flower -¿ a very beautiful flower.*

These are just a few of the mental transformations that are likely energetically.