Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Technical Memo

TM-91-15

Prototypical Concept Formation

An Alternative Approach to Knowledge Representation

Stefan Bussmann

December 1991

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern und Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Daimler Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Krupp-Atlas, Mannesmann-Kienzle, Philips, Sema Group Systems, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ❑ Intelligent Engineering Systems
- ❑ Intelligent User Interfaces
- ❑ Intelligent Communication Networks
- ❑ Intelligent Cooperative Systems.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Prof. Dr. Gerhard Barth
Director

Prototypical Concept Formation
An Alternative Approach to Knowledge Representation

Stefan Bussmann

DFKI-TM-91-15

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-9104).

Contents

§ 1. Introduction

Concept formation is a fundamental component of cognitive processing. In a world which is continuous and constantly changing we have to cope with bulks of sensory data every second. Human beings would not survive if they did not reduce incoming data to the necessary and important information they need to deal with our environment. One way of accomplishing this task is to abstract properties of objects, in order to identify or group them under one concept. Thus, concept formation is a method of organizing our environment at the level of abstraction at which we want to deal with it. Structuring and generalizing the world into concept hierarchies can help us to break down its complexity.

In this paper we want to understand the term *concept* as an abstract form which denotes a group of objects. How the group could be structured or defined is what we will discuss later on. The idea is that humans use concepts to refer to a certain part of the world. The topic gets more complicated, if we do not restrict the meaning of *object*, since humans can build concepts of almost anything which incluces their own cognitive processing. For instance, we have a notion of thoughts or even concepts, but this makes concept formation more than a mere fundamental component of human thinking. To avoid complex interactions and dependencies we will only discuss concept formation of objects which are situated in our environment, so called *natural objects*. A natural object could be a table, a chair, a tree, or even colour, noise or rain. Events and actions could also be denoted by concepts. We regard, loosely speaking, all "objects" we can find in our environment as natural.

Even at this rather basic level of concept formation we have to deal with specialized concepts as time, space, and identity. These and other concepts constitute the prerequisite to understand and reason about our environment at all. They are so much fundamental that it is not obvious, whether we learned or inherited them. The ability to identify a perceived object as being identical to a different object which we have encountered at a previous moment is not a trivial cognitive act and is not necessarily given to us a priori. Other concepts we do not seem to form the usual way include equivalence of fluids or other materials, casuality, the notion of number or probability. It's not the aim of this paper to investigate on these concepts, but rather to discuss the concept formation we use for most objects in our environment.

In chapter 2 of this paper we will review the classical approach to concept formation and the criticism which led to the prototype theory. (An overview of this can be found in Medin & Smith 1984.) Prototype theory is the main topic of this paper and we will discuss it in more detail in chapter 3 and point out some criticism in chapter 4. We will not cover probabilistic approaches which also tried to overcome the shortcomings of classical concept formation. In chapter 5, we will give an overview of prototypical models. Then, in chapter 6, we will present our approach and discuss its implementation which we have realized in order to get an idea how practical such an approach is.

§ 2. Classical Concept Formation and its Criticism

2.1 Classical Concept Formation

Classical concept formation identifies a concept with an exact definition. A **classical concept** has a fixed number of properties which are necessary and sufficient for an object to be an instance of the concept. Additionally, every concept has a number of rules which specify how these properties have to be fulfilled. For example, we expect a bird to have wings and be able to fly. This is a conjunctive rule since both properties have to be present. Different rules, as disjunctive, exclusive, or any other logical connectives, may be used.

A classical concept is completely defined by its necessary properties and their connection rules. There is no need for more information. The defining properties imply a clear membership by checking the properties. That means one can decide unequivocally if an object is an instance of a concept. Consequently, classical concepts denote a (sharp) part of the world, which is also called *class*. Instances of a class are equal members of the concept, since all satisfy the definition. Especially Bruner, Goodnow & Austin (1956) have studied classical concepts. They focused on aspects as how humans find the defining properties given the connection rules and how they find the rules given the defining properties.

Classical concept formation has the implicit assumption that the concepts denote parts of a total set (the world) which contains all objects. All combinations of properties are expected to be equally likely. This arbitrariness allows us to form a concept by simply choosing a set of properties and the connection rule. Any choice would create a useful concept hierarchy. The advantages of the classical approach are obvious: The concepts are defined logically, allowing us to reason about them. Secondly, we can easily decide concept membership and we have a clear concept boundary. As a consequence, classical concepts are easy to work with.

2.2 The Criticism

At the beginning of the seventies criticism grew that the concepts studied in the experiments had been artificial. Several psychologists started to examine natural concepts as colour (e.g. Rosch 1973) and came up with concept structures differing considerably from classical concepts. Mervis & Rosch (1981) pointed out three qualities that classical concepts missed:

1. Concepts are not arbitrarily defined,
2. instances are not equal in their rights,
3. there are no clear concept boundaries.

2.2.1 Nonarbitrariness of Natural Concepts

As pointed out, it is crucial for classical concepts that they can be arbitrarily defined. Yet, this would only make sense, if all combinations of attribute values were equally likely to occur which is not true for natural objects. Let's consider three often used qualities to classify animals: "coat" (fur, feathers), "oral opening" (mouth, beak), and "primary mode of locomotion" (flying, on foot). Thus the total set can be divided into eight classes:

1) those with fur and mouth, which move primarily on foot,
2) those with fur and mouth, which move primarily by flying,
3) those with fur and beak, which move primarily on foot,
4) those with fur and beak, which move primarily by flying,
5) those with feathers and mouth, which move primarily on foot,
6) those with feathers and mouth, which move primarily by flying,
7) those with feathers and beak, which move primarily on foot,
8) those with feathers and beak, which move primarily by flying.

In reality only combinations 1) and 8) can be found, namely mammals and birds. The world seems to be nonarbitrarily structured (which can be supported by more examples of natural objects). Mervis & Rosch conclude that we do not choose concepts but that the world itself is structured and it proposes the concepts we use. One reason is that attributes always appear in certain combinations (like fur, mouth, and moving on foot).

Mervis & Rosch also found (human) concept hierarchies to be fairly structured. They distinguished three levels of concrete objects: the basic level, the superordinate level, and the subordinate level. Let´s take the concept chair as an example. Chair itself would be of the basic level, a camping chair of a subordinate level, and furniture of a more general, the superordinate level of categorization. This kind of hierarchical organization could be confirmed for almost all natural concepts.

Rosch et al. (1976) conducted experiments on the basic level. The following results were mostly true for basic level concepts:

1) a person uses similar motor actions for interacting with category members,
2) category members have similar overall shapes,
3) a mental image can reflect the entire category,
4) category members have many attributes in common.

Additionally, objects of basic level concepts are recognized as instances more rapidly and children learn these concepts first. From their results Mervis & Rosch conclude that arbitrariness is not true for natural objects, but in contrast, we classify our world as we experience it.

2.2.2 Unequivalence of Instances

Instances of a classical concept are members of the same right because they have fulfilled the definition of that concept. The group of members is just a set of undistinguished objects. But this is not true for natural concepts. They contain representative and nonrepresentative members. Regard the classical example: Talking about birds we think about robins or blackbirds, but hesitate to call a pelican or an ostrich a bird. From the classical point of view we should mention and regard them equally likely. Nevertheless humans treat them differently. They list unnecessary when asked to give defining properties, e.g. flying for birds. Still they accept objects as instances, even though these miss properties which they have listed as defining.

Experiments of Rosch (1975) and others supported the unequivalence of instances. Representative members were classified as instances more rapidly than nonrepresentative. Representative members are mentioned before nonrepresentative, if asked to list examples. Typical members are regarded more similar to untypical than vice versa. And representative members are learned first. All this is strong evidence that members of natural concepts are not treated equally.

2.2.3 Unclear Concept Boundaries

A major (practical) problem for classical concept formation was to give exact definitions for natural objects. Despite much effort to provide these definitions we still do not know satisfying definitions for concepts as human beings or birds. Furthermore, logical problems arose like the frog-tadpole paradox: Imagine a newly born tadpole in a small aquarium. We take a picture of it every hour. This produces a sequence of pictures on which the tadpole slowly develops into a grown-up frog. This slow development embodies the difficulty to find a picture on which the creature is still a tadpole but becomes a frog on the next one. If we presuppose classical concept formation, such a picture should exist. But it does not!

Many experiments showed that natural concepts do not have these clear boundaries, the frog-tadpole paradox is one example. Humans showed disagreement between their own opinions when asked to classify nonrepresentative concept members (McCloskey & Glucksberg 1978). This is due to the fact that untypical members share attributes with representatives of other concepts. But it's not that we do not notice these boundaries but that they cannot be found in nature.

2.2.4 Additional Differences

Abstraction is the most crucial part in classical concept formation. One can understand a classical concept by simply knowing the definition without ever knowing a member. In the mid-seventies, the usefulness of abstraction was challenged. In experiments of Reber (1976) and Brooks (1978) situations arose in which concepts could be learned without the knowledge of any abstraction rules. They were even hindering. A possible explanation would be that concepts are resembled by instances, not definitions.

Furthermore, Mervis & Rosch (1981) questioned whether concepts could be expressed by simply listing the defining properties. Properties as colour and shape can be treated as concepts, but cannot be cut up. One should examine the nature of attributes more closely. As mentioned, the attribute colour is a concept (Rosch 1973) but is often treated as an attribute. There is no clear cut concerning attributes and concepts. Easily we can make an attribute a concept, if we start to reason about this attribute.

Even though none of these deficiencies can prove classical concept formation as false. As we will see later on, the collection of the criticism proves it to be cognitively inadequate and one has to find better approaches. This is how prototype theory came about.

§ 3. Prototype Theory

The idea of the prototype approach is to overcome the problems of classical concept formation by representing a concept by its typical members. These objects have a special name.

Prototype:
We will call an object which is (in some way) typical of a group of objects a ***prototype***.

Concept membership is determined by comparing an object to the prototypes of the concept. This results in membership being a dimension, whereas the membership of an object is stronger the greater the similarity between this object and the prototypes.

Taken the concept birds Europeans regard a blackbird as the best representative, Americans mostly choose robins as a prototype. Classifying a raven we find the bird very similar to robins, even though they are not completely alike. But comparing it with other animals we believe it to be a true bird. Whereas it takes us more time to classify a penguin as a bird (and not everybody does this intuitively). The reason for our hesitation is that a penguin is also similar to a seal, which is a fish. But once again, it is not a typical fish. Since similarity to a bird is greater than to a fish we stick to the classification as a bird.

Thus, prototype theory dissolves the problems of classical concept formation: Members of a concept are no longer regarded as equal, instead we have a differentiated structure depending on the instances. Secondly, concepts do not need concept boundaries anymore. The paradox of the frog and the tadpole looses its contradiction. The tadpole becomes more and more dissimilar to the concept tadpole and gets more similar to a frog. And this is not a logical contradiction, but pure experience of nature.

Prototype theory does not demand an exact definition of the concept. Not all prototypes must have the same (to the concept important) attributes. It is easy to imagine typical chairs that have three or four legs, even though the number of legs is crucial to the concept of chair. It is not required that every prototype is very similar to every other prototype, but a prototype is similar to most of them. This is known as family resemblance.

Prototype theory resolves the deficiencies of classical concepts, but we have lost the easy handling of concept formation. We need new models to regain the practicability, so we can efficiently classify and reason about concepts. First, we will consider prototype theory in more detail, before we examine prototype models in chapter 5.

3.1 Family Resemblance

Since we do not have clear boundaries in prototype theory to determine the members of a concept, one can ask what causes a concept to be coherent. Prototype theory represents a concept by its most representative members. But these have to be determined considering the total set of all objects. If we have a notion of the concept, for example by a list of representative attributes, we can find the prototypes by determing the objects which are most similar to the given attributes. But one or a group of objects which fit the given attributes or are very similar to the attributes and <u>to each other</u> does not have to exist.

This problem can be avoided by family resemblance. Family resemblance means that no member of a group is similar to every other member of the group, but every member is similar to most of the others. For example, every member of a family is similar to most of the family members (that's where the name comes from). But it occurs that children look more similar to one parent. Nevertheless, one would recognize the group as a family. Applying this to prototype theory, there is a set of attributes which are typical for the concept and no prototype of the concept must have all attributes, but a prototype has most of them. This implies that there does not need to be one most typical prototype.

Pure prototype theory creates the new problem of how to decide which object is a prototype. This is due to the missing boundaries. The gradience of concept membership must also be applied to the choice of prototypicality. Otherwise we would reintroduce clear boundaries (at least concerning typical members). Regarding the quality of being a prototype as a gradient dimension solves our problem, if we have given the attributes a priori. If we want to form the concept out of given objects prototypes have to be chosen by certain qualities. For instance, let's determine similarity using a distance function in the space of attributes (later on we will see a more adequate similarity function). Then prototypical concepts are naturally given by clusters of objects, who have a great family resemblance but a great dissimilarity to other objects or concepts in the space. Objects in the middle of the cluster can be regarded as prototypes. We should mention that this is, for several reasons, only one way to model prototype theory. The prototypical approach does not imply it.

A criterion for prototypicality is the cue validity. It is the conditional probability that a property is shared by a concept member. It thus gives a clue how much a property implies concept membership. Cue validity is usually determined by counting the presence of a property in the set of concept members. Of course, due to different similarity values the presence of a property has to be weighted by the similarity of that object to the concept. On the other hand given a cluster of objects, one can determine the cue validity of every attribute and form a fictitious prototype. The cue validity for the attribute of having wings is very high for the concept birds, but for sure it is not one since we know birds that do not fly! Objects that have properties with high cue validity values, which are not necessarily one, are prototypes.

Rosch & Mervis (1975) conducted experiments to test if family resemblance occurs in natural categories. They chose superordinate level concepts and basic level concepts, as well as artificial ones. The results showed a high correlation between family resemblance and prototypicality.

> "The more prototypical a category member, the more attributes it has in common with other members of the category and the less attributes with contrasting categories. Thus, prototypes appear to be just those members of the category which most reflect the redundancy structure of the category as a whole." (Rosch & Mervis 1975, page 604).

Family resemblance gives us a mechanism to explain the internal structure of prototypical concepts, but it is not enough to explain, and this is due to the nature of prototype theory, how prototypes are determined. We will need more assumptions to make prototype theory practical. But if we lack clear boundaries, as well concerning the concept membership as the determination of prototypes, how do concepts correlate to other concepts?

3.2 Concept Hierarchies

Rosch et al. (Rosch, Mervis, Gray, Johnson & Boyes-Braem 1976) regard human concept formation as mostly predetermined, as pointed out above. Humans always build their concepts and the interrelations of concepts according to their environment in order to have least disagreements. They group objects which they experienced to belong together.

Rosch et al. showed through experiments that the three proposed levels of categorization are cognitively adequate, they form a correct model of human concept formation. The basic level proved to be the most important. Once again, their results: The basic level was the most abstract to support identification of perceived objects. Objects are first classified as members of basic concepts. Children are able to sort instances of basic level concepts before other levels. Names of basic level concepts are the most frequent words (in the English language) and are learned first by children. All experiments revealed the basic level to be the most dominant and supported its significance for (human) concept formation.

Similar results were found by Mervis & Crisafi (1982). They investigated closer the question which level is learned first. They proposed that the level with highest differentiation to other concepts would be the first to learn. Differentiation means the degree of similarity between instances of the own concept and the dissimilarity to instances of other concepts. Subordinate level concepts scored lowest differentiation. Similarity and dissimilarity grew less. Whereas basic level concepts had the highest score. The results of Mervis & Crisafi correlated to the model of three levels of categorization.

If humans (try to) model nature, the question is why do not they share identical concepts? Differences appear among people, but most clearly among cultures. This is due to the way they perceive their environment. First, they may ignore certain properties or interrelations. Secondly, they may overstress properties. Different information about the domain or special education change viewpoints of individuals. An example:

> "Airplane appeared to be the basic level for most of the students participating in our experiments. One subject, however, was a former airplane mechanic. His taxonomy was interesting. The lists of attributes common to airplanes produced by most subjects were paltry compared to the lengthy lists of additional attributes which he could produce. Furthermore, his motor programs as a mechanic were quite distinct for the attributes of the engine of different types of planes. Finally, his visual view of airplanes was not the canonical top and side images of the public; his canonical view was of the undersides and engines." (Rosch et al. 1976, page 430)

This is an example how different experience can alter one´s concepts. Nevertheless, continuing differentiation is limited. If more details, as the airplane mechanic knew, lower family resemblance and strengthen similarity to other concepts, only subordinate concepts are produced and basic levels stay basic.

Stressing of certain properties changes the interior structure of concepts. Humans do this when they look at prototypes. They list properties as defining which are actually shared by representative members but not by all instances of the concept. This is a major method to cut down complexity of the world. Concepts seem to be represented by prototypes. Only when caused to reason more about the concept they take nonrepresentative instances into account.

No matter how much concepts and concept hierarchies tend to differ, humans seem to use the same overall method to model their environment. The three types of hierarchy levels proved to be a good model of human concept formation when modelling natural concepts.

3.3 Similarity Measures

Prototype theory depends on the notion of similarity in order to classify objects. Even though it is not concerned about how similarity is calculated, it gives us a few hints about which properties similarity measures should fulfill. Usually, similarity measures are proposed to be commutative. But prototype theory criticized on classical concept formation that prototypes are more similar to other objects than vice versa. Thus, we need more cognitively adequate similarity measures. We would like to discuss, as an example, the proposed measure by Tversky & Gati (1978).

Their similarity measure is based on a comparison of properties. The similarity S(a,b) between object a and object b is calculated from the set of shared properties prop(a) $\cap$ prop(b), the set of properties, owned only by object a, prop(a) - prop(b), and the properties prop(b) - prop(a). Tversky & Gati set their similarity function to be a function of these three arguments (feature-matching function). As a prerequisite we need a function that maps a set of properties into the set of real numbers, e.g. a function that computes the cardinality, then we can define similarity as follows:

$$S(a,b) = \theta\, f(prop(a) \cap prop(b)) - \alpha\, f(prop(a) - prop(b)) - \beta\, f(prop(b) - prop(a))$$

with $\theta, \alpha, \beta \geq 0$

The parameters θ, α, and β are free to choose from the set of real numbers. For example, setting $\alpha = \beta = 0$ reduces similarity to a function of only the common attributes. Tversky & Gati conducted experiments to find out more about these parameters. Three major topics were investigated:

1) the difference between similarity and dissimilarity,
2) the asymmetry of similarity, and
3) effects of contexts.

Concerning the first topic, their results implied different parameters for similarity and dissimilarity. Humans stress common attributes, if estimating the similarity and vice versa, if computing the dissimilarity. Tversky & Gati also concluded from their experimental data that humans use similarity measures that are not commutative, thus supporting the asymmetry of comparing an object to a prototype. Similarity asymmetry can be easily modeled by ensuring that α is greater than β. The results concerning the last topic led to the conclusion that we use different parameters depending on the context. Thus, we do not have a fixed set of parameters, but vary our similarity functions. (For more details, see the article of Tversky & Gati 1978).

3.4 Application of Prototype Theory

The initial criticism on classical concept formation was the usage of artificial objects. Prototype theory proved disagreement among natural concepts and developed a different concept model. Examples of natural objects are chairs, vegetables, or letters. Then the question arose, if the model fits natural categories, does it also fit other concepts. Classical concept formation was not completely refuted, since it stays true for well-defined concepts as numbers. It was only proved that the classical method cannot be applied to natural concepts. But the other way around, can prototype theory be applied to other than natural objects? Following this, concepts as belief, psychological situations, emotions, linguistic concepts, concepts about scenes and many more were studied (for a summary refer to Medin & Smith 1984).

Cantor & Mischel (1979) studied concepts about persons such as arrogant or extrovert. They found family resemblance as a major reason for classification. Nevertheless, such concepts showed considerable differences: The taxonomy is not strictly ordered as the one of natural concepts. A person can belong to several concepts, being arrogant and helpful at the same time. Classification has effects on the person classifying as well as on the classified and it changes the expected behaviour. Properties used to classify are more abstract in comparison to natural concepts.

In summary, it seems promising that prototype theory can be used to model other types of concepts, but the preliminary results show that it has to be modified or extended to fit the new domains.

§ 4. Criticism and Conclusion

4.1 Criticism

As we have mentioned, in the early seventieth prototype theory emerged out of criticism on classical concept formation. Natural concepts proved to be nonarbitrary, boundless and structured. Human behaviour was used to conclude about cognitive representation. It is proved that we use (natural and many other) concepts in a prototypical fashion. But do we represent them this way? Armstrong, Gleitman & Gleitman (1983) were able to show that this conclusion is false.

Rosch asked whether the classical approach could be applied to natural concepts and discovered its inadequacy. Armstrong et al. did it the other way around: They applied prototype theory to well-defined, natural concepts. Do such concepts exist? Take the concept of even numbers as an example, it has a clear definition. Another example is the concept grandmother, defined by mother of a mother. Armstrong et al. conducted the same experiments with these concepts as Rosch did and found a prototypical structure. Humans did distinguish between typical and nontypical instances (which is the core of prototype theory), and prototypes were classified faster. Even afterwards, the subjects of the experiments confirmed the clear definitions of the concepts. Despite of their own knowledge subjects continued to treat these concepts prototypically in the following experiments.

What does this ambiguity mean? It seems that humans can use concepts as prototypical as well as classical. The results of the experiments by Armstrong et al. (1983) proved that humans are able to define the concept grandmother and differentiate as much a woman can be called a grandmother (judging by perceptual information). Abstractly speaking, we distinguish between classifying objects and reasoning about a concept. Armstrong et al. proposed that, on the one hand, we use an identification function to classify the objects, maybe by comparison with prototypes of that concept, and, on the other hand, we have a core to represent the concept abstractly, which could be a definition in case of well-defined concepts, and use this core, for example, in reasoning processes. This hypothesis could explain the results, but lacks any experimental grounds. Even if this hypothesis were true, why do not we use the concept core to classify objects, at least it would be correct. One explanation would be that it is easier to reason with the help of perceived properties (the outlook of a woman) than with abstract properties (having a child - this may be hard to find out). From the psychological viewpoint we have no hints.

Armstrong et al. criticism includes more. Rosch & Mervis (1975) asked their subjects to list all properties of a concept that they knew. But the fact that the list missed properties that were true for all instances, does not imply that they do not exist. Do we really mention all relevant properties or do we even know them? The answer is no.

First of all, we surpress certain properties. Talking about chairs we list quite a lot of attributes we believe to be important, but for sure, we do not think about chairs being physical objects. Implicitly we presume that the listener has similar experience as we do and knows these trivial facts. Secondly, are we able to express all attributes? This is more a linguistic question, whether our language is powerful enough to express our thoughts. Furthermore, are attributes sufficient to capture the notion of every concept? Do the attributes comprise the concepts meaning or is the whole more than the sum of its parts?

> "What looks like a box, smells like a lox, and flies? The answer is a flying lox-box." (Armstrong et al. 1983, page 303)

Why do we laugh about this riddle? The answer given is a simple combination of the expected properties, and that seems unnatural to us. Most concepts embody more. And the philosophical questions is, whether properties suffice to express the full meaning of concepts. (For more criticism on the prototype theory see Armstrong et al. 1983, Medin & Smith 1984).

4.2 Conclusion

The results of the experiments that led to prototype theory are stable. They have been verified and not been challenged until now. But what do they say?

The experiments showed that human concepts of natural objects are well-structured. We do not create them arbitrarily. They are suggested by our environment, especially nature. Furthermore, we do not regard all instances as equal in their rights. We have a kind of degree as much an object is a "good" example of a concept. We do not draw clear-cut lines between different concepts. All these results proved classical concept formation as cognitively inadequate and showed the need for a new theory.

Yet, these result do not imply a specific theory. They simply show how we understand and use concepts, but they do not state at all how we represent them. The wide variety of psychological models we will discuss in the next chapter supports this fact, since most of them explain the results. The prototype theory is no more and no less than a criterion as far a model for human concept formation is cognitively adequate. A model for concept formation must support prototype theory to be adequate, but may have more features or differ considerably in concept representation. One example for slightly different results on human concept formation is the article of Armstrong et al. (1983).

Prototypical representation is not implied by prototype theory. It is even unknown, if attribute representations will suffice. It seems reasonable that experience with objects as motor programs contribute to the notion of a concept. Neither does prototype theory imply that basic level concepts are atomic concepts. There must be concepts that are relatively simpler than other concepts and treated differently. But how they themselves look like is left open.

Even if the prototypical approach is not necessarily the one and only adequate explanation of human concept formation, it seems to be a promising way of simulating human cognition. Though, prototype theory is only a framework that has to be filled in with concreteness, if it is supposed to be simulated on a computer. We will cover the question of how to accomplish this in the following chapters of this paper.

§ 5. Prototypical Models

We will give a short survey of models which explain, more or less, prototypical behaviour. It is not our aim to be complete or discuss the models thoroughly, but to give an overall impression on what has been done to find good prototypical models. It should be mentioned that not all of these models were invented to explain prototype theory completely.

5.1 Array Models

Estes (1986) proposed three kinds of models which were called exemplar-memory models, feature-frequency models, and prototype models. The differentiation refers to the way concepts are represented and objects are classified, whereas the title "array model" points out that objects are represented as arrays of attributes which is less important to our discussion.

Exemplar-memory models store all instances of a concept in order to denote the concepts meaning. New objects are classified by computing the similarity to all instances of that concept. All similarities are summed and put into relation to the accumulated similarities of other concepts. One possibility is to calculate the probability of being an instance, as Estes did. The object is classified as an instance of the concept which scores the highest value.

Feature-frequency models count the occurence of an attribute in the set of instances. These are divided by the total number of instances to receive the relative frequency of an attribute. The product of all frequencies is equal to the probability that the object is an instance of the concept. Again, the object is classified by the maximum likelihood principle. Similar models were discussed by Neumann (1974), who proposed a more general feature-frequency model, and by Hayes-Roth & Hayes-Roth (1977), who discussed another variation.

Estes prototype model represents a concept by creating a prototype whose attributes reflect the relative occurence of each attribute in the set of instances. The similarity of a new object to a concept is computed solely by comparing it to the prototype and the object is classified analogously to the exemplar-memory model. The prototype represents the central tendency of the concept and may change as the concept changes.

All kinds of models show prototypical behaviour when classifying new objects. Nevertheless, there are several deficiencies concerning prototype theory: First of all, concepts are not allowed to share objects, and thus they have clear concept boundaries. Additionally, the model gives no explanation how the nonarbitrariness of natural concepts could be reflected. Finally, they give no hint about the relationships of concepts (as the three level hierarchy). Even though these models explain an important part of prototype theory, they need to be expanded to cover all of it.

5.2 Distributed Models

Hintzman (1986) discussed the multiple-trace memory model. In such a model every occurence of an object is experienced as unique and stored separately in the long-term memory (as a memory trace). When the mental process attempts to classify a new object, it sends the objects representation to the long-term memory. In parallel every memory trace is activated according to its similarity to the stimulus and returns an echo. The echos are combined by overlapping them. In detail, the combination of the echos is computed by weighing the echos according to the intensities of the individual echos.

Concepts are formed by grouping instances which have a high family resemblance. Thus, these instances score high intensities when a prototype of that concept is presented. On the other hand, nonrepresentative examples score low intensities for that concept and may reach similar or higher values concerning other concepts. This explains prototypical classification and realizes missing boundaries. But the multiple-trace memory model gives no hint at how the concepts are formed and how they interrelate. Once again, this approach would have to be expanded to cover prototype theory.

McClelland & Rumelhart (1985) chose a connectionistic approach to concept learning. Their model consists of many simple, widely connected units, that are able to store one value and send its value to connected units. A weight is associated to every connection between units. At every cycle a unit receives a net input equal to the sum of the signals multiplied by the weights. Positive input values increase the value of a unit, negative decrease it in a range of -1 to +1. Additionally, every cycle the value is decreased by a small factor. Units are organized into moduls, in which they have a higher degree of connection.

In order to classify objects, a modul has to learn how to reconstruct the representation of a prototype when it receives the input of an instance. This is accompilshed by adjusting the weights which incorporate the knowledge of the net. McClelland & Rumelhart discuss a method, called delta-rule, that achieves the learning. (This approach is similar to pattern recognition with neural networks.) Concepts can be learned fairly well, if the prototypes are sufficiently different, and the net can learn its concepts incrementally. Apart from that, the criticism on this model is mostly the same as on the multiple-trace memory model of Hintzman.

5.3. Concept Learning in AI

In his paper "Learning structural descriptions from examples" Winston (1975) describes a method to learn visual concepts. The idea is to find "near misses" that fail to be instances by only few attributes. But the notion of a near miss assumes classical concept formation. In his own words:

> "To begin with I want to make clear a distinction between a description of a particular scene and a model of a concept. A model is like a ordinary description in that it carries information about the various parts of a configuration, but a model is more in that it exhibits and indicates those relations and properties that must and must not be in evidence in any example of the concept involved." (p. 156).

Sowa (1984, chapter 4.1) represented concepts as graphs with the nodes being concepts or concept relations. Concept graphs may have formal parameters as nodes that are filled out when the concept is instantiated. Prototypes are typical instantiations of concepts. Sowa's concepts are fairly classical. They are represented by a definition, the conceptual graph. Sowa loosens this restriction, as he explains with the help of the concept bus: "Although the relationships ... are commonly true for buses, they may be violated in any particular case." (p. 130) But it is questionable, if a definition should be given at all having so many violations as in the case of natural concepts. Sowa has relaxed classical concepts, but it seems doubtful whether that will suffice to cover prototype theory.

Bobrow & Winograd (1977) shared some ideas of prototype theory when they invented the "knowledge representation language" (KRL). Concepts are defined as units which consist of descriptions. The "description matcher" classifies an object by comparing it to the units which play the role of a prototype, the most typical instance. Nevertheless, this description matcher reintroduces conceptual boundaries when returning "match" or "fail" (p. 273) as the result of one comparison. On the other hand, objects are allowed to be in different concepts according to the context. A concrete implementation would have to determine as far KRL covers prototype theory (which was not the aim of the article by Bobrow & Winograd).

Stereotypes are similar to prototypes. Rich (1989) discusses the use of stereotypes to model users in dialogue systems. At the start of the dialogue the system has only few information about the user. To compensate this lack they make assumptions about the user by assigning him to stereotypes. These stereotypes are triggered by words or sentences the user utters. For example, the mentioning of costs suggests a business man. The additional information by stereotypes can be used to make inferences. But this information has simply default character. It has to be withdrawn whenever further conversation proves the triggering of the stereotype to be false. This demonstrates one aspect of prototype application: Since, as we have seen, prototype classification are of degree or have to be taken back, one can not use classical logic to make inferences, but has to take into account that classifications of properties of concepts are not objectively true. Of course, default logic is only one way to deal with this kind of uncertainty.

§ 6. Implementation

6.1 Reflections on Implementations

If we want to program prototypical concept formation on a computer, we have quite a lot of freedom, but also some system restrictions. There are three main aspects of such a program:

1) How do we simulate prototypical concept formation in the first place?
2) How can we make use of prototypical concept classifications and hierarchies? (i.e. what kind of inferences are possible?)
3) Which kind of information do we need to achieve powerful concept formation?

6.1.1 Prototypical Concept Formation

Prototype theory says that instances of a concept are not equal. To model this property we need more information about the object than its mere concept membership. Either we extract values as the degree of membership or we know the object as a part of our world which seems to be more realistic. Furthermore, we have to model the internal structure of a concept implied by the inequivalence of its members which is closely related to the ability to compare objects. We do not necessarily use a similiarity measure, but we have some notion how objects relate. The experiments of Tversky & Gati (1978) have shown that this notion has certain properties, as asymmetry.

A second point is how to model the concept itself. We can not use any representation that would denote a set, since concepts miss exact boundaries. One alternative would be to store the central tendency, a generalized prototype, and define concept membership as the similarity to the central tendency. In any case, we can not compute concept membership to be true or false anymore, we need some other semantics to express classification. Especially, as we have seen, classification may change depending on the context.

The unarbitrariness raises the question of how to find a conceptual copy of nature's structure that suits our purposes. Human concepts are not arbitrarily set, but humans are oriented towards reality. Nevertheless, that can not be enough. Concept formation also depends on the person's point of view. Put another way, one's state contributes to one's resulting concepts. On the one hand, we will have superior mental processes who demand certain concepts to be formed. And on the other hand, this is the more interesting aspect, we have to detect the structure that inhibits the total set of objects.

Concept hierachy is another aspect of prototype theory we would have to model. Especially Rosch and her associates found out that we primarily use three levels of concepts, whereas the middle level is the most basic one. In this hierarchy we are able to classify objects at different levels, depending highly on our (changing) mental context. For example, we regard a dime to be a coin, simply money or a means of payment, being coin subordinate, money the basic level, and means of payment superordinate, more general. Classification of a dime changes as context changes: Using a dime for payment, we would call it some means of payment. But talking about metals, a dime is an ordinary coin.

6.1.2 Usage of Prototypical Concepts

Since prototypical concepts miss classical structure we can not use them classically. First, we can not determine concept membership to be true or false. Secondly, concept membership changes depending on the context we are in. The point is not that we could not decide concept membership - we could draw an artificial line somewhere, but that it would be cognitively inadequate to do so. It is human mental strength to be flexible.

In any case, we have to decide concept membership somehow if the system should be of any use. But viewing membership as a degree could clutter up our superordinate mental processes with too much numeric data. It seems reasonable to find a flexible semantics that would be a compromise between the extreme of giving a detailed description as a result and the extreme of simply saying yes or no. This semantics will have an impact on the superordinate processes. Nonmonotonic reasoning seems appropriate to deal with prototype systems. Additionally, it could be an improvement to take prototypes as representatives and to base the reasoning process on their qualities, even though this might be incorrect.

6.1.3 Knowledge Representation

Concept formation is based on the ability to store knowledge. Its power depends strongly on the power of knowledge representation. Humans understand concepts as time, casuality, and processes. We can not expect a system to reason about concepts as seats, functional objects, or means of transportation, if we miss the notion of time or events. Prototype theory does not need such fundamental concepts to work. In contrast, it is questionable if these concepts can be represented by prototypical concept structures. Prototype concepts can be build without fundamental concepts. But if we compare any prototype system to the abilities of human thinking, we have to keep in mind that human beings use these concepts.

The concept of a bicycle is a rather simple concept to humans. The concept bicycle needs as a prerequisite the notion of moving which implies time and space to be fully understood. Such concepts as bicycle embody the notion of functionality which we understand perfectly because we have made the experience of using an object through our actions. As long as a concept formation system does not have these understandings of fundamental concepts as we experience them every day one can not expect a concept system to be as powerful as human concept formation. We have to keep this fact in mind, if we judge the results of prototype theory.

6.2 Our Implementation

We implemented a prototypical model for concept formation in Common Lisp. Since prototype theory misses the easy handling, we asked ourselves how practical such a system could be. Our aim was not to write an efficient and in various domains usable program, but to answer this question in principle, because we do not know a system which already implemented prototype theory as a whole. Thus, the objects that constitute the input to the system may appear too simple or boring, but they suited our purpose well. On the other hand, our model leaves many questions open which could only be answered by testing it in a more realistic scenario. This is especially true for questions concerning the tuning of the system. Again, we did not aim at such a thorough examination. Tuning and similar problems, anyhow, may vary according to the domain the system will be used in.

Furthermore to keep things simple, we regarded concept formation as a process of classifying perceived objects. We left out all aspects of building concepts on internal states or thoughts, as well as any influence of other mental processes. Our model is concerned with classifying natural objects without a teacher, the conceptual structure has to be found without any help. The general idea is that the system receives the representation of an object as input and has to classify the object. During continuous classifications the system develops a conceptual structure of the world (which is made up of its input). Since we excluded any superior processes, the system reaches its goal if the created concepts reflect prototype theory.

In the following we will describe how the model looks like and in chapter 6.3 we will discuss our experience testing the model.

6.2.1 Representation of Objects and Their Similarity

Object Representation:

Objects are represented by a list of boolean attributes, an object has a property or it does not, but there exists no dimension of an attribute.

```
(Vogel1
        (ATTRIBUTE fliegen)
        (ATTRIBUTE fuesse)
        (ATTRIBUTE ohren)
        (ATTRIBUTE schnabel)
        (ATTRIBUTE augen)
        (ATTRIBUTE fluegel)
        (ATTRIBUTE federn))
```

Figure 1: Example of an Object

Figure 1 shows an object representation which consists of its (meaningless) name and a list of attributes.

The system itself treats an object as a unit, it does not know how the object is represented. It can only use a function to compute the similarity between two objects. This similarity function determines a measure as it was proposed by Tversky & Gati (1978). The cardinalities of the three sets which include shared attributes and the attributes owned by only one object are each multiplied by a system parameter and added to yield the following sum (if f denotes the cardinality function):

$$S(A,B) = \theta\, f(prop(A) \cap prop(B)) - \alpha\, f(prop(A) - prop(B)) - \beta\, f(prop(B) - prop(A))$$

This value is mapped onto the range of [0,1]. Thus, the system can compute the similarity of two objects which is between 0 (meaning no similarity) and 1 (being identical).

The model is independant of object representation. It is able to classify any object if supplied with an appropriate similarity function. Though, the usefulness of the similarity function has a high impact on the performance of the model. The similarity should make "sense" if the produced conceptual structure is supposed to be natural. Being more precise, the similarity function is a parameter value to the system: The conceptual structure is a function of the similarity function. Depending on the domain the similarity function may be a major problem. It may be challenging to find a "good" similarity function for special object representations. It should be mentioned that we could use a different similarity function than the one of Tversky & Gati, but it should have the properties prototype theory requires.

6.2.2 Concept Representation

We modeled a concept by its central tendency.

Central Tendency:

The ***central tendency*** is the midpoint according to the similarity function of all prototypes (of that concept).

The representation of the central tendency depends only on the object representation, the model simply requires that the similarity function is extended to be also applied to central tendencies and object pairs. A central tendency is produced whenever a concept is created, namely by clustering (see later on). The central tendency is treated as the most typical instance, actually it is an artificial prototype (for an example see figure 2).

```
(CT
      (ATTRIBUTE fliegen        0.71)
      (ATTRIBUTE fuesse         1.0)
      (ATTRIBUTE ohren          1.0)
      (ATTRIBUTE schnabel       1.0)
      (ATTRIBUTE augen          1.0)
      (ATTRIBUTE fluegel        1.0)
      (ATTRIBUTE federn         0.85))
```

Figure 2: An Example of a Central Tendency

Concept membership is decided by comparison with the central tendency. We define an object to be a prototype of the concept if it scores a similarity higher than a parameter ASSO-MIN1 (prototype similarity, used for classification), i.e. 95 percent. The idea of the parameter is that its value is changed according to the system's requirements, even though right now we have no idea how this could be done (this is also true for all other parameters that will be introduced later on).

This kind of concept representation ensures prototypical behaviour of the model. By determing a "midpoint" of the concept we leave out any concept boundaries which can be set at any point or which can be omitted at all as the system marches on. Secondly, instances remain objects of the domain because we keep their represenation. In fact, a concept is only a data structure including a central tendency and pointers to objects that have been classified as members. So we always keep in touch with the object itself (and the model keeps in touch with reality - as it perceives reality). At any time the system can reclassify any object, restructure the whole concept or switch between the conceptual and object level of reasoning.

6.2.3 Concept Hierarchies

Concepts are units in their own rights, they only exist because we have found objects that belong together. This set of concepts is organized in an acyclic graph which represents the conceptual hierarchy of the concepts (examples are given in appendix B). Whenever we have a link between two concepts, one is denoted to be the more general concept. Being more general is defined as the fact that the central tendency of the subordinate concept is a prototype of the superordinate concept, the superordinate concept **subsumes** the other one. Due to the structure of prototypical concepts and the fact that central tendencies are created out of a set of objects we cannot expect any attributes to be inherited. It may happen that attributes are common in the subordinate concept that are also common on the superordinate concept, but since attributes do not even have to be shared by all members they are not necessarily shared by the members of the subordinate instances.

The graph need not be a tree. A central tendency may be similar to several concepts resulting in several superordinate concepts. The only restriction we made was that the graph should be acyclic which means no concept is a subordinate concept of itself. In the implementation of our model we did not ensure the conceptual hierarchy to be complete, it is only correct. Completeness was not supplied, even though it may have been (in the small test domain), because it is not clear whether completeness is needed or wanted (huge concept structures would make completeness a costly property). As the model is right now, it would be no problem to add this feature.

6.2.4 Classification

When the system is asked to classify an object it searches through the concept hierarchy to find the concept(s) the object is most similar to. The algorithm starts with an initial set of concepts which are chosen either by the system or the "user". For every concept it computes the object's similarity and the similarities of upper and lower concepts (see figure 3). This cycle continues until all remaining concepts have a similarity lower than CLASS-MIN or their difference to the present maximum is greater than CLASS-RANGE. The system performs a kind of search guided by the similarity function.

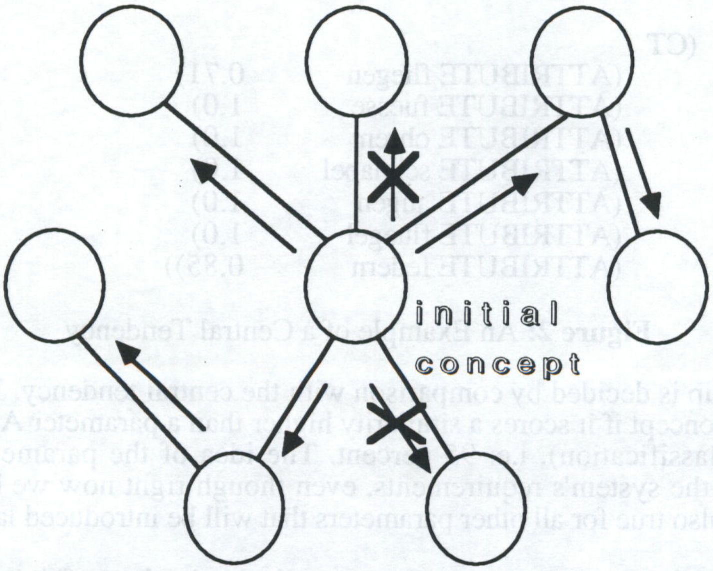


Figure 3: Path through the Hierarchy while Classifying

At the end we have a set of concepts with their corresponding similarity values. First, we build the set of concepts that have a value greater than PROT-SIM (prototype similarity), maybe 80%, meaning the object is a prototype of the concept. If concepts subsume other concepts according to the hierarchy, the concepts with lower values are removed from the set. The object is assigned to all of the remaining concepts. If the set is empty, concepts with a value higher than INST-SIM (instance similarity), for example 50%, are gathered. The idea is that the object could be an (untypical) instance. Again, we remove subsuming concepts and assign it to all concepts. If the last set was also empty, the object is assigned to a dummy concept which collects all objects that have not been classified yet.

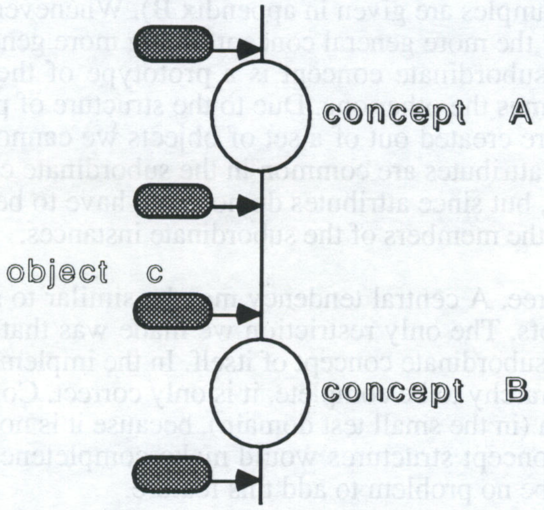


Figure 4

Why can we expect to find all concepts the object is very similar to? Or could it happen that we miss the most similar concept? We want to make it plausible why the search method will work in a "natural" hierarchy, even though this is no prove and we do not know if it would work in any (natural) concept hierarchy. Assume we have two concepts A and B with attribute sets prop(A) and prop(B), whereby prop(A) is a subset of prop(B). Thus, A is a superconcept of B (see figure 4).

Now we want to classify the object c with attribute set prop(c). Our similarity function computes two values:

$$S(A,c) = \theta\, f(prop(A) \cap prop(c)) - \alpha\, f(prop(A) - prop(c)) - \beta\, f(prop(c) - prop(A))$$
$$S(B,c) = \theta\, f(prop(B) \cap prop(c)) - \alpha\, f(prop(B) - prop(c)) - \beta\, f(prop(c) - prop(B))$$

We abbreviate:

$$\theta_1 := \theta\, f(prop(A) \cap prop(c)),\quad \alpha_1 := \alpha\, f(prop(A) - prop(c)),\quad \beta_1 := \beta\, f(prop(c) - prop(A))$$
$$\theta_2 := \theta\, f(prop(B) \cap prop(c)),\quad \alpha_2 := \alpha\, f(prop(B) - prop(c)),\quad \beta_2 := \beta\, f(prop(c) - prop(B))$$

If c belongs to a subordinate concept of B, then prop(B) is a subset of prop(c), implying $\theta_1 < \theta_2$, $\alpha_1 = \alpha_2 = 0$ and $\beta_1 > \beta_2$, resulting in $S(A,c) < S(B,c)$ and c being classified as B. Vice versa, if c belongs to a superordinate concept of A, we have $\theta_1 = \theta_2$, $\alpha_1 < \alpha_2$ and $\beta_1 = \beta_2 = 0$, and c gets classified as A. If c is in the middle of A and B, classification depends on prop(c) having only a few attributes more than prop(A) or only few less than prop(B). In the first case, we have $\theta_1 \approx \theta_2$, $\alpha_1 = 0$, $\alpha_2 > 0$, $\beta_1 \approx 0$, $\beta_2 = 0$, and c is classified as a member of A. In the second case, c gets classified analogously as a member of B.

As we have argued above, hierarchical links do not fulfill the condition that the attributes of the subordinate concept are a superset of the superordinate attributes. Though it is probable that in a natural environment we will have a hierarchy that will come close to such an extreme case, as discussed above. In such an *almost perfect* hierarchy classification will yield very similar results, and we expect this to be the case in natural domains. Another crucial point is the initial set. Obviously this set has an influence on the performance of the classification. It is desirable that this set contains the basic level concept of the hierarchy, even though at present it is not clear how basic level concepts are detected.

6.2.5 Clustering

Whenever a certain number of objects (INSERT-MIN) have been inserted into a concept, we check if the concept can be divided into more specialized concepts. A clustering algorithm is called which depends solely on the similarity between objects. The similarity it uses has its own parameters in order to control clustering independantly. If two objects have a similarity greater than CLUST-PROT (a parameter similar to PROT-SIM), they are assumed to be in the same concept. The transitive cover implied by CLUST-PROT is determined, and if its cardinality is greater than CLUST-MIN a new concept is created. (Notice: the relation implied by CLUST-PROT is not transitive itself.) All objects that are not member of a new cluster are classified in the set of the new concepts.

The new concepts are inserted into the hierarchy as upper or lower concepts. Succesive clustering may create a concept several times, resulting in the hierarchy to diverge. To avoid this the central tendency of every new concept is classified and concepts are merged for which the central tendency was classified as a prototype. Thus the hierarchy becomes stable, if known objects are inserted repeatedly. The fact that concepts may be merged may create concepts with more than one upper concept, destroying the tree-like structure of the hierarchy.

6.3 Test Results and Discussion

The program was tested with a scenario of birds. Even though the names of the objects are very suggestive, one should keep in mind that the system only takes into account the representation of objects and these representations are not equivalent to what we know as birds. It was easier to work with names than some abstract labelling, but one should not push similarity to far. Our criterion of success was whether the system would find the structure of the test set we created.

The test set was built out of prototypes which are listed in appendix A. Several prototypes were created which stood in relations due to the sharing of attributes. For every prototype, objects were instantiated which owned most attributes of the prototype, but lacked some by probability which differed for every attribute. This ensured the total set of objects (that did not include the prototypes) to have a structure which is a required property according to prototype theory to be a natural set of objects. In total 335 objects were created and presented to the system successively in mixed order. The performance of the system was measured by the resulting hierarchy of concepts and the classification of each object, while the order of object presentation was changed on every test run.

The usual course of a test run was that the first objects were classified as unknown until the dummy concept got clustered and new concepts emerged (appendix B gives some stages of a test run). After each clustering, classification kept going until another concept was clustered, usually resulting in a change of the concept hierarchy. The hierarchy was expanded until it became stable. Stability was finally reached when the presented objects differed only slightly from the objects already classified.

At the end of each test run the system had received the same input of objects, but in totally different order. Despite this, it was able to construct every time the same concepts, being stable after a sufficiently large input. The intended hierarchy was partly discovered. The system did not create all expected links. This may be due to the insufficiency of the similarity function or due to the structure of the input set. In total the system did a good job and the test runs proved, what we expected, that the model would reconstruct the structure of the input set if it could be detected by the similarity function.

6.3.1 Prototype Theory

The model was developed to simulate prototypical concept formation. Thus, the question is: Does it reach its goal? Objects are treated as members in their own rights. It is just remarked that they are instances, they keep their internal representation. Combined with the similarity, this creates an individual structure for every concept. Unclear boundaries are simulated by grades of concept membership. The model defines a lower limit for prototypical, conceptual and untypical similarity with the notion of being very similar, in this case it must be a prototype, being similar, meaning the object is a good example (but does not embody all ideas of the concept), and being less similar than dissimilar, having a high probability of belonging to a different concept. If these limits were used rigidly to distinguish these three sets, few would have been gained. If the superior process regards these sets as abstractions to use if detailed insight is of no interest, prototype behaviour emerges. Whether the process may change the parameters or the system itself searches for a constellation that results in a most "suitable" concept structure is left for future work. Even though we reintroduced numerical limits, this should not be mistaken for conceptual boundaries, since even if the values were rigid, an object could be classified as a member of several concepts.

Nonarbitrariness is found and reflected in the concept hierarchy, if it could be expressed by the similarity function. In that case, classification and clustering produce a hierarchy that reflects this structure. Of course, in our example the three level hierarchy as proposed by prototype theory was not discovered because the test set did not have this structure. Viewing the resulting hierarchies, it seems promissing that the system would detect such a hierarchy structure, if, again, it could be expressed by the similarity function. In addition, the notion of basic level concepts could be exploited to start classification at such concepts. One should mention that the system does a poor job, if the total set has none or little structure. In such a case, the clustering algorithm would find only few concepts which would have high cardinality, resulting in a small hierarchy from which only few information about the total set could be retrieved. But no structure means arbitrariness which should not be the case in natural concepts the system was designed to model.

6.3.2 Importance of Similarity

The expressive power of the similarity functions has a great impact on the performance of the model, as we have seen so far. It is used by the classification and the clustering algorithm. One aspect is that it influences the time complexity of the program. It should be fairly easy to compute. Especially the clustering algorithm is a quadratic function of the similarity function. Classification is only linear because each concept has to be visited only once. All other computations are neglectable. The complexity of the whole system depends solely on the similarity function.

But the similarity is also a bottle-neck concerning the semantics of the system. Any structural decisions are based on the similarity of two objects (or prototypes). No matter how similar we experience two objects, the system's behaviour is determined by the value the similarity function computes. It is crucial to our approach, if the measure of Tverky & Gati (or any measure at all) can express what we intuitively find similar. If the similarity does not reflect our notion of it the system could not simulate human concept formation.

6.4 Conclusion

We reached our goal to show that it is in principle possible to implement prototypical concept formation. The results promise that this approach can be improved to become practically useful. Nevertheless, it is obvious that more theoretical work should be done, especially on the problem of similarity, and the model should be applied to a more challenging domain. A second direction would be to extend the model with specialized concepts as time, space or functionality to improve its expressive power. A system that is supposed to form humanlike concepts must know these concepts. Prototype theory is one promissing step towards understanding and simulating humanlike concept formation.

Bibliography

Armstrong, S. L., Gleitman L. R., Gleitmann, H. 1983. What some concepts might be. *Cognition* 13:263-308

Bobrow, D. G., Winograd T. 1977. An Overview of KRL, a Knowledge Representation Language. See *Brachman & Levesgue 1985*, 264-285.

Brachman, R. J., Levesque, H. J. eds. 1985. *Readings in Knowledge Representation.* California: Morgan Kaufmann Publishers, Inc.

Brooks, L. R. 1978. Nonanalytic concept formation and memory for instances. See *Rosch & LLoyd 1978*, 169-215.

Bruner, J. S., Goodnow, J. J., Austin, J. G. 1956. *A Study of thinking*. New York: Wiley.

Cantor, N., Mischel, W. 1979. Prototypes in person perception. *Adv. Exp. Soc. Psychol.* 12:3-52

Estes, W. K. 1986. Array models for category learning. *Cognitive Psychol.* 18:500-549

Hayes-Roth, B., Hayes-Roth, F. 1977. Concept learning and the recognition and classification of exemplars. *J. Verbal Learning and Verbal Behavior* 16:321-338

Hintzman, D. L. 1986. "Schema abstraction" in a multiple-trace memory model. *Psychol. Rev.* 93:411-428

Kobsa, A., Wahlster, W. 1989. *User models in Dialog systems*. Springer Verlag.

McClelland, J. L., Rumelhart, D. E. 1985. Distributed memory and the representation of general and specific information. *J. Exp. Psychol.: General* 114:159-188

McClocksey, M., Glucksberg, S. 1978. Natural categories: Well defined or fuzzy-sets? *Memory & Cognition* 6:462-472

Medin, D. L., Smith, E. E. 1984. Concepts and concept formation *Annual Rev. Psychol.* 35:113-138

Mervis, C. B., Crisafi, M. A. 1982. Order of acquisition of subordinate-, basic- and superordinate-level categories. *Child Dev.* 53:258-266

Mervis, C. B., Rosch, E. 1981. Categorization of natural objects. *Annual Rev. Psychol.* 32:89-115

Neumann, P. G. 1974. An attribute frequency model for the abstraction of prototypes. *Memory & Cognition* 2:241-248

Reber, A. S. 1976. Implicit learning of synthetic languages: The role of instructional set. *J. Exp. Psychol.-Hum. Learn. Mem.* 2:88-94

Rich, E. 1989. Stereotypes and user modeling. See *Kobsa & Wahlster 1989*, 35-51

Rosch, E. 1973. Natural Categories. *Cognitive Psychol.* 4:328-350

Rosch, E. 1975. Cognitive representations of semantic categories. *J. Exp. Psychol.-General* 104:192-233

Rosch, E., Lloyd, B. B., eds. 1978. *Cognition and Categorization.* Hillsdale, NJ: Erlbaum.

Rosch, E., Mervis, C. B. 1975. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychol.* 7:573-605

Rosch, E., Mervis, C. B., Gray, W., Johnson, D., Boyes-Braem, P. 1976. Basic objects in natural categories. *Cognitive Psychol.* 3:382-439

Sowa, J. F. 1984. *Conceptual structures: Information processing in mind and machine.* Addison-Wesley.

Tversky, A., Gati, I. 1978. Studies of similarity. See *Rosch & Lloyd 1978*, 79-98

Winston, P. H. 1975. Learning structural descriptions from examples. See *Brachman & Levesque 1985*, 142-168.

Appendix A

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
;;;            Central Tendencies
;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;             the animal world
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;             birds

(setq ct-vogel  '(CT
                  (ATTRIBUTE fluegel    1.0)
                  (ATTRIBUTE augen      1.0)
                  (ATTRIBUTE schnabel   1.0)
                  (ATTRIBUTE ohren      1.0)
                  (ATTRIBUTE fuesse     1.0)
                  (ATTRIBUTE federn     0.9)
                  (ATTRIBUTE fliegen    0.8)))

; sparrows

(setq ct-drossel '(CT
                  (ATTRIBUTE fluegel    1.0)
                  (ATTRIBUTE augen      1.0)
                  (ATTRIBUTE schnabel   1.0)
                  (ATTRIBUTE ohren      1.0)
                  (ATTRIBUTE fuesse     1.0)
                  (ATTRIBUTE federn     0.9)
                  (ATTRIBUTE fliegen    0.8)

                  (ATTRIBUTE mittel                 1.0)
                  (ATTRIBUTE gedrungen              1.0)
                  (ATTRIBUTE kraeftigeBeine         1.0)
                  (ATTRIBUTE Hornplatten            1.0)
                  (ATTRIBUTE gebogenerSchnabel      1.0)
                  (ATTRIBUTE singvogel              1.0)
                  (ATTRIBUTE mittelSchwanz          1.0)))

(setq ct-star    '(CT
                  (ATTRIBUTE fluegel    1.0)
                  (ATTRIBUTE augen      1.0)
                  (ATTRIBUTE schnabel   1.0)
                  (ATTRIBUTE ohren      1.0)
```

```
                (ATTRIBUTE fuesse     1.0)
                (ATTRIBUTE federn     0.9)
                (ATTRIBUTE fliegen    0.8)

                (ATTRIBUTE kurzschwanz      1.0)
                (ATTRIBUTE gedrungen        1.0)
                (ATTRIBUTE langfluegelig    1.0)
                (ATTRIBUTE duennerschwanz   1.0)
                (ATTRIBUTE kraeftigBeine    1.0)
                (ATTRIBUTE langerSchnabel   1.0)))

(setq ct-meise      '(CT
                (ATTRIBUTE fluegel    1.0)
                (ATTRIBUTE augen      1.0)
                (ATTRIBUTE schnabel   1.0)
                (ATTRIBUTE ohren      1.0)
                (ATTRIBUTE fuesse     1.0)
                (ATTRIBUTE federn     0.9)
                (ATTRIBUTE fliegen    0.8)

                (ATTRIBUTE spitzerSchnabel  1.0)
                (ATTRIBUTE kraeftigeBeine   1.0)
                (ATTRIBUTE scharfeKrallen   1.0)
                (ATTRIBUTE kurzeSchwingen   1.0)
                (ATTRIBUTE kurzerSchwanz    1.0)))

(setq ct-kuckuck '(CT
                (ATTRIBUTE fluegel    1.0)
                (ATTRIBUTE augen      1.0)
                (ATTRIBUTE schnabel   1.0)
                (ATTRIBUTE ohren      1.0)
                (ATTRIBUTE fuesse     1.0)
                (ATTRIBUTE federn     0.9)
                (ATTRIBUTE fliegen    0.8)

                (ATTRIBUTE langeFluegel     1.0)
                (ATTRIBUTE langerSchwanz    1.0)
                (ATTRIBUTE mittel           1.0)
                (ATTRIBUTE fluechtig        1.0)
                (ATTRIBUTE aussenzehe       1.0)))

(setq ct-papagei  '(CT
                (ATTRIBUTE fluegel    1.0)
                (ATTRIBUTE augen      1.0)
                (ATTRIBUTE schnabel   1.0)
                (ATTRIBUTE ohren      1.0)
```

```
                (ATTRIBUTE fuesse    1.0)
                (ATTRIBUTE federn    0.9)
                (ATTRIBUTE fliegen   0.8)

                (ATTRIBUTE dickerSchnabel       1.0)
                (ATTRIBUTE kraeftigerSchnabel   1.0)
                (ATTRIBUTE Wachshaut            1.0)
                (ATTRIBUTE kraeftigeFuesse      1.0)
                (ATTRIBUTE farbig               1.0)))

(setq ct-falke    '(CT
                (ATTRIBUTE fluegel   1.0)
                (ATTRIBUTE augen     1.0)
                (ATTRIBUTE schnabel  1.0)
                (ATTRIBUTE ohren     1.0)
                (ATTRIBUTE fuesse    1.0)
                (ATTRIBUTE federn    0.9)
                (ATTRIBUTE fliegen   0.8)

                (ATTRIBUTE mittel          1.0)
                (ATTRIBUTE spitzeFluegel   1.0)
                (ATTRIBUTE Reisshaken      1.0)
                (ATTRIBUTE gedrungen       1.0)
                (ATTRIBUTE Faenge          1.0)
                (ATTRIBUTE starkeFluegel   1.0)))

(setq ct-bussard  '(CT
                (ATTRIBUTE fluegel   1.0)
                (ATTRIBUTE augen     1.0)
                (ATTRIBUTE schnabel  1.0)
                (ATTRIBUTE ohren     1.0)
                (ATTRIBUTE fuesse    1.0)
                (ATTRIBUTE federn    0.9)
                (ATTRIBUTE fliegen   0.8)

                (ATTRIBUTE Faenge          1.0)
                (ATTRIBUTE mittel          1.0)
                (ATTRIBUTE starkeFluegel   1.0)
                (ATTRIBUTE breitFluegel    1.0)
                (ATTRIBUTE dickerKopf      1.0)))

(setq ct-adler    '(CT
                (ATTRIBUTE fluegel   1.0)
                (ATTRIBUTE augen     1.0)
                (ATTRIBUTE schnabel  1.0)
                (ATTRIBUTE ohren     1.0)
```

```
                    (ATTRIBUTE fuesse       1.0)
                    (ATTRIBUTE federn       0.9)
                    (ATTRIBUTE fliegen      0.8)

                    (ATTRIBUTE Faenge              1.0)
                    (ATTRIBUTE gross               1.0)
                    (ATTRIBUTE starkeFluegel       1.0)
                    (ATTRIBUTE Krallen             1.0)
                    (ATTRIBUTE breitFluegel        1.0)
                    (ATTRIBUTE befiederterKopf     1.0)))

(setq ct-adler2     '(CT
                    (ATTRIBUTE fluegel      1.0)
                    (ATTRIBUTE augen        1.0)
                    (ATTRIBUTE schnabel     1.0)
                    (ATTRIBUTE ohren        1.0)
                    (ATTRIBUTE fuesse       1.0)
                    (ATTRIBUTE federn       0.9)
                    (ATTRIBUTE fliegen      0.8)

                    (ATTRIBUTE Faenge              1.0)
                    (ATTRIBUTE gross               1.0)
                    (ATTRIBUTE starkeFluegel       1.0)
                    (ATTRIBUTE Krallen             1.0)
                    (ATTRIBUTE breitFluegel        1.0)
                    (ATTRIBUTE befiederterKopf     1.0)

                    (ATTRIBUTE a     1.0)
                    (ATTRIBUTE b     1.0)
                    (ATTRIBUTE c     1.0)
                    (ATTRIBUTE d     1.0)
                    (ATTRIBUTE e     1.0)))

(setq ct-adler3     '(CT
                    (ATTRIBUTE fluegel      1.0)
                    (ATTRIBUTE augen        1.0)
                    (ATTRIBUTE schnabel     1.0)
                    (ATTRIBUTE ohren        1.0)
                    (ATTRIBUTE fuesse       1.0)
                    (ATTRIBUTE federn       0.9)
                    (ATTRIBUTE fliegen      0.8)

                    (ATTRIBUTE Faenge              1.0)
                    (ATTRIBUTE gross               1.0)
                    (ATTRIBUTE starkeFluegel       1.0)
                    (ATTRIBUTE Krallen             1.0)
```

```
                    (ATTRIBUTE  breitFluegel        1.0)
                    (ATTRIBUTE  befiederterKopf  1.0)

                    (ATTRIBUTE a      1.0)
                    (ATTRIBUTE b      1.0)
                    (ATTRIBUTE c      1.0)
                    (ATTRIBUTE d      1.0)
                    (ATTRIBUTE e      1.0)

                    (ATTRIBUTE f      1.0)
                    (ATTRIBUTE g      1.0)
                    (ATTRIBUTE h      1.0)
                    (ATTRIBUTE i      1.0)
                    (ATTRIBUTE j      1.0)))

(setq ct-strauss   '(CT
                   (ATTRIBUTE  fluegel      1.0)
                    (ATTRIBUTE augen        1.0)
                    (ATTRIBUTE  schnabel    1.0)
                    (ATTRIBUTE  ohren       1.0)
                    (ATTRIBUTE fuesse       1.0)
                    (ATTRIBUTE federn       0.9)
                    (ATTRIBUTE fliegen      0.8)))

(setq  ct-pinguin '(CT
                    (ATTRIBUTE fluegel      1.0)
                    (ATTRIBUTE augen        1.0)
                    (ATTRIBUTE  schnabel    1.0)
                    (ATTRIBUTE ohren        1.0)
                    (ATTRIBUTE fuesse       1.0)
                    (ATTRIBUTE federn       0.9)
                    (ATTRIBUTE fliegen      0.8)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;             fish

(setq ct-fisch        '(CT
                    (ATTRIBUTE augen            1.0)
                    (ATTRIBUTE flossen          1.0)
                    (ATTRIBUTE kiemen           1.0)
                    (ATTRIBUTE  schuppen        0.9)
                    (ATTRIBUTE  schwimmen       1.0)))

(setq  ct-raubtier  '(CT
                    (ATTRIBUTE augen            1.0)
                    (ATTRIBUTE ohren            1.0)
```

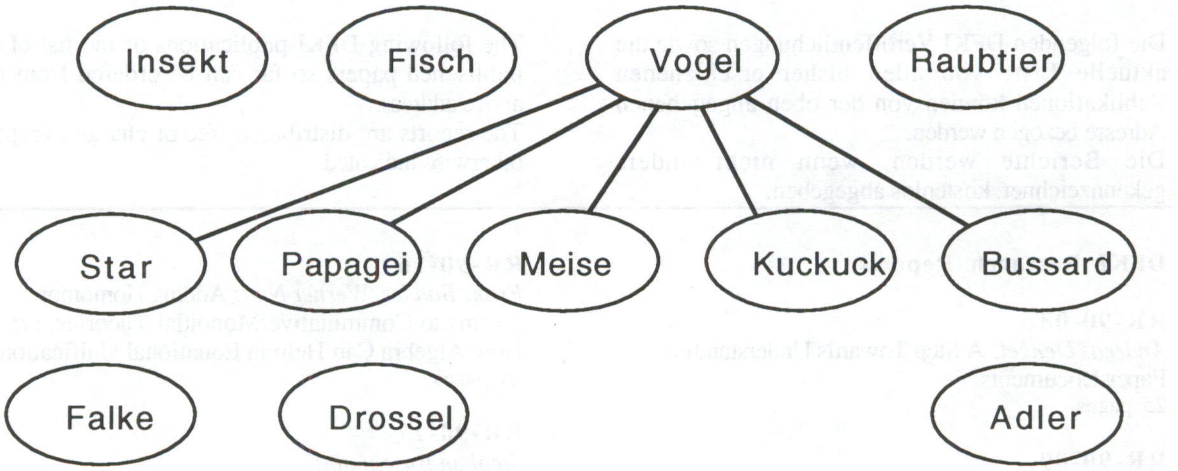
```
                  (ATTRIBUTE maul          1.0)
                  (ATTRIBUTE beine         0.9)
                  (ATTRIBUTE fell          0.8)
                  (ATTRIBUTE fuesse        0.9)
                  (ATTRIBUTE krallen       0.8)
                  (ATTRIBUTE laufen        0.8)))

(setq ct-insekt  '(CT
                  (ATTRIBUTE augen         1.0)
                  (ATTRIBUTE fuehler       1.0)
                  (ATTRIBUTE fluegel       0.8)
                  (ATTRIBUTE panzer        1.0)
                  (ATTRIBUTE fliegen       0.7)))
```

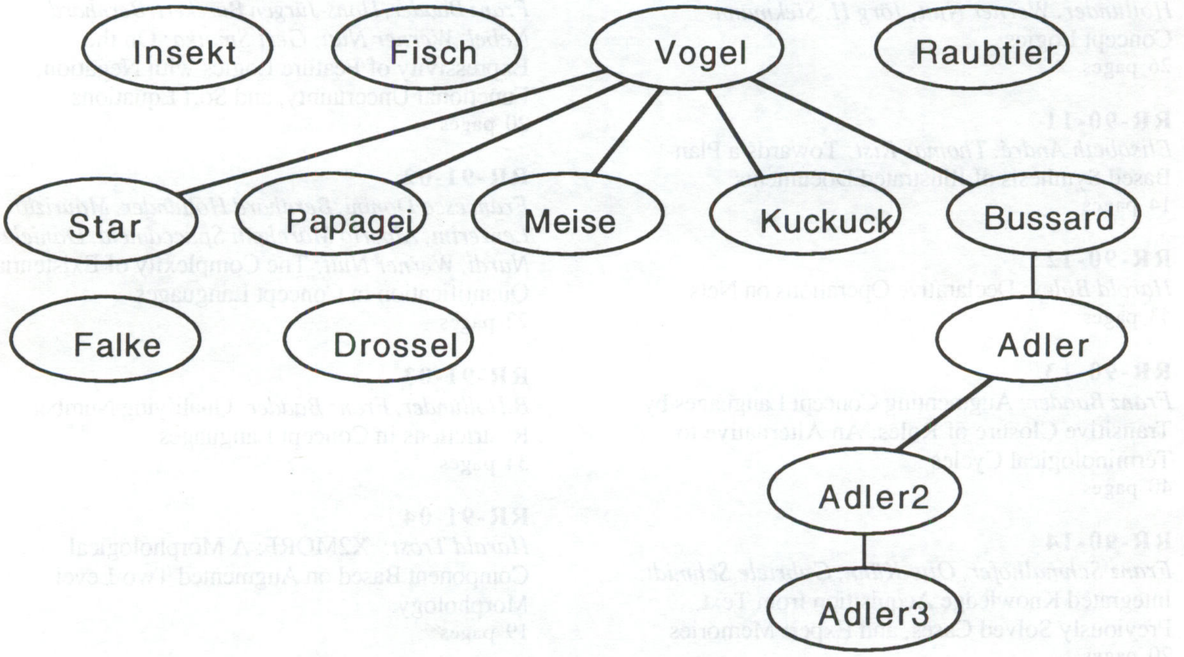
Appendix B

1. **Clustering:**

2. **Clustering:**



3. **Clustering:**



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
D-6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all publisched papers so far can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-90-08
Andreas Dengel: A Step Towards Understanding Paper Documents
25 pages

RR-90-09
Susanne Biundo: Plan Generation Using a Method of Deductive Program Synthesis
17 pages

RR-90-10
Franz Baader, Hans-Jürgen Bürckert, Bernhard Hollunder, Werner Nutt, Jörg H. Siekmann: Concept Logics
26 pages

RR-90-11
Elisabeth André, Thomas Rist: Towards a Plan-Based Synthesis of Illustrated Documents
14 pages

RR-90-12
Harold Boley: Declarative Operations on Nets
43 pages

RR-90-13
Franz Baader: Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles
40 pages

RR-90-14
Franz Schmalhofer, Otto Kühn, Gabriele Schmidt: Integrated Knowledge Acquisition from Text, Previously Solved Cases, and Expert Memories
20 pages

RR-90-15
Harald Trost: The Application of Two-level Morphology to Non-concatenative German Morphology
13 pages

RR-90-16
Franz Baader, Werner Nutt: Adding Homomorphisms to Commutative/Monoidal Theories, or: How Algebra Can Help in Equational Unification
25 pages

RR-90-17
Stephan Busemann:
Generalisierte Phasenstrukturgrammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung
114 Seiten

RR-91-01
Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, Gert Smolka: On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations
20 pages

RR-91-02
Francesco Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, Werner Nutt: The Complexity of Existential Quantification in Concept Languages
22 pages

RR-91-03
B.Hollunder, Franz Baader: Qualifying Number Restrictions in Concept Languages
34 pages

RR-91-04
Harald Trost: X2MORF: A Morphological Component Based on Augmented Two-Level Morphology
19 pages

RR-91-05
Wolfgang Wahlster, Elisabeth André, Winfried Graf, Thomas Rist: Designing Illustrated Texts: How Language Production is Influenced by Graphics Generation.
17 pages

RR-91-06
Elisabeth André, Thomas Rist: Synthesizing Illustrated Documents: A Plan-Based Approach
11 pages

RR-91-07
Günter Neumann, Wolfgang Finkler: A Head-Driven Approach to Incremental and Parallel Generation of Syntactic Structures
13 pages

RR-91-08
Wolfgang Wahlster, Elisabeth André, Som Bandyopadhyay, Winfried Graf, Thomas Rist: WIP: The Coordinated Generation of Multimodal Presentations from a Common Representation
23 pages

RR-91-09
Hans-Jürgen Bürckert, Jürgen Müller, Achim Schupeta: RATMAN and its Relation to Other Multi-Agent Testbeds
31 pages

RR-91-10
Franz Baader, Philipp Hanschke: A Scheme for Integrating Concrete Domains into Concept Languages
31 pages

RR-91-11
Bernhard Nebel: Belief Revision and Default Reasoning: Syntax-Based Approaches
37 pages

RR-91-12
J.Mark Gawron, John Nerbonne, Stanley Peters: The Absorption Principle and E-Type Anaphora
33 pages

RR-91-13
Gert Smolka: Residuation and Guarded Rules for Constraint Logic Programming
17 pages

RR-91-14
Peter Breuer, Jürgen Müller: A Two Level Representation for Spatial Relations, Part I
27 pages

RR-91-15
Bernhard Nebel, Gert Smolka: Attributive Description Formalisms ... and the Rest of the World
20 pages

RR-91-16
Stephan Busemann: Using Pattern-Action Rules for the Generation of GPSG Structures from Separate Semantic Representations
18 pages

RR-91-17
Andreas Dengel, Nelson M. Mattos: The Use of Abstraction Concepts for Representing and Structuring Documents
17 pages

RR-91-18
John Nerbonne, Klaus Netter, Abdel Kader Diagne, Ludwig Dickmann, Judith Klein: A Diagnostic Tool for German Syntax
20 pages

RR-91-19
Munindar P. Singh: On the Commitments and Precommitments of Limited Agents
15 pages

RR-91-20
Christoph Klauck, Ansgar Bernardi, Ralf Legleitner FEAT-Rep: Representing Features in CAD/CAM
48 pages

RR-91-21
Klaus Netter: Clause Union and Verb Raising Phenomena in German
38 pages

RR-91-22
Andreas Dengel: Self-Adapting Structuring and Representation of Space
27 pages

RR-91-23
Michael Richter, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner: Akquisition und Repräsentation von technischem Wissen für Planungsaufgaben im Bereich der Fertigungstechnik
24 Seiten

RR-91-24
Jochen Heinsohn: A Hybrid Approach for Modeling Uncertainty in Terminological Logics
22 pages

RR-91-25
Karin Harbusch, Wolfgang Finkler, Anne Schauder: Incremental Syntax Generation with Tree Adjoining Grammars
16 pages

RR-91-26
M. Bauer, S. Biundo, D. Dengler, M. Hecking, J. Koehler, G. Merziger: Integrated Plan Generation and Recognition - A Logic-Based Approach -
17 pages

RR-91-27
A. Bernardi, H. Boley, Ph. Hanschke, K. Hinkelmann, Ch. Klauck, O. Kühn, R. Legleitner, M. Meyer, M. M. Richter, F. Schmalhofer, G. Schmidt, W. Sommer:
ARC-TEC: Acquisition, Representation and Compilation of Technical Knowledge
18 pages

RR-91-28
Rolf Backofen, Harald Trost, Hans Uszkoreit:
Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends
11 pages

RR-91-29
Hans Uszkoreit: Strategies for Adding Control Information to Declarative Grammars
17 pages

RR-91-30
Dan Flickinger, John Nerbonne:
Inheritance and Complementation: A Case Study of *Easy* Adjectives and Related Nouns
39 pages

RR-91-31
H.-U. Krieger, J. Nerbonne:
Feature-Based Inheritance Networks for Computational Lexicons
11 pages

RR-91-32
Rolf Backofen, Lutz Euler, Günther Görz:
Towards the Integration of Functions, Relations and Types in an AI Programming Language
14 pages

RR-91-33
Franz Baader, Klaus Schulz:
Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures
33 pages

RR-91-34
Bernhard Nebel, Christer Bäckström:
On the Computational Complexity of Temporal Projection and some related Problems
35 pages

RR-91-35
Winfried Graf, Wolfgang Maaß: Constraint-basierte Verarbeitung graphischen Wissens
14 Seiten

DFKI Technical Memos

TM-91-01
Jana Köhler: Approaches to the Reuse of Plan Schemata in Planning Formalisms
52 pages

TM-91-02
Knut Hinkelmann: Bidirectional Reasoning of Horn Clause Programs: Transformation and Compilation
20 pages

TM-91-03
Otto Kühn, Marc Linster, Gabriele Schmidt:
Clamping, COKAM, KADS, and OMOS: The Construction and Operationalization of a KADS Conceptual Model
20 pages

TM-91-04
Harold Boley (Ed.):
A sampler of Relational/Functional Definitions
12 pages

TM-91-05
Jay C. Weber, Andreas Dengel, Rainer Bleisinger:
Theoretical Consideration of Goal Recognition Aspects for Understanding Information in Business Letters
10 pages

TM-91-06
Johannes Stein: Aspects of Cooperating Agents
22 pages

TM-91-08
Munindar P. Singh: Social and Psychological Commitments in Multiagent Systems
11 pages

TM-91-09
Munindar P. Singh: On the Semantics of Protocols Among Distributed Intelligent Agents
18 pages

TM-91-10
Béla Buschauer, Peter Poller, Anne Schauder, Karin Harbusch: Tree Adjoining Grammars mit Unifikation
149 pages

TM-91-11
Peter Wazinski: Generating Spatial Descriptions for Cross-modal References
21 pages

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann:
Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel:
ODA-based modeling for document analysis
14 pages

TM-91-15
Stefan Bussmann: Prototypical Concept Formation
An Alternative Approach to Knowledge Representation
28 pages

DFKI Documents

D-91-03
Harold Boley, Klaus Elsbernd, Hans-Günther Hein, Thomas Krause: RFM Manual: Compiling RELFUN into the Relational/Functional Machine
43 pages

D-91-04
DFKI Wissenschaftlich-Technischer Jahresbericht 1990
93 Seiten

D-91-06
Gerd Kamp: Entwurf, vergleichende Beschreibung und Integration eines Arbeitsplanerstellungssystems für Drehteile
130 Seiten

D-91-07
Ansgar Bernardi, Christoph Klauck, Ralf Legleitner
TEC-REP: Repräsentation von Geometrie- und Technologieinformationen
70 Seiten

D-91-08
Thomas Krause: Globale Datenflußanalyse und horizontale Compilation der relational-funktionalen Sprache RELFUN
137 Seiten

D-91-09
David Powers, Lary Reeker (Eds.):
Proceedings MLNLO'91 - Machine Learning of Natural Language and Ontology
211 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-10
Donald R. Steiner, Jürgen Müller (Eds.):
MAAMAW'91: Pre-Proceedings of the 3rd European Workshop on „Modeling Autonomous Agents and Multi-Agent Worlds"
246 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-91-11
Thilo C. Horstmann: Distributed Truth Maintenance
61 pages

D-91-12
Bernd Bachmann:
$\mathrm{H}^{\mathrm{iera}}\mathrm{C}_{\mathrm{on}}$ - a Knowledge Representation System with Typed Hierarchies and Constraints
75 pages

D-91-13
International Workshop on Terminological Logics
Organizers: Bernhard Nebel, Christof Peltason, Kai von Luck
131 pages

D-91-14
Erich Achilles, Bernhard Hollunder, Armin Laux, Jörg-Peter Mohren: *KRIS*: *K*nowledge *R*epresentation and *I*nference *S*ystem
- Benutzerhandbuch -
28 Seiten

D-91-15
Harold Boley, Philipp Hanschke, Martin Harm, Knut Hinkelmann, Thomas Labisch, Manfred Meyer, Jörg Müller, Thomas Oltzen, Michael Sintek, Werner Stein, Frank Steinle:
μCAD2NC: A Declarative Lathe-Worplanning Model Transforming CAD-like Geometries into Abstract NC Programs
100 pages

D-91-16
Jörg Thoben, Franz Schmalhofer, Thomas Reinartz: Wiederholungs-, Varianten- und Neuplanung bei der Fertigung rotationssymmetrischer Drehteile
134 Seiten

D-91-17
Andreas Becker:
Analyse der Planungsverfahren der KI im Hinblick auf ihre Eignung für die Abeitsplanung
86 Seiten

D-91-18
Thomas Reinartz: Definition von Problemklassen im Maschinenbau als eine Begriffsbildungsaufgabe
107 Seiten

D-91-19
Peter Wazinski: Objektlokalisation in graphischen Darstellungen
110 Seiten

Prototypical Concept Formation

An Alternative Approach to Knowledge Representation

Stefan Bussmann

TM-91-15
Technical Memo