The Relevance of Artificial Intelligence for Human Cognition

Helmar Gust and **Kai–Uwe Kühnberger**
Institute of Cognitive Science
Albrechtstr. 28
University of Osnabrück, Germany
{hgust, kkuehnbe}@uos.de

Abstract

We will discuss the question whether artificial intelligence can contribute to a better understanding of human cognition. We will introduce two examples in which AI models provide explanations for certain cognitive abilities: The first example examines aspects of analogical reasoning and the second example discusses a possible solution for learning first-order logical theories by neural networks. We will argue that artificial intelligence can in fact contribute to a better understanding of human cognition.

Introduction

Quite often artificial intelligence is considered as an engineering discipline, focusing on solutions for problems in complex technical systems. For example, building a robot navigation device in order to enable the robot to act in an unknown environment poses problems like the following ones:

- How is it possible to detect obstacles by the sensory devices of the robot?
- How is it possible to circumvent time-critical planning problems of a planning system that is based on a time consuming deduction calculus?
- Which problem solving abilities are available to deal with occurring unforeseen problems?
- How is it possible to identify early enough dangerous objects, surfaces, enemies etc., in particular, if they are never seen before by the robot?

Although such problems do have certain similarities to classical questions in cognitive science it is usually not assumed that solutions for the robot can be analogously transferred to solutions for cognitive science. For example, a solution for a planning problem of a mobile robot does not necessarily have any consequences for strategies to solve planning problems in cognitive agents like humans. Quite often it is therefore claimed that engineering solutions for technical devices are not cognitively adequate.

On the other hand, it is frequently assumed that cognitive science and, in particular, the study of human cognition try to develop attempts for solutions of problems that are usually considered as hard for artificial intelligence. Examples are human abilities like adaptivity, creativity, productivity, motor coordination, perception, emotions, or goal generation of autonomous agents. It seems to be the case that these aspects of human cognition do not have simple solutions that can be straightforwardly implemented in a machine. Therefore, human cognition is often considered as a reservoir for new challenges in artificial intelligence.

In this paper, we will discuss the question: *Can artificial intelligence contribute to our understanding of human cognition?* We will argue for the existence of such a contribution (contrary to the discussion above). Our arguments are based on own results in two domains of AI research: first, analogical reasoning and second, the learning of logical first-order theories by neural networks. We claim that appropriate solutions in artificial intelligence can provide explanations in cognitive science by using well-established formal methods, the rigorous specification of the problem, and the practical realization in a computer program. More precisely by modeling analogical reasoning with formal tools we will be able to get an idea how creativity and productivity of human cognition as well as efficient learning without large data sets is possible. Furthermore, learning first-order theories by neural networks can be used to explain why human cognition is often model-based and less time-consuming than a formal deduction. Therefore, artificial intelligence can contribute to a better understanding of human cognition.

The paper will have the following structure: First, we will discuss an account for modeling analogical reasoning and we will sketch the consequences for cognitive science. Second, we will roughly discuss the impact of a solution for learning logical inferences by neural networks. We will give an explanation for some empirical findings. Finally we will summarize the discussion.

Example 1: Analogical Reasoning

The Analogy between Water and Heat

It is quite undisputed that analogical reasoning is an important aspect of human cognition. Although there has been a strong endeavor during the last twenty-five years to develop a theory of analogies and, in particular, a theory of analogical learning, no generally accepted solution has been pro-

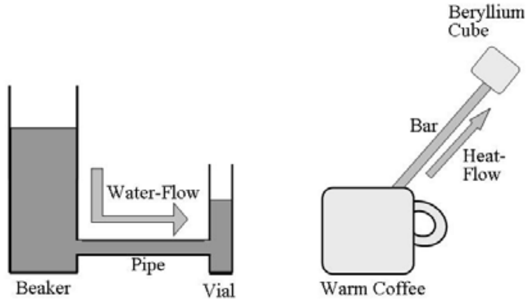

Figure 1: The diagrammatic representation of the heat-flow analogy.

posed yet. Connected with the problem of analogical reasoning is the problem of interpreting metaphorical expressions (Gentner et al. 2001). Similarly to analogies, there is no convincing automatic procedure that computes the meaning of metaphorical expressions in a broad variety of domains either. Recently published, the monograph (Gentner, Holyoak, and Kokinov 2001) can be seen as a summary of important approaches towards a modeling of analogies.

Figure 1 represents the analogy between a water-flow system, where water is flowing from the beaker to the vial, and a heat-flow system, where heat is flowing from the warm coffee to a beryllium cube. The analogy consists of the association of water-flow on the source side and heat-flow on the target side. Although this seems to be a rather simple analogy, a non-trivial property of this association must be modeled: The concept *heat* is a theoretical term and not anything that can be measured directly by a physicist. Therefore the establishment of an analogical relation between the water-flow system and the heat-flow system must productively generate an additional concept *heat* flowing from warm coffee to a cold beryllium cube. This leads to an analogy where the measureable heights of the water levels in the beaker and the vial correspond to the temperature of the warm coffee and the temperature of the beryllium cube, respectively.

HDTP – A Theory Computing Analogies

Heuristic-Driven Theory Projection (HDTP) is a formally sound theory for computing analogical relations between a source domain and a target domain. HDTP computes analogical relations not only by associating concepts, relations, and objects, but also complex rules and facts between the target and the source domain. In (Gust, Kühnberger, and Schmid 2005a) the syntactic, semantic, and algorithmic properties of HDTP are specified. Unlike to well-known accounts for modeling analogies like the structure-mapping engine (Falkenhainer, Forbus, and Gentner 1989) or Copycat (Hofstadter 1995), HDTP produces abstract descriptions of the underlying domains, is heuristic-driven, i.e. allows to include various types of background knowledge, and has a model theoretic semantics induced by an algorithm.

Syntactically, HDTP is defined on the basis of a many-sorted first-order language. First-order logic is used in order to guarantee the necessary expressive power of the account. An important assumption is that analogical reasoning crucially contains a generalization (or abstraction) process. In other words, the identification of common properties or relations is represented by a generalization of the input of source and target. Formally this can be modeled by an extension of the so-called theory of anti-unification (Plotkin 1970), a mathematically sound account describing the possibility of generalizing terms of a given language using substitutions. More precisely, an anti-unification of two terms t_1 and t_2 can be interpreted as finding a generalized term t (or structural description t) of t_1 and t_2 which may contain variables, together with two substitutions Θ_1 and Θ_2 of variables, such that $t\Theta_1 = t_1$ and $t\Theta_2 = t_2$. Because there are usually many possible generalizations, anti-unification tries to find the most specific one. An example should make this idea clear. Assume two terms $t_1 = f(x, b, c)$ and $t_2 = f(a, y, c)$ are given. Generalizations are, for example, the terms $t = f(x, y, c)$ and $t' = f(x, y, z)$ together with their corresponding substitutions.[1] But t is more specific than t', because the substitution Θ substituting z by c can be applied to t'. This application results in: $t'\Theta = t$. Most specific generalizations of two terms are commonly called anti-instances.

Table 1: A simplified description of the algorithm HDTP-A omitting formal details. A precise specification of this algorithm can be found in (Gust, Kühnberger, and Schmid 2005a).

Input: A theory Th_S of the source domain and a theory Th_T of the target domain represented in a many-sorted predicate logic language.
Output: A generalized theory Th_G such that the input theories Th_S and Th_T can be reestablished by substitutions.

Selection and generalization of fact and rules.
    Select an axiom from the target domain (according to a heuristics h).
    Select an axiom from the source domain and construct a generalization (together with corresponding substitutions).
Optimize the generalization w.r.t. a given heuristics h'.
    Update the generalized theory w.r.t. the result of this process.
Transfer (project) facts of the source domain to the target domain provided they are not generalized yet.
    Test (using an oracle) whether the transfer is consistent with the target domain.

Given two input theories Th_S and Th_T for source and target domains respectively, the algorithm HDTP-A computes anti-instances together with a generalized theory Th_G. Table 1 makes the algorithm more precise: First, an axiom from the target domain is selected guided by an appropriate heuristics h, for example, measuring the syntactic complexity of the axiom. Then an axiom of the source domain is searched in order to construct a generalization together with

[1] As usual we assume that $a, b, c, \ldots$ denote constants and $x, y, z, \ldots$ denote variables.

Table 2: Examples of corresponding concepts in the source and the target domains of the heat-flow analogy.

Source	**Target**	**A**
(1) connected(beaker,vial,pipe)	connected(coffe in cup,b_cube,bar)	connected (A,B,C)
(2) liquid(water)	liquid(coffee)	liquid(D)
(3) height(water in beaker,t1) $>$ height(water in vial,t2)	temp(coffe in cup,t1) $>$ temp(b_cube,t1)	T(A,t1) $>$ T(B,t1)
(4) height(water in beaker,t1) $>$ height(water in beaker,t2)	temp(coffee in cup,t1) $>$ temp(coffe in cup,t2)	T(A,t1) $>$ T(A,t2)
(5) height(water in vial,t2) $>$ height(water in vial,t1)	temp(b_cube,t2) $>$ temp(b_cube,t1)	T(B,t2) $>$ T(B,t1)

substitutions. The generalization is optimized using another heuristics h', for example, the length of the necessary substitutions. Finally axioms from the source domain are projected to the target domain. Then the transferred axioms are tested for consistency with the target domain using an oracle.

Applying this theory to our example depicted in Figure 1 yields the intuitively correct result. Table 2 depicts some of the crucial associations that are important for establishing the analogy. We summarize the corresponding substitutions Θ_1 and Θ_2 in the following list:

$A \longrightarrow$ beaker / coffee in cup
$B \longrightarrow$ vial / b_cube
$C \longrightarrow$ pipe / bar
$D \longrightarrow$ water / coffee
$T \longrightarrow \lambda x, t$: height(water in x,t) / temperature

The example – although seemingly simple – has a relatively complicated aspect: The system associates an abstract property $\lambda x, t$: height(water in x,t) with *temperature*. The concept *heat* must be introduced as a counterpart of water in the target domain by projecting the structure of the λ-term above to the target domain by the following equation: temperature(x,t) = height(heat in x,t)

HDTP was applied to a variety of domains, for example, naive physics (Schmid et al. 2003) and metaphors (Gust, Kühnberger, and Schmid 2005b). The algorithm HDTP-A is implemented in SWI-Prolog. The core program is available online (Gust, Kühnberger, and Schmid 2003).

Explanations for Cognitive Science

We would like to argue for the claim that the sketched productive solutions of analogical reasoning problems can have an impact to the understanding of human cognition. The first argument is that HDTP is a theory and specifies analogical reasoning on a syntactic, a semantic, and an algorithmic level. This is quite often different in frameworks developed from a cognitive science perspective. Usually those accounts give precise descriptions of psychological experiments, often they try to find psychological generalizations, but regularly they lack a formally specified explanation why some empirical data can be measured. The advantage of an AI solution to analogies is that a fine-grained analysis of analogy making can be achieved due to the formally specified logical basis and the algorithmic specification. This enables us to specify precisely which assumptions must be made in order to be able to establish analogical relations.

Analogical reasoning shows an important feature that distinguishes this type of inferences from other types of reasoning, like inductive learning, case-based reasoning, or exemplar-based learning: All these latter forms of learning are based on a rather large number of training instances which are usually barely structured. Learning is possible, because many instances are available. Therefore generalizations of existing data can primarily be computed due to a large number of examples, whereas given domain theories play usually a less important role. In contrast to these types of learning, analogical learning is based on a rather small number of examples: In many cases only one (rich) conceptualization of the source domain and a rather coarse conceptualization of the target domain is available. But on the other hand analogies are based on sufficient background knowledge. A cognitive science explanation for analogical inferences must take this into account. It is not sufficient to apply standard learning algorithms to explain analogical learning, but accounts need to be used that explain precisely why the background knowledge is sufficient in one application but insufficient in another. Furthermore, accounts are needed that can explain whether a particular analogical relation can be established without taking into account a spelled-out theory, or whether such a theory is in fact necessary. Precisely this can be achieved by applying HDTP.

Because the discovery of a sound analogical relation provides *immediately* a new conceptualization of the target domain, this may be a hint for the explanation of sudden insights. Notice that such insights could have a certain connection to the Gestalt laws. Such Gestalt laws can be interpreted as the concurrency of different analogical relations. Therefore analogical reasoning can be extended to further higher cognitive abilities.

We summarize why the modeling of analogies using HDTP contributes to the understanding of human cognition:

- HDTP is a theory, not a description of empirical data, explaining productive capabilities of human cognition.
- HDTP provides a fine-grained analysis of analogical transfers on a syntactic, semantic, and algorithmic level.
- HDTP provides an explanation why analogical learning is possible without a large number of examples.
- An extension to other cognitive abilities seems to be promising.

Example 2: Symbols and Neural Networks

The Problem

The gap between symbolic and subsymbolic models of human cognition is usually considered as a hard problem. On the symbolic level recursion principles ensure that the formalisms are productive and allow a very compact representation: Due to the compositionality principle it is possible to compute the meaning of a complex (logical) expression using the meaning of the embedded subexpressions. On the other hand, it is assumed that neural networks are non-compositional on a principal basis making it difficult to represent complex data structures like lists, trees, tables, formulas etc. Two aspects can be distinguished: The representation problem (Barnden 1989) and the inference problem (Shastri and Ajjanagadde 1990). The first problem states that, if at all, complex data structures can only be used implicitly and the representation of structured objects is a non-trivial challenge for neural networks. The second problem tries to model inferences of logical systems with neural accounts.

A certain endeavor has been invested to solve the representation problem as well as the inference problem. It is well-known that classical logical connectives like conjunction, disjunction, or negation can be represented by neural networks. Furthermore it is known that every Boolean function can be learned by a neural network (Steinbach and Kohut 2002). Although it is therefore possible to represent propositional logic with neural networks, this is not true for first-order logic (FOL). The corresponding problem, usually called the variable-binding problem, is caused by the usage of quantifiers $\forall$ and $\exists$, which may bind variables that occur at different positions in one and the same formula. There are a number of attempts to solve the problem of representing logical formulas with neural networks: Examples are sign propagation (Lange and Dyer 1989), dynamic localist representations (Barnden 1989), or tensor product representations (Smolensky 1990). Unfortunately these accounts have certain non-trivial side-effects. Whereas sign propagation and dynamic localist representations lack the ability of learning, tensor product representations result in an exponentially increasing number of elements to represent variable bindings, just to mention some of the problems.

With respect to the inference problem of connectionist networks the number of proposed solutions is rather small and relatively new. An attempt is (Hitzler, Hölldobler, and Seda 2004) in which a logical deduction operator is approximated by a neural network. In (D'Avila Garcez, Broda, and Gabbay 2002), tractable fragments of predicate logic are learned by connectionist networks.

Closing the Gap between Symbolic and Subsymbolic Representations

In (Gust and Kühnberger 2004) and (Gust and Kühnberger 2005) a framework was developed that enables neural networks to learn logical first-order theories. The idea is rather simple: Because interpretation functions of FOL cannot be learned directly by neural networks (due to their heterogeneous structure and the variable-binding problem) logical formulas are translated into a homogeneous variable-free representation. The underlying structure for this representation is a topos (Goldblatt 1984), a category theoretic structure that can be interpreted as a model of FOL (Gust 2000). In a topos, logical expressions correspond simply to constructions of arrows given other arrows. Therefore every construction can be reduced to one single operation, namely

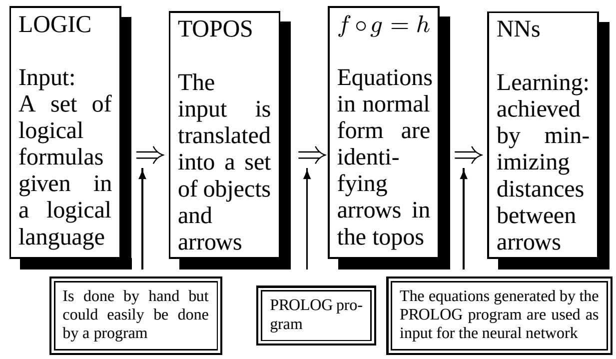

Figure 2: The general architecture of an account transferring logical theories into a variable-free representation and feeding a neural network with equations of the form $f \circ h = g$.

the concatenation of arrows, i.e. the concatenation of set-theoretic functions (in the easiest case that the topos is isomorphic to the category $\mathbf{SET}$ of sets and set theoretic functions). In a topos, not every arrow corresponds directly to a symbol (or a complex string of symbols). Similarly there are symbols that have no direct representation in a topos: For example, variables do not occur in a topos but are hidden or indirectly represented. Another example of symbols that have no simple representation in a topos are quantifiers.

Figure 2 depicts the general architecture of the system. Given a representation of a first-order logical formula in a topos, a Prolog program generates equations $f \circ g = h$ of arrows in normal form that can be fed to a neural network. The equations are determined by constructions that exist in a topos. Examples are products, coproducts, or pullbacks.[2] The network is trained using these equations and a simple backpropagation algorithm. Due to the fact that a topos codes implicitly symbolic logic, we call the representation of logic in a topos the *semisymbolic* level. In a topos, an arrow connects a domain with a codomain. In the neural representations, all these entities (domains, codomains, and arrows) are represented as points in a n-dimensional vector space.

The structure of the network is depicted in Figure 3. In order to enable the system to learn logical inferences, some basic arrows have static (fixed) representations. These representations correspond directly to truth values.

- The truth value *true* : $(1.0, 1.0, 1.0, 1.0, 1.0)$
- The truth value *false* : $(0.0, 0.0, 0.0, 0.0, 0.0)$

Notice that the truth value *true* and the truth value *false* are maximally distinct. First results of learning FOL by this approach are promising (Gust and Kühnberger 2005). Both, the concatenation operation and the representations of the arrows together with their domains and codomains are learned by the network. Furthermore the network does not only learn a certain input theory but rather a model of the input theory,

[2] Simply examples in set theory for product constructions are Cartesian products. Coproducts correspond to disjoint unions of sets. Pullbacks are generalized products.

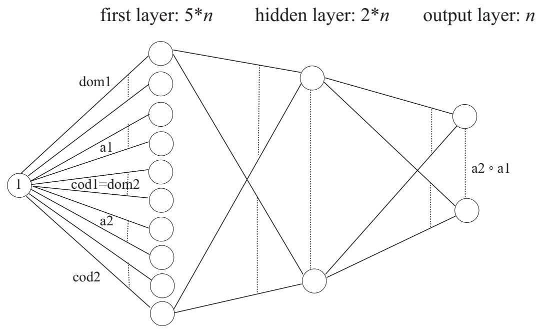

Figure 3: The structure of the neural network that learns composition of first-order formulas.

i.e. the input together with the closure of the theory under a deduction calculus.

The translation of first-order formulas into training data of a neural network allows, in principal, to represent models of symbolic theories in artificial intelligence and cognitive science (that are based on FOL) with neural networks.[3] In other words the account provides a recipe – and not just a general statement of the possibility – of how to learn models of theories based on FOL with neural networks. The sketched account tries to combine the advantages of connectionist networks and logical systems: Instead of representing symbols like constants or predicates using single neurons, the representation is rather distributed, realizing the very idea of distributed computations in neural networks. Furthermore the neural network can be trained quite efficiently to learn a model without any hardcoded devices. The result is a distributed representation of a symbolic system.[4]

Explaining Inferences as the Learning of Models

A logical theory consists of axioms specifying facts and rules about a certain domain together with a calculus determining the "correct" inferences that can be drawn from these axioms. From a computational point of view this quite often generates problems, because inferences can be rather resource consuming. Modeling logical inferences with neural networks as sketched in the subsection above allows a very efficient way of drawing inferences, simply because the interpretation of possible queries is "just there", namely implicitly coded by the distribution of the weights of the network. The account explains why time-critical deductions can be performed by humans using models instead of calculi. It is important to emphasize that the neural network does not only learn the input, but a whole *model* making the input true. In a certain sense these models are overdetermined, i.e. they assign truth values to every query, even though the theory does not determine a truth value. Nevertheless the models are consistent with the theory. This distinguishes the trained neural network from a symbolic theorem prover. Whereas the theorem prover just deduces the theorems of the theory consistent with the underlying logic, the neural network assigns values to every query.

There is empirical evidence from the famous Wason selection-task (and the various versions of this task) that human behavior is (in our terminology) rather model-based than theory-based, i.e. human behavior can be deductive without having an inference mechanism (Johnson-Laird 1983). In other words, humans do not perform deductions if they reason logically, but rather apply a model of the corresponding situation. We can give an explanation of this phenomenon using the presented neural network account: Humans act mostly according to a model they learned (about, for example, a situation, a scene, or a state of affairs) and not according to a theory plus an inference mechanism.

There is a certain tendency of our learned models towards a closed-world assumption. Consider the following rules:

All humans are mortal.
All mortal beings ascend to heaven.
All beings in heaven are angels.

If we know that Socrates is human we would like to deduce that Socrates is an angel. But if we just know that the robot is not mortal, we would rather like to deduce that the robot is not an angel. The models learned by the neural network provide hints for an explanation of these empirical observations: The property of the robot to be non-human propagates to the property of the robot to be non-angel. This provides evidence for an equivalence between *The robot is human* and *The robot is an angel* in certain types of underdetermined situations.

A difficult problem for cognitive science and symbol-based robotics are modelings of time constraints. On the one hand, it is possible for humans to be quite successful in a hostile environment in which time-critical situations occur and rapid responses and actions involving some kind of planning are necessary. On the other hand, symbol-based machines often have significant problems in solving such tasks. A natural explanation is that humans do not deduce anything, but rather apply an appropriate model in certain circumstances. Again this type of explanation can be modeled by the sketched connectionist approach. All knowledge about a state of affairs is just there, namely implicitly coded in the weights of the network. Clearly, the corresponding model can be wrong or imprecise, but a reaction in time-critical situations is always possible.

Although the gap between symbolic and subsymbolic approaches in cognitive science and AI is obvious, there is still no generally accepted solution for this problem. In particular, in order to understand human cognition the question is often raised of how an explanation for the emergence of conceptual knowledge from subsymbolic sensory data and the emergence of subsymbolic motor behavior from conceptual knowledge are possible at all. To put the task into the symbol-neural distinction (without discussing the differences between the two formulations): how can rules be retrieved from trained neural networks and how can symbolic

[3] Notice that a large part of theories in artificial intelligence are formulated with tools taken from logic and are mostly based on FOL or subsystems of FOL.

[4] In a certain sense the presented account is an extreme case of a distributed representation, opposing the other extreme case of a purely symbolic representation. Human cognition is probably neither of the two extreme cases.

knowledge (including complex data structures) be learned by neural networks. Clearly we do not claim to solve this problem, but at least our approach shows how one direction – namely the learning of logical first-order theories by neural networks – can uniformly be solved. In this approach two major principles are realized: first, the network can learn, and second, the topology of the network does not need to be changed in order to learn new input. We do not know any other approach that realizes these two principles.

Again we summarize the arguments why an AI solution for logical inferences using neural networks can contribute to the understanding of human cognition:

- The presented account explains why logical inferences are often based on models or situations not on logical deductions.
- It is possible to explain, why complex inferences can be realized by humans but are rather time-consuming to realize for deduction calculi.
- Last but not least, we can give hints how neural networks – usually considered as inappropriate for the deduction of logical facts – can be used to perform logical inferences.

Conclusions

In this paper, we discussed two AI models that provide solutions for certain aspects of higher cognitive abilities. These models were used to argue for the claim that artificial intelligence can contribute to a better understanding of human cognition. In particular, we argued that the computation of analogies using HDTP can explain the creativity of analogical inferences in a mathematically sound framework without reference to a large number of examples. Furthermore we argued that the modeling of logical theories using neural networks can explain why humans usually apply models of situations, but do not perform deductions in order to make logical inferences. This observation can be used to explain, why humans are quite successful in time-critical circumstances, whereas machines using sophisticated deduction algorithms must fail. We believe that these ideas can be extended to other applications like planning problems (in the case of representing symbolic theories with neural networks) or aspects of perception (in the case of analogical reasoning). Last but not least, it seems to be possible to combine both accounts – for example, by modeling analogical learning through neural networks – in order to achieve a unified theory of cognition, but this remains a task for future research.

References

Barnden, J.A. 1989. Neural net implementation of complex symbol processing in a mental model approach to syllogistic reasoning. In Proceedings of the International Joint Conference on Artificial Intelligence, 568-573.

D'Avila Garcez, A., Broda, K., and Gabbay, D. 2002. *Neural-Symbolic Learning Systems: Foundations and Applications*. Berlin Heidelberg, Springer.

Falkenhainer, B., Forbus, K., and Gentner, D. 1989. The structure-mapping engine: Algorithm and example, *Artificial Intelligence*, 41:1-63.

Gentner, D., Bowdle, B., Wolff, P., and Boronat, C. 2001. Metaphor is like analogy. In Gentner, D., Holyoak, K. & Kokinov, B. (ed.): *The analogical mind: Perspectives from cognitive science*, 199-253. Cambridge, MA: MIT Press.

Gentner, D., Holyoak, K., and Kokinov, B. 2001. *The Analogical Mind. Perspectives from Cognitive Science*, Cambridge, MA: MIT Press.

Goldblatt, R. 1984. *Topoi, the categorial analysis of logic*. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam.

Gust, H. 2000. Quantificational Operators and their Interpretation as Higher Order Operators. M. Böttner & W: Thümmel, *Variable-free Semantics*, 132-161, Osnabrück.

Gust, H. and Kühnberger, K.-U. 2004. Cloning Composition and Logical Inference in Neural Networks Using Variable-Free Logic. AAAI Fall Symposium Series 2004, Symposium: Compositional Connectionism in Cognitive Science, Washington D.C., 25-30.

Gust, H. and Kühnberger, K.-U. 2005. Learning Symbolic Inferences with Neural Networks. In: B. Bara, L. Barsalou & M. Bucciarelli (Hrsg.): CogSci 2005: XXVII Annual Conference of the Cognitive Science Society, 875-880. Lawrence Erlbaum.

Gust, H., Kühnberger, K.-U., and Schmid, U. 2003. Anti-unification of axiomatic systems. Available on the www: http://www.cogsci.uni-osnabrueck.de/ helmar/analogy/.

Gust, H., Kühnberger, K.-U., and Schmid, U. 2005a. Metaphors and Heuristic-Driven Theory Projection. Forthcoming in *Theoretical Computer Science*.

Gust, H., Kühnberger, K.-U., and Schmid, U. 2005b. Ontologies as a Cue for the Metaphorical Meaning of Technical Concepts, forthcoming in A. Schalley & D. Khlentzos (eds.): *Mental States: Evolution, Function, Nature*, John Benjamins Publishing Company.

Hitzler, P., Hölldobler, S., and Seda, A. 2004. Logic programs and connectionist networks. *Journal of Applied Logic*, 2(3):245-272.

Hofstadter, D. and The Fluid Analogies Research Group, *Fluid concepts and creative analogies*, New York, 1995.

Johnson-Laird, P. 1983. *Mental Modes: Towards a Cognitive Science of Language, Inference, and Consciousness*, Cambridge, Mass.

Lange, T., and Dyer, M.G. 1989. High-level inferencing in a connectionist network. Technical report UCLA-AI-89-12.

Plotkin, G. 1970. A note on inductive generalization. *Machine Intelligence* 5:153-163.

Schmid, U., Gust, H., Kühnberger, K.-U., and Burghardt, J. 2003. An Algebraic Framework for Solving Proportional and Predictive Analogies, in Schmalhofer, F., Young, R., and Katz, G. (eds.): Proceedings of EuroCogSci 03. The European Cognitive Science Conference 2003, 295-300. Lawrence Erlbaum Associates.

Shastri, L. and Ajjanagadde, V. 1990. From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences* 16:417-494.

Smolenski, P. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence* 46(1-2):159-216.

Steinbach, B. and Kohut, R. 2002. Neural Networks – A Model of Boolean Functions. In Steinbach, B. (ed.): Boolean Problems, Proceedings of the 5th International Workshop on Boolean Problems, 223-240. Freiberg.