When Almost Is Not Even Close: Remarks on the Approximability of HDTP

Tarek R. Besold[1] and Robert Robere[2]

[1] Institute of Cognitive Science, University of Osnabrück, tbesold@uos.de
[2] Department of Computer Science, University of Toronto, robere@cs.toronto.edu

Abstract. A growing number of researchers in Cognitive Science advocate the thesis that human cognitive capacities are constrained by computational tractability. If right, this thesis also can be expected to have far-reaching consequences for work in Artificial General Intelligence: Models and systems considered as basis for the development of general cognitive architectures with human-like performance would also have to comply with tractability constraints, making in-depth complexity theoretic analysis a necessary and important part of the standard research and development cycle already from a rather early stage. In this paper we present an application case study for such an analysis based on results from a parametrized complexity and approximation theoretic analysis of the Heuristic Driven Theory Projection (HDTP) analogy-making framework.

Introduction: Cognition, Tractability and AGI

The declared goal of what is known as Artificial General Intelligence (AGI) is the (re)creation of intelligence in an artificial system on a level at least comparable to humans. As detailed in [1], an AI research project, in order to qualify for being an AGI project, amongst others has to "*(...) be based on a theory about 'intelligence' as a whole (which may encompass intelligence as displayed by the human brain/mind, or may specifically refer to a class of non-human-like systems intended to display intelligence with a generality of scope at least roughly equalling that of the human brain/mind)*". Thus, even in the second case, humans and their performance and capabilities (although not without any alternative) stay the main standard of comparison for the targeted type of system — which in turn makes it seem highly likely that understanding and subsequently modeling and implementing human-style cognitive capacities could play a crucial role in achieving the goals of AGI.

A common way of proceeding in designing models and computational implementations of cognitive faculties is based on taking a computational-level theory of the cognitive process (as, e.g., often directly originating from research in cognitive psychology) and to construct an algorithmic-level implementation simulating the respective cognitive faculty. Different researchers over the last decades have proposed the use of mathematical complexity theory, and namely the concept of NP-completeness, as an assisting tool in specifying the necessary properties and limiting constraints relevant when deciding for a particular computational-level theory of a cognitive faculty, bringing forth the so called "*P-Cognition thesis*": Human cognitive capacities are hypothesized to be of the polynomial-time computable type. Following the recognition that using polynomial-time computability as criterion might be overly restrictive, [2] with the "FPT-*Cognition thesis*" recently introduced a relaxed version of the original thesis, demanding for human cognitive capacities to be fixed-parameter tractable for one or more input parameters that are small in practice (i.e., stating that the computational-level theories have to be in FPT).[3]

Taking inspiration in the latter formulation of the idea that complexity considerations can provide guidance in selecting suitable models and computational-level theories of cognitive capacities, in [5] we for the first time presented a basic version of a similar thesis suitable for the use in attempts to (re)create cognitive capacities in an artificial system:

Tractable AGI thesis
Models of cognitive capacities in artificial intelligence and computational cognitive systems have to be fixed-parameter tractable for one or more input parameters that are small in practice (i.e., have to be in FPT).

From a purely theoretical point of view, this requirement seems rather intuitive and reasonable, though its practical use and implications might not be initially obvious. In this paper we therefore want to have a closer look at a worked example, illustrating the possible usefulness of the Tractable AGI thesis by providing an in-depth analysis of the computational complexity of Heuristic-Driven Theory Projection (HDTP), an established framework for computational analogy-making. The choice for using HDTP in the analysis was not arbitrary: In our eyes the system suits the purpose as it addresses a core cognitive capacity, its use of first-order logic as representation formalism makes it sufficiently expressive as to be considered a general domain system, and the overall approach and applied techniques reflect acknowledged standards in the field.

(In)Tractability and Heuristic-Driven Theory Projection

During the course of a day, we use different kinds of reasoning processes: We solve puzzles, play instruments, or discuss problems. Often we will find ourselves in situations in which we apply our knowledge of a familiar situation to a structurally similar novel one. Today it is undoubted that one of the basic elements of human cognition is the ability to see two a priori distinct domains as similar based on their shared relational structure (i.e., analogy-making). Key abilities within everyday life, such as communication, social interaction, tool use and the handling of previously unseen situations crucially rely on the use of analogy-based strategies and procedures. Relational matching, one of the key mechanisms underlying analogy-making, is also the basis of perception, language, learning, memory and thinking, i.e., the constituent elements of most conceptions of cognition [6] — some prominent cognitive scientists even consider analogy the core of cognition itself [7].

Because of this crucial role of analogy in human cognition researchers in cognitive science and artificial intelligence have been creating computational models of

[3] A problem P is in FPT if P admits an $O(f(\kappa)n^c)$ algorithm, where n is the input size, κ is a parameter of the input constrained to be "small", c is an independent constant, and f is some computable function. (For an introduction to parametrized complexity theory see, e.g., [3, 4].)

analogy-making since the advent of computer systems. This line of work has resulted in several different frameworks for computational analogical reasoning, featuring systems as prominent as Hofstadter's Copycat [8] or the famous Structure-Mapping Engine (SME) [9] and MAC/FAC [10]. Whilst the latter two systems implement a version of Gentner's Structure-Mapping Theory (SMT) [11], more recently a different, generalization-based approach has been proposed: Heuristic-Driven Theory Projection (HDTP) [12, 13].

In what follows, we want to give a detailed complexity analysis of the mechanisms underlying the analogy process in HDTP as worked example for how the study of complexity properties can provide important contributions to understanding and designing an artificial cognitive system. For doing so, we will first introduce the theoretical basis of HDTP, before continuing with a presentation of results addressing parametrized complexity and approximation theoretic properties of the model, and in conclusion discussing the obtained insights in the broader context of AGI and cognitive modeling.

Heuristic-Driven Theory Projection for Computational Analogy-Making

The Heuristic-Driven Theory Projection framework [12] has been conceived as a mathematically sound framework for analogy-making. HDTP has been created for computing analogical relations and inferences for domains which are given in form of a many-sorted first-order logic representation [14]. Source and target of the analogy-making process are defined in terms of axiomatisations, i.e., given by a finite set of formulae. HDTP aligns pairs of formulae from the two domains by means of anti-unification: Anti-unification tries to solve the problem of generalizing terms in a meaningful way, yielding for each term an anti-instance, in which distinct subterms have been replaced by variables (which in turn would allow for a retrieval of the original terms by a substitution of the variables by appropriate subterms).[4] As of today, HDTP extends classical first-order anti-unification to a restricted form of higher-order anti-unification, as mere first-order structures have shown to be too weak for the purpose of analogy-making [13]: Just think of structural commonalities which are embedded in different contexts, and therefore not accessible by first-order anti-unification only.

Restricted higher-order anti-unification as presently used in HDTP was introduced in [16]. In order to restrain generalizations from becoming arbitrarily complex, a new notion of substitution is introduced. Classical first-order terms are extended by the introduction of variables which may take arguments (where classical first-order variables correspond to variables with arity 0), making a term either a first-order or a higher-order term. Then, anti-unification can be applied analogously to the original first-order case, yielding a generalization subsuming the specific terms. As already indicated by the naming, the class of substitutions which are applicable in HDTP is restricted to (compositions of) the following four cases: renamings, fixations, argument insertions, and permutations (see Def. 3 below). In [16], it is shown that this new form of (higher-order) substitution is a real extension of the first-order case, which has proven to be capable

[4] In [15], Plotkin demonstrated that for a proper definition of generalization, for a given pair of terms there always is a generalization, and that there is exactly one least general generalization (up to renaming of variables).

of detecting structural commonalities not accessible to first-order anti-unification. On the downside, in the higher-order case, the least general generalization loses its uniqueness. Therefore, the current implementation of HDTP ranks generalizations according to a complexity measure on generalizations (which in turn is based on a complexity measure for substitutions), and finally chooses the least complex generalizations as preferred ones [17].

In order to anti-unify not only terms, but formulae, HDTP extends the notion of generalization also to formulae by basically treating formulae in clause form and terms alike (as positive literals are structurally equal to function expressions, and complex clauses in normal form may be treated component wise). Furthermore, analogies do in general not only rely on an isolated pair of formulae from source and target, but on two sets of formulae. Here, heuristics are applied when iteratively selecting pairs of formulae to be generalized: Coherent mappings outmatch incoherent ones, i.e., mappings in which substitutions can be reused are preferred over isolated substitutions, as they are assumed to be better suited to induce the analogical relation . Once obtained, the generalized theory and the substitutions specify the analogical relation, and formulae of the source for which no correspondence in the target domain can be found may be transferred, constituting a process of analogical transfer between the domains.

The HDTP framework has successfully been tested in different application scenarios, and its use in several others has been proposed and theoretically grounded. Amongst others [18] shows a way how HDTP can be applied to model analogical reasoning in mathematics by a case study on the inductive analogy-making process involved in establishing the fundamental concepts of arithmetic, [14] applies HDTP to conceptual blending in the mathematics domain by providing an account of a process by which different conceptualizations of number can be blended together to form new conceptualizations via recognition of common features, and judicious combination of distinctive ones. On the more theoretical side, [19] considers how the framework could fruitfully be applied to modeling human decision-making and rational behaviour, [20] elaborates on how to expand HDTP into a domain-independent framework for conceptual blending, and [21] provides considerations on the applicability of HDTP in computational creativity.

The Complexity of HDTP: Results, Interpretation, and Implications

This section continues our work originally started in [5]. There, for the first time parametrized complexity results of HDTP had been presented. As basis for this analysis we had used the observation that HDTP can naturally be split into two distinct mechanisms, namely the analogical matching of input theories, and the re-representation of input theories by deduction in First-Order Logic (FOL). Clearly, from a complexity point of view, this type of re-representation is undecidable due to the undecidability of FOL. Therefore the analysis focused on the analogical matching mechanism only.

In the following, after introducing some necessary terminology and concepts, we provide a compact reproduction of the main results from [5] concerning the current version and implementation of HDTP as described in [13] as basis for further discussion. Then, we will add new approximation algorithmic theoretic considerations to the study of HDTP before proceeding with a general discussion and more detailed interpretation of the overall complexity theoretic insights (including both, results taken from [5] and our newly added analysis) against the background of the Tractable AGI thesis.

Terms and substitutions in restricted higher-order anti-unifications

HDTP uses many-sorted term algebras to define the input conceptual domains.

Definition 1. Many-sorted signature
A many-sorted signature $\Sigma = \langle Sort, Func \rangle$ is a tuple containing a finite set *Sort* of *sorts*, and a finite set *Func* of *function symbols*. An n-ary function symbol $f \in Func$ is specified by $f : s_1 \times s_2 \times \cdots \times s_n \to s$, where $s, s_1, \ldots, s_n \in Sort$. We will consider function symbols of any non-negative arity, and we will use 0-ary function symbols to represent *constants*.

Definition 2. Terms in HDTP
Let $\Sigma = \langle Sort, Func \rangle$ be a many-sorted signature, and let $\mathcal{V} = \{x_1 : s_1, x_2 : s_2, \ldots\}$ be an infinite set of sorted variables, where the sorts are chosen from *Sort*. Associated with each variable $x_i : s_i$ is an *arity*, analogous to the standard arity of function symbols. For any $i \geq 0$, we let $\mathcal{V}_i$ be the variables of arity i.
The set $Term(\Sigma, \mathcal{V})$ and the function $sort : Term(\Sigma, \mathcal{V}) \to Sort$ are defined inductively as follows:

1. If $x : s \in \mathcal{V}$, then $x \in Term(\Sigma, \mathcal{V})$ and $sort(x) = s$.
2. If $f : s_1 \times s_2 \times \cdots \times s_n \to s$ is a function symbol in Σ, and $t_1, \ldots, t_n \in Term(\Sigma, \mathcal{V})$ with $sort(t_i) = s_i$ for each i, then $f(t_1, \ldots, t_n) \in Term(\Sigma, \mathcal{V})$ with $sort(f(t_1, \ldots, t_n)) = s$.

We now fix one term algebra and introduce the term substitutions and generalizations allowed in restricted higher-order anti-unification [13].

Definition 3. Substitutions in restricted higher-order anti-unification

1. A renaming $\rho(F, F')$ replaces a variable $F \in \mathcal{V}_n$ with another variable $F' \in \mathcal{V}_n$:
   $F(t_1, \ldots, t_n) \xrightarrow{\rho(F,F')} F'(t_1, \ldots, t_n)$.
2. A fixation $\phi(F, f)$ replaces a variable $F \in \mathcal{V}_n$ with a function symbol $f \in \mathcal{C}_n$:
   $F(t_1, \ldots, t_n) \xrightarrow{\phi(F,f)} f(t_1, \ldots, t_n)$.
3. An argument insertion $\iota(F, F', V, i)$ is defined as follows, where $F \in \mathcal{V}_n, F' \in \mathcal{V}_{n-k+1}, V \in \mathcal{V}_k, i \in [n]$:
   $F(t_1, \ldots, t_n) \xrightarrow{\iota(F,F',V,i)}$
   $F'(t_1, \ldots, t_{i-1}, V(t_i, \ldots, t_{i+k-1}), t_{i+k}, \ldots, t_n)$.
   It "wraps" k of the subterms in a term using a k-ary variable, or can be used to insert a 0-ary variable.
4. A permutation $\pi(F, \tau)$ rearranges the arguments of a term, with $F \in \mathcal{V}_n$, $\tau : [n] \to [n]$ a bijection:
   $F(t_1, \ldots, t_n) \xrightarrow{\pi(F,\tau)} F(t_{\tau(1)}, \ldots, t_{\tau(n)})$.

A *restricted substitution* is a substitution which results from the composition of any sequence of unit substitutions.

Previous results on the parametrized complexity of HDTP

In [5], we defined three (increasingly complex and expressive) versions of higher-order anti-unification by successively admitting additional types of unit substitutions to be included in the anti-unification process, and subsequently analyzed the computational complexity of the resulting notions.

Problem 1. F Anti-Unification
Input: Two terms f, g, and a natural $k \in \mathbb{N}$
Problem: Is there an anti-unifier h, containing at least k variables, using only renamings and fixations?

Problem 2. FP Anti-Unification
Input: Two terms f, g, and naturals $l, m, p \in \mathbb{N}$.
Problem: Is there an anti-unifier h, containing at least l 0-ary variables and at least m higher arity variables, and two substitutions σ, τ using only renamings, fixations, and at most p permutations such that $h \xrightarrow{\sigma} f$ and $h \xrightarrow{\tau} g$?

Problem 3. FPA Anti-Unification
Input: Two terms f, g and naturals $l, m, p, a \in \mathbb{N}$.
Problem: Is there an anti-unifier h, containing at least l 0-ary variables, at least m higher arity variables, and two substitutions σ, τ using renamings, fixations, at most p permutations, and at most a argument insertions such that $h \xrightarrow{\sigma} f$ and $h \xrightarrow{\tau} g$?

Theorem 1. Complexity of HDTP Analogy-Making I
1.) F Anti-Unification is solvable in polynomial time.
2.) FP Anti-Unification is NP-complete and W[1]-hard w.r.t. parameter set $\{m, p\}$.
3.) Let r be the maximum arity and s be the maximum number of subterms of the input terms. Then FP Anti-Unification is in FPT w.r.t. parameter set $\{s, r, p\}$.
4.) FPA Anti-Unification is NP-complete and W[1]-hard w.r.t. parameter set $\{m, p, a\}$.

With respect to the original version of HDTP, [5] also offers some insight. As detailed in [12], instead of using restricted higher-order anti-unifications, HDTP initially was based on a mechanism reducing higher-order to first-order anti-unifications by introducing subterms built from what was called "*admissible sequences*":

Definition 4. Admissible Sequence
Let $Term(\Sigma, \mathcal{V})$ be a term algebra. Given a term t denote the set of all subterms of t as $st(t)$ and the set of variables in t as $var(t)$.
Let X be a set of terms. A set $\{t_1, \ldots, t_n\} \subseteq X$ is called *admissible relative to X* if $\bigcup_{i=1}^{n} var(t_i) = \bigcup_{t \in X} var(t)$.

Problem 4. Function Admissible-Sequence
Input: A term $f(t_1, t_2, \ldots, t_n) \in Term(\Sigma, \mathcal{V})$, a natural $k < |st(f(t_1, t_2, \ldots, t_n))|$
Problem: Is there a set $X \subseteq st(f(t_1, t_2, \ldots, t_n))$ such that $|X| \leq k$ and X is admissible relative to $st(f(t_1, t_2, \ldots, t_n))$?

Unfortunately, also this version of analogy-making in HDTP turns out to be intractable.

Theorem 2. Complexity of HDTP Analogy-Making II
Function Admissible-Sequence is W[2]-Hard (and NP-Complete) w.r.t. parameter k.

Approximation theoretic complexity of HDTP

Before presenting our approximation theoretic results, we have to introduce some technical machinery. For doing so, we presuppose basic knowledge of and familiarity with fundamental concepts from approximation theory as can, e.g., be obtained from the introduction given in [22].

In the following, let PTAS denote the class of all NP optimization problems that admit a polynomial-time approximation scheme, let APX be the class of NP optimization problems allowing for constant-factor approximation algorithms, and let APX-*poly* be the class of NP-optimization problems allowing for polynomial-factor approximation algorithms. We also have that PTAS $\subseteq$ APX $\subseteq$ APX-*poly* (with each inclusion being proper in case P $\neq$ NP).

Problem 5. MAXCLIQUE
Input: An n vertex, m-edge graph G
Problem: Compute and return the maximal clique in G.

Theorem 3. Approximation of MAXCLIQUE [23]
MAXCLIQUE is NP-hard to approximate to below $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$.

This theorem, combined with the NP-hardness of CLIQUE, implies that MAXCLIQUE is not in APX-*poly*.

From Theorem 1 we know that FP Anti-Unification is W[1]-hard to compute for a parameter set m, p, where m is the number of higher-arity variables and p is the number of permutations used. From the point of view of complexity theory, this shuts the door on any polynomial-time algorithm to compute generalizations which are "sufficiently complex" (i.e., with a lower bound on the number of higher-arity variables) while, simultaneously, upper bounding the number of permutations (according to [5] W[1]-hardness already is given for a single permutation). Although this is a strong negative result, it begs the following question: What if one considers generalizations which merely approximate the "optimal" generalization in some sense – and what is the right way to measure the quality of generalizations in the first place?

In [16], a measure of complexity for any composition of the substitutions allowed in the context of restricted higher-order anti-unification was introduced.

Definition 5. Complexity of a substitution
The complexity of a basic substitution σ is defined as $C(s) = \begin{cases} 0, & \text{if } \sigma \text{ is a renaming.} \\ 1, & \text{if } \sigma \text{ is a fixation or permutation.} \\ k+1, & \text{if } \sigma \text{ is a } k\text{-ary argument insertion.} \end{cases}$
The complexity of a restricted substitution $\sigma = \sigma_1 \circ \cdots \circ \sigma_n$ (i.e., the composition of any sequence of unit substitutions) is the sum of the composed substitutions: $C(\sigma) = \sum_{i=1}^{n} C(\sigma_i)$.

Consider the problem of finding a generalization which maximizes the complexity over all generalizations. First it should be noted that this may not be without merit: Intuitively, a complex generalization would contain the "most information" present over all of the generalizations chosen (i.e., one may think of this as the generalization maximizing the "information load", which is another idea put forward in [16]).

But now, taking the proof of the $W[1]$-hardness of FP anti-unification which made use of the maximum clique problem (MAXCLIQUE), we also can obtain an inapproximability result: It is known that MAXCLIQUE is NP-hard to approximate under a multiplicative factor of $O(n^{1-\varepsilon})$ for any $\varepsilon > 0$ [23]. This implies that MAXCLIQUE $\notin$ APX if P $\neq$ NP, and especially that it is hard for the class APX-poly. Finally, taking into account that the reduction given in [5] is actually approximation preserving, we may state that:

Theorem 4. Complexity of HDTP Analogy-Making III
FP anti-unification is not in APX.

Concerning an interpretation of the results presented in the last section and this one, especially when also having the Tractable AGI thesis in mind, several points should be emphasized. First, note that the W[2]-hardness of the function admissible-sequence problem clearly shows that problems can already arise when only treating with reductions from higher-order to first-order anti-unification. And also the result showing that FP higher-order anti-unification is W[1]-hard gives a hint at the difficulty introduced by the operations admissible within the restricted higher-order anti-unification on the complexity of the analogy-making process. Indeed, the only way that FP anti-unification can restructure the order of the terms is by argument permutations, and the results show that even allowing as few as one single permutation is enough to imply computational hardness. Unfortunately, the situation does not improve when not only considering precise solutions to the problem, but also taking into account approximation techniques: As we just showed FP anti-unification stays computationally hard in that it does not allow for a polynomial-time approximation algorithm with approximation ratio bounded by a constant factor.

But we explicitly want to point out that this — although being a rather strong statement — should not be considered an exclusively negative result: The given parametrized complexity results specifically point out that there is a complexity "dichotomy", and that giving an efficient algorithm requires at least the restriction to a permutationless input (as permutation operations have been identified as sources of intractability by the analysis). On the positive side, Theorem 1, no. 1, shows that in the (restricted) case of F Anti-Unification there is a tractable algorithm, and by the Tractable AGI thesis it suggests that we cannot do much better using this model of analogy, thus also providing positive guidance for the development of a tractable model.

Moreover, the found restrictions and limitations also allow for a connection to results from experimental studies on human analogy-making. In [24], it is reported that anxiety and time pressure made participants of an analogical-reasoning experiment switch from a preference for complex relational mappings to simple attribute-based mappings, i.e., an impairment on available working memory (as known consequence of anxiety) and computation time caused a change from a complexity-wise more demanding strategy using complex relational structures to a simple single-attribute-based procedure. The computation of analogies between highly relational domains makes the use of permutations almost unavoidable, whereas exclusively attribute-based analogical mappings are more likely to be computable without requiring a re-ordering of the argument structure of functions — thus making the experimentally documented switch from the more complex relational to the simpler surface-based form of analogy-making seem quite natural and expectable in case of strongly restricted computational resources.

Regardless, the obtained complexity results cast a shadow over the ambitions of using HDTP as basis for the development of a general computational theory of creativity and a uniform, integrated framework of creativity as, e.g., hinted at in [21]: Scalability to problem domains other than small and rather strongly restricted example scenarios is at the best doubtful (even when "only" considering approximate solutions to the respective problems), and any deeper rooted form of cognitive adequacy (if this shall be considered as one of the models goals) seems implausible. Of course this does not mean that Heuristic-Driven Theory Projection as an approach to modeling and formalizing high-level cognitive capacities entirely has to be abandoned, but only qualifies the naive approach to creating a general cognitive system based on HDTP as almost certainly unfeasible. Provided that future research searches for and investigates possibilities of mitigating the computational intractability of the approach (for instance by redefining restricted higher-order anti-unification and the complexity of substitutions in a fitting manner, or by changing the underlying representation formalism into a computationally less demanding form), the limitations and barriers introduced by the previously given complexity theoretic results might be suspended or avoided.

Future Work & Conclusion

Concerning the introduction of the Tractable AGI thesis into cognitive systems research and artificial intelligence, this paper merely can be seen as a very first step, leaving ample space for further work: The presented thesis has to be further specified, its implications and ramifications have to be identified and discussed, and more than anything else its merit and applicability have to be further demonstrated and supported by additional examples and theoretical underpinnings. With respect to our working example HDTP and its analysis in complexity theoretic terms, the approximation theoretic parts of the analysis could be complemented and completed by further investigations starting out from a structural approximation perspective [25]. Here, the basic idea is shifting interest from solutions that approximate the optimal solution according to some objective function towards solutions which best "structurally" approximate the optimal solution (the term "structurally" needs to be defined on a problem-by-problem basis).

Concludingly, we again want to express our firm belief that methods originally taken from complexity theory can be fruitfully applied to models and theories in artificial intelligence and cognitive systems research, allowing on the one hand for in-depth analysis of particular systems and theories, and on the other hand for the formulation and evaluation of general guiding principles.

References

1. Wang, P., Goertzel, B.: Introduction: Aspects of Artificial General Intelligence. In Goertzel, B., Wang, P., eds.: Advances in Artificial General Intelligence - Proc. of the AGI Workshop 2006. Volume 157 of Frontiers in Artificial Intelligence and Applications., IOS Press (2007) 1–16
2. van Rooij, I.: The tractable cognition thesis. Cognitive Science **32** (2008) 939–984
3. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer (2006)
4. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer (1999)
5. Robere, R., Besold, T.R.: Complex Analogies: Remarks on the Complexity of HDTP. In: Twentyfifth Australasian Joint Conference on Artificial Intelligence. Volume 7691 of Lecture Notes in Computer Science., Springer (2012)
6. Schwering, A., Kühnberger, K.U., Kokinov, B.: Analogies: Integrating multiple cognitive abilities - guest editorial. Journal of Cognitive Systems Research **10**(3) (2009)
7. Hofstadter, D.: Epilogue: Analogy as the core of cognition. In Gentner, D., Holyoak, K., Kokinov, B., eds.: The Analogical Mind: Perspectives from Cognitive Science, Cambridge, MA, MIT Press (2001) 499–538

8. Hofstadter, D., Mitchell, M.: The copycat project: a model of mental fluidity and analogy-making. In Holyoak, K., Barnden, J., eds.: Advances in Connectionist and Neural Computation Theory. Volume 2: Analogical Connections., New York, NY, USA, Ablex (1994) 31–112
9. Falkenhainer, B., Forbus, K., Gentner, D.: The structure-mapping engine: Algorithm and examples. Artificial Intelligence **41**(1) (1989) 1 – 63
10. Gentner, D., Forbus, K.: MAC/FAC: A Model of Similarity-based Retrieval. Cognitive Science **19** (1991) 141–205
11. Gentner, D.: Structure-mapping: A theoretical framework for analogy. Cognitive Science **7**(2) (1983) 155–170
12. Gust, H., Kühnberger, K.U., Schmid, U.: Metaphors and Heuristic–Driven Theory Projection (HDTP). Theoretical Computer Science **354** (2006) 98–117
13. Schwering, A., Krumnack, U., Kühnberger, K.U., Gust, H.: Syntactic principles of Heuristic-Driven Theory Projection. Journal of Cognitive Systems Research **10**(3) (2009) 251–269
14. Guhe, M., Pease, A., Smaill, A., Martinez, M., Schmidt, M., Gust, H., Kühnberger, K.U., Krumnack, U.: A computational account of conceptual blending in basic mathematics. Journal of Cognitive Systems Research **12**(3) (2011) 249–265
15. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence **5** (1970) 153–163
16. Krumnack, U., Schwering, A., Gust, H., Kühnberger, K.: Restricted higher-order anti-unification for analogy making. In: Twentieth Australian Joint Conference on Artificial Intelligence, Springer (2007)
17. Schmidt, M., Gust, H., Kühnberger, K.U., Krumnack, U.: Refinements of restricted higher-order anit-unification for heuristic-driven theory projection. In: Proc. 34th Annual German Conference on Artificial Intelligence, Springer (2011)
18. Guhe, M., Pease, A., Smaill, A., Schmidt, M., Gust, H., Kühnberger, K.U., Krumnack, U.: Mathematical reasoning with higher-order anti-unifcation. In: Proc. of the 32st Annual Conference of the Cognitive Science Society. (2010) 1992–1997
19. Besold, T.R., Gust, H., Krumnack, U., Abdel-Fattah, A., Schmidt, M., Kühnberger, K.U.: An argument for an analogical perspective on rationality & decision-making. In van Eijck, J., Verbrugge, R., eds.: Proc. of the Workshop on Reasoning About Other Minds (RAOM-2011). Volume 751 of CEUR Workshop Proceedings., CEUR-WS.org (July 2011) 20–31
20. Martinez, M., Besold, T.R., Abdel-Fattah, A., Kühnberger, K.U., Gust, H., Schmidt, M., Krumnack, U.: Towards a Domain-Independent Computational Framework for Theory Blending. In: AAAI Technical Report of the AAAI Fall 2011 Symposium on Advances in Cognitive Systems. (2011) 210–217
21. Martinez, M., Besold, T.R., Abdel-Fattah, A., Gust, H., Schmidt, M., Krumnack, U., Kühnberger, K.U.: Theory Blending as a Framework for Creativity in Systems for General Intelligence. In Wang, P., Goertzel, B., eds.: Theoretical Foundations of AGI, Atlantis Press (2012)
22. Vazirani, V.: Approximation Algorithms. Springer (2001)
23. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic numbers. In: Proc. of the 38th ACM Symposium on Theory of Computing. (2006) 681–690
24. Tohill, J., Holyoak, K.: The impact of anxiety on analogical reasoning. Thinking & Reasoning **6**(1) (2000) 27–40
25. van Rooij, I., Wareham, T.: Intractability and approximation of optimization theories of cognition. Journal of Mathematical Psychology **56**(4) (2012) 232–247