Advances in Cognitive Systems 4 (2016) 13-24 Submitted 8/2015; published 6/2016

Why Generality Is Key to Human-Level Artificial Intelligence

Tarek R. Besold TAREKRICHARD.BESOLD@UNIBZ.IT
The KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy

Ute Schmid UTE.SCHMID@UNI-BAMBERG.DE
Faculty Information Systems and Applied Computer Science, University of Bamberg, Germany

Abstract

In this essay, we advocate the position that research efforts working towards solving human-level AI (HLAI) necessarily must rely on general mechanisms and (models of) cognitive capacities, whereas domain-specific systems or task-dependent approaches offer only minor help towards the ultimate goal. We revisit psychological research on intelligence and the application of psychometric methods in AI, followed by a discussion of actual cognitive systems in light of the previous conceptual considerations. As part of these elaborations, we examine HDTP and IGOR2 in some detail as prototypical approaches that model a general capacity or implement a general mechanism. In the conclusion, we point out four characteristics that we consider generally desirable properties of HLAI systems.

1. Introduction: Intelligence, Cognition, and Computer Systems

When asked for a definition of AI as a field of study and its aims, one possible answer would be a variation of Nilsson (2009)'s statement: AI is that science devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment. This implies an understanding of AI that is very inclusive, introducing a continuum of capacity levels ranging from fairly low-level technological systems and lower animals at the one end to humans (and possibly beyond) on the other. So called "strong" or human-level AI (HLAI) research is commonly situated at the latter part of the spectrum, aiming at machines on par with humans in that they are comparable in their capacity to—among many others—reason, pursue and achieve goals, perceive and respond to different types of environmental stimuli, process information, or engage in scientific and creative activities.

This line of investigation goes back to the origin of AI research. In the 1956 Dartmouth proposal, McCarthy et al (2006) laid out the program for generations of researchers to follow: *The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.* Although there might be disagreement about the detailed interpretation of "precisely described" or "simulate", HLAI as a field still rests on the assumption that the (re-)creation of higher-level human capacities is possible and will eventually be achieved (Besold, 2013; Kühnberger & Hitzler, 2009). Researchers in "weak" AI and computational cognitive modelling, on the other hand, usually make

the weaker assumption that computability provides an appropriate conceptual apparatus for theories of the mind. That is, computational models can simulate human information processes, thereby either providing tools that take over specific functions or allow detailed and consistent generative descriptions of aspects of cognition (Johnson-Laird, 1988).

As we will show, these different foci must have important ramifications for the type of methods and approaches studied, as well as for what cognition and intelligence mean in the respective research context: *While standard AI can confine itself to the modelling and study of individual mental capacities as isolated subparts of the mind, HLAI necessarily must take a general and holistic approach to intelligence and cognition.*

Three conceptual cornerstones are shared by most, if not all, current endeavours in HLAI research: A computational theory of mind, the Church-Turing thesis, and (one dimension of) the physical symbol system hypothesis (PSSH). The computational theory (Pylyshyn, 1980) bridges the gap between humans and computers by advocating that the human mind and brain can be seen as an information-processing system, and that reasoning and thinking are processes that adhere with computation as formal symbol manipulation. The Church-Turing thesis (Kleene, 1952) adds limitations on the computational power of such a system by establishing Turing computability as a valid characterization of computability in general. The PSSH (Newell, 1980) operates on a different level: taking the computational characteristic of cognition and intelligence as given, it states as general criterion that "[t]he necessary and sufficient condition for a physical system to exhibit general intelligent action is that it be a physical symbol system". For the current discussion, the "sufficient" part is relevant as it opens the way to machine intelligence. While remaining uncommitted to the particular computational paradigm used for implementing them, rule-based and rule-following systems are deemed capable of providing a valid framework for (re-)creating general intelligence.[1]

Although different in nature and level of description, the three statements all claim generality and independence of domain or context. In our opinion, this is a necessary feature of all theoretical and methodological accounts of intelligence and is intimately connected to the general nature of intelligence itself. But this raises another issue. Although the term intelligence is commonplace, no unanimously accepted definition exists (Sternberg & Detterman, 1986). The characterization and measurement of intelligence has been delegated to the field of psychometrics, which studies the theory of psychological measurement, as well as the development and implementation of the corresponding instruments and procedures. Typically, intelligence tests consist of subtests that address different aspects of cognition, such as visual-spatial, verbal-linguistic, and logical-mathematical abilities (Sternberg, 2000). On the other hand, Raven (2000)'s culture-free Progressive Matrices incorporate only one type of item. Theorists following Spearman (1927) emphasise an

1. In the terminology of Derthick and Plaut (1986), rule-based systems are characterized by explicitly encoded rules, rule-following systems by implicitly encoded ones that, although not explicitly represented, still manifest themselves in the rule-following nature of the system behavior. An example for the latter class is Rumelhart and McClelland (1986)'s connectionist system for phonological representations of the past tense of English verbs from the present tense form. Due to the lack of explicit rules, the rule-following type of system is at first excluded from the class of physical symbol systems. Still, if the symbol level is interpreted as explaining knowledge-level behavior rather than directly implementing it, one can maintain a weakened version of the PSSH. A rule-based approach can then also provide an explanation for intelligence in the case of rule-following systems.

all-encompassing general factor of intelligence that is prerequisite for more specific abilities. This perspective corresponds to the notion of generality in HLAI.

Probably the most noted and ambitious research program applying tests from psychometrics to HLAI comes from Bringsjord and Schimanski (2003):

> Psychometric AI is the field devoted to building information-processing entities capable of at least solid performance on all established, validated tests of intelligence and mental ability, a class of tests that includes not just the rather restrictive IQ tests, but also tests of artistic and literary creativity, mechanical ability, and so on.

Although this proposal challenges AI to develop general-purpose systems that can solve a variety of problems, it does not explicitly rule out attempts that employ patchwork systems with many specialized "island solutions" (Besold, 2014a). When looking at computational approaches to intelligence tests (Hernández-Orallo et al., 2016), indeed most only address one specific class of problems.

In the sections that follow, we review two computational systems that illustrate this holistic approach to intelligence and cognition. First we introduce the analogy system HDTP as a computational model that exhibits cross-domain cognitive capacities. Second, we describe IGOR2, a program learning system that incorporates a general mechanism for inductive inference. After this, we briefly discuss selected other cognitive systems with claims to generality. We conclude with a short account of what we consider desirable properties for architectures aiming at HLAI.

2. Towards General Capacities: HDTP and Analogy

We have argued that, to avoid reliance on many inflexible "island solutions", accounts of cross-domain general capacities are necessary in order to guarantee generalizability and domain-independent functionality. A paradigmatic example for such a high-level capacity is analogical reasoning as modelled by the Heuristic-Driven Theory Projection (HDTP) framework (Schmidt et al., 2014).

HDTP is a conceptual framework and implemented analogy engine that relies on the computation of explicit generalizations between domains. It is similar to the classical SME (Falkenhainer et al., 1989) in that both are symbolic, operating over logic-like languages, and rely heavily on syntactic properties of domain elements to determine mappings (Wiese et al., 2008). However, HDTP computes an explicit generalization of the source and target domain theories into a least-general subsuming theory that later determines the transfer/evaluation phase of the analogy process. Figure 1 depicts HDTP's generalization-based view on analogy making. Given a source and a target domain, it computes a common generalization that establishes correspondences between domain elements using a restricted form of higher-order anti-unification (Schwering et al., 2009). The system uses these correspondences to transfer and adapt knowledge from the better-informed source to some target domain.

To clarify HDTP's analogy process, Table 1 and Table 2 reproduce a classical example from the literature—Rutherford's analogy between the solar system and the atom, establishing a conceptual equivalence between the atom's nucleus and the solar system's sun, the electrons and the planets, and so forth. The system uses the enriched generalization in Table 2 to instantiate an augmented

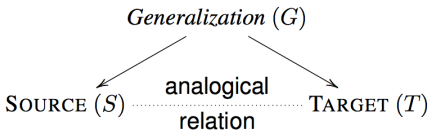

Figure 1. A schematic overview of HDTP's generalization-based approach to analogy.

version of the target theory which also features governing laws for the atomic model about the revolution of electrons in orbit around the nucleus.

Table 1. Formalization of the solar system model (source domain) and Rutherford's atom model (target domain) as used by HDTP.

Sorts: *real*, *object*, *time*
Entities: *sun*, *planet*, *nucleus*, *electron* : *object*
Shared functions of both theories: *mass* : *object* $\rightarrow$ *real* $\times$ $\{kg\}$, *dist* : *object* $\times$ *object* $\times$ *time* $\rightarrow$ *real* $\times$ $\{m\}$
Functions of the solar system theory: *force* : *object*$\times$*object*$\times$*time* $\rightarrow$ *real*$\times\{N\}$, *gravity* : *object*$\times$*object*$\times$*time* $\rightarrow$ *real*$\times\{N\}$, *centrifugal* :*object* $\times$ *object* $\times$ *time* $\rightarrow$ *real* $\times$ $\{m\}$
Predicates of the solar system theory: *revolves_around* : *object* $\times$ *object*
Facts of the solar system theory: (α_1) *mass(sun)* $>$ *mass(planet)*, (α_2) *mass(planet)* > 0, (α_3) $\forall t$: *time* : *gravity(planet, sun, t)* > 0, (α_4) $\forall t$: *time* : *dist(planet, sun, t)* > 0
Laws of the solar system theory:
(α_5) $\forall t$: *time*, o_1 : *object*, o_2 : *object* : *dist*$(o_1, o_2, t) > 0 \wedge$ *gravity*$(o_1, o_2, t) > 0$ $\rightarrow$ *centrifugal*$(o_1, o_2, t) = -$*gravity*(o_1, o_2, t),
(α_6) $\forall t$: *time*, o_1 : *object*, o_2 : *object* : $0 <$ *mass*$(o_1) <$ *mass*$(o_2) \wedge$ *dist*$(o_1, o_2, t) > 0 \wedge$ *centrifugal*$(o_1, o_2, t) < 0$ $\rightarrow$ *revolves_around*(o_1, o_2)
Functions of the atom model theory: *coulomb* : *object* $\times$ *object* $\times$ *time* $\rightarrow$ *real* $\times$ $\{N\}$
Facts of the atom model theory: (β_1) *mass(nucleus)* $>$ *mass(electron)*, (β_2) *mass(electron)* > 0, (β_3) $\forall t$: *time* : *coulomb(electron, nucleus, t)* > 0, (β_4) $\forall t$: *time* : *dist(electron, nucleus, t)* > 0

Because HDTP relies on many-sorted first-order logic and purely syntactic generalization, the framework has shown remarkable generality. The system was originally designed for modelling the Rutherford analogy, poetic metaphors (Gust et al., 2006), and heat flow (Falkenhainer et al., 1989; Schwering et al., 2009), but it has since been applied without major changes to other tasks in different domains. As Besold (2014b) has recounted, HDTP has been used for modelling sensory-motor-based transfer learning in mathematics and physics education, knowledge transfer between disparate domains (embodied sensory-motor interaction and abstract, structured knowledge) and subsequent concept formation. Also, Guhe et al. (2010) have shown how HDTP can model an inductive process for establishing the fundamental concepts of arithmetic starting from concrete experience.

Besides knowledge transfer and concept formation, another related but distinct capacity that HDTP can model is concept blending (Fauconnier & Turner, 1998) of theories. This is an important form of combinatorial creativity that joins familiar ideas in an unfamiliar way to produce novel ones. Martinez et al. (2012) outline HDTP's perspective on concept blending and reconstruct Argand's discovery of the complex plane, a famous example of theory blending. On a related note, Guhe et al. (2011) describe a process blending different conceptualizations of number to form new

Table 2. Common generalization between models of the solar system and the Rutherford atom, enriched by transfer axioms γ_5 and γ_6: Axioms α_5 and α_6 from the source domain—although not matched by corresponding axioms in the target theory—give rise to γ_5 and γ_6 by reuse of anti-unifications established at earlier stages. In particular, one generalizes $gravity$ and $coulomb$ from α_3 and β_3, respectively, to the generic variable F, and then reapplies the $gravity \rightarrow F$ anti-unification in generalizing α_5 into γ_5. The predicate $revolves\_around$ from α_6 is unmatched but carried over to γ_6.

Sorts:	*real*, *object*, *time*
Entities:	*X*, *Y* : *object*
Functions:	*mass* : *object* $\rightarrow$ *real* $\times$ {*kg*}, *dist* : *object* $\times$ *object* $\times$ *time* $\rightarrow$ *real* $\times$ {*m*}, *F* : *object* $\times$ *object* $\times$ *time* $\rightarrow$ *real* $\times$ {*N*}, *centrifugal* :*object* $\times$ *object* $\times$ *time* $\rightarrow$ *real* $\times$ {*m*}
Predicates:	*revolves_around* : *object* $\times$ *object*
Facts:	(γ_1) *mass(X)* $>$ *mass(Y)*, (γ_2) *mass(Y)* > 0, (γ_3) $\forall t$: *time* : *F(X, Y, t)* > 0, (γ_4) $\forall t$: *time* : *dist(X, Y, t)* > 0
Laws:	
(γ_5) $\forall t$: *time*, o_1 : *object*, o_2 : *object* : *dist*$(o_1, o_2, t) > 0 \wedge$ *F*$(o_1, o_2, t) > 0$ $\rightarrow$ *centrifugal*$(o_1, o_2, t) =$ *−F*(o_1, o_2, t),	
(γ_6) $\forall t$: *time*, o_1 : *object*, o_2 : *object* : $0 <$ *mass*$(o_1) <$ *mass*$(o_2) \wedge$ *dist*$(o_1, o_2, t) > 0 \wedge$ *centrifugal*$(o_1, o_2, t) < 0$ $\rightarrow$ *revolves_around*(o_1, o_2)	

conceptualizations via recognition of common features and judicious combination of distinctive ones.

In summary, HDTP's capacities of analogy, cross-domain generalization, cross-domain specialization, and detection of congruence relations lets the framework support new cognitive faculties without significantly changing the underlying model. This versatility results from several design decisions that set the framework apart from most other computational models of analogy. The choice of many-sorted first-order logic as a very expressive representation language allows modelling of rich domains with complex internal structure and recursively-defined relations among domain elements. Also its generalization-based approach provides—through the explicit availability of the generalization over domain theories—information and structure that can be reused for addressing many tasks beyond analogy that are essential for other key capacities such as learning.

3. Towards General Mechanisms: IGOR2 and Program Learning

IGOR is a system for inductive programming, that is, for the automatic generation of computer programs from incomplete specifications, typically input/output examples. Since program construction from examples is based on inductive inference, it can be seen as a special area of machine learning: The input/output examples constitute the training data and the hypothesis formalism is usually a declarative programming language. The successor IGOR2 has a broader scope than most comparable systems (Kitzelmann, 2009) and it can learn functional programs for insertion sort, reverse, odd/even, multiplication by addition, and Fibonacci numbers from examples (Hofmann et al., 2009). For induction of a linear recursive function with one recursive call, IGOR2 typically needs only four examples. Table 3 illustrates its abilities: The input are examples for reversing a list and the output is a generalized program that returns the empty list if the input is empty. For nonempty lists, with x as first element of a list and xs as a (possibly empty) remainder, the *last* element of this list is inserted

Table 3. Inductive programming with IGOR2: Generalizing *reverse* from examples

Specification of Data Types and Target Function

```
data [a] = [] | a:[a]        reverse :: [a]-> [a]
```

I/O Examples

```
reverse  [] =  []        reverse [a,b]   = [b,a]
reverse [a] = [a]        reverse [a,b,c] = [c,b,a]
```

Generalized Program

```
reverse   []   = []        reverse (x:xs) = last (x:xs) : reverse(init (x:xs))
```

Automatically Induced Functions (*renamed* from $f1$, $f2$)

```
last [x]    = x                      init [a]    = []
last (x:xs) = last xs                init (x:xs) = x:(init xs)
```

(operator ':') in front of the reversed *init* list (the list without the first element). The subfunctions *last* and *init* are themselves induced automatically.

We can use this problem to illustrate the basic ideas of IGOR2's induction algorithm. Kitzelmann (2010) provides formal details. In addition to the input/output examples, the algebraic data type for the target function must be specified. For *reverse*, this is the data type *list* defined over arbitrary elements *a*. Algebraic data types are specified using constructors, which are a minimal set of functions from which instances of this type can be built. Here these are the empty list *[ ]* and the function ':' (*cons*), which inserts an element in front of a list. The target function *reverse* is specified to have a list as input and output. For the example, we omit the additional specification of background knowledge, but one could define the *last* function in advance and use it during program construction.

Hypothesis construction in IGOR2 is based on anti-unification of sets of equations. It constructs hypotheses in the form of partial programs by applying an induction operator and carryies out a best-first search that prefers hypotheses with fewer equations. Anti-unification of the four examples in Table 3 results in the overly general expression *reverse x = y*. The body of this function contains an unbound variable y that tells IGOR2 program induction has not yet terminated, and the system applies three induction operators to generate successor hypotheses: partitioning examples into sets of equations divided by case distinction; replacement of the right-hand side of an equation by a program call; and replacement of subterms with unbound variables by induced subfunctions. Partitioning results in separating the treatment of the empty list from the other three cases. For the remaining three cases, subfunctions are introduced (*reverse (x:xs) = f1(x:xs):f2(x:xs)*) and for each subfunction, new training examples are abduced:

```
f1 [a] = a        f1 [a,b] = b        f1 [a,b,c] = c
f2 [a] = []       f2 [a,b] = [a]      f2 [a,b,c] = [b,a].
```

Table 4. Samples of number series used to test IGOR2's induction abilities.

Constant	15 15 16 15 15 16 15	$f(n-3)$	Geometric	3 6 12 24	$f(n-1) \times 2$
Arithmetic	2 3 8 11 14	$f(n-1)+3$		6 7 8 18 21 24 54	$f(n-3) \times 3$
	1 2 3 12 13 14 23	$f(n-3)+11$		5 10 30 120 600	$f(n-1) \times n$
Fibonacci	1 2 3 5 8 13 21 34	$f(n-1)+f(n-2)$		3,7,15,31,63	$2 * f(n-1)+1$
	3 4 12 48 576	$f(n-1) \times f(n-2)$			

For both subfunctions, IGOR2 calls its induction process recursively. Function f_1 corresponds to the function *last* in Table 3 and function f_2 is elaborated to a recursive call of the target function *reverse* and another subfunction corresponding to *init*.

Schmid and Kitzelmann (2011) have applied IGOR2 to a variety of problem domains. Without any modification, the system can learn recursive solution procedures for the Tower of Hanoi, building a tower of sorted blocks, and the rocket problem (Veloso & Carbonell, 1993). Furthermore, it can learn simple phrase-structure grammars from example sentences and recursive relations such as *ancestor*, as well as the transitivity of the *is-a* relation in a concept hierarchy. More recently, Hofmann, Kitzelmann, and Schmid (2014) have applied IGOR2 to number series problems like those in many intelligence tests. Such problems aim to measure inductive reasoning ability, which some researchers claim to be a crucial component of general intelligence (Sternberg, 2000). Table 4 shows a selection of problems presented to, and successfully solved by, the system.

Although IGOR2 was designed specifically for inductive programming, the underlying approach offers a generic mechanism for generalizing productive rules from example experience. We use the term 'productive rule' following Chomsky (1959)'s characterization of human linguistic competence. A set of rules is productive when it can be applied to input of arbitrary complexity. For example, the *reverse* program in Table 3 has the competence to reverse lists of arbitrary length. Contrary to Chomsky, we propose that this human ability to generalize productive rules from a few observations is not restricted to language but applies to all areas where such knowledge can be obtained by detection of regularities.

Of course, IGOR2 is not the only possible approach to generalizing over regularities, but it has some characteristics that make it a better candidate than many others. First, the system learns on the symbol- or knowledge level (Rosenbloom et al., 1987). This makes it relevant to cognitive models that assume that information in working memory can be inspected and verbalized. Second, IGOR2's learning strategy uses not generate and test but a mechanism that builds hypotheses based on regularities detected in the observations. A cognitive system that generates many arbitrary hypotheses and checks them against examples seems implausible. Third, IGOR2 has no built-in heuristics that are tuned to a specific domain of application; it uses only a simple preference bias to guide search that prefers hypotheses with fewer rules and recursive calls.

4. Other Approaches Aiming for Generality

HDTP and IGOR2 are far from the only cognitive systems that aim for generality (see, e.g., Langley et al., 2009, for an overview). Here we discuss two paradigms and corresponding architectures that, in our opinion, also qualify as general approaches to HLAI, namely Langley and Choi's (2006) ICARUS and Forbus and Hinrich's (2006) COMPANION cognitive systems.

ICARUS is an integrated cognitive architecture for physical agents, storing concepts and skills as distinct forms of knowledge. Skills specify reactive goal-relevant reactions to classes of problems, while concepts provide percept-based relational descriptions of environmental situations. The architecture is subdivided into an inference module, a planning module, and an execution module. Conceptual memory stores information about general object classes and their relationships, while skill memory archives knowledge about ways of acting. Within this, ICARUS infers beliefs from percepts in a bottom-up way while connecting goals to skills and actions in a top-down fashion, operating on a recognition-action cycle. Although this constitutes a general set up without strong commitments to particular representations or classes of tasks, the architecture can also perform problem solving using a variant of means-end analysis. Through its planning and learning modules, ICARUS can incrementally learn new concepts and skills, which it can subsequently apply to handle similar situations reactively. This capability to acquire new abilities and to expand its conceptualization of the environment let it handle several quite disparate tasks and scenarios, ranging from classical problem solving hurdles like the Tower of Hanoi, logistics planning, FreeCell solitaire, simulated urban driving, and first-person shooter scenarios (Choi et al., 2007). Another agent-based cognitive system that we should mention in passing in this context is MIRCOPSI (Bach, 2009), which focuses on the impact of motivation and emotion on action selection: Although not directly inspired by ICARUS, there are clear conceptual similarities in that MICROPSI agents (which can model human problem solving) combine associative learning, reinforcement learning, and planning to acquire knowledge about their environment and navigate in pursuit of resources.

The COMPANION cognitive architecture aims to understand how to build intelligent systems that are social beings, making them different from skill-oriented architectures like ICARUS. The resulting systems collaborate with human users, working through complex arguments, automatically retrieving relevant precedents, providing cautions and counter-indications, and offering supporting evidence. This necessarily involves the incremental assimilation of new information and permanent learning about domains, users, and themselves. As Forbus, Klenk, and Hinrichs (2009) state, this is achieved by seven general features: (i) Generalization-based analogy lets the system learn from experience and reuse past experiences in new situations. (ii) Extensive conceptual knowledge provides a foundation for acquiring or learning domain content. (iii) Flexible, resource-sensitive reasoning takes into account constraints on memory and processing time. (iv) Coarse-grained parallel processing implements companions as distributed systems by allocating individual nodes of a cluster computer to a few semi-independent, asynchronous processes. (v) Costly learning tasks are offloaded to dedicated nodes in the background, while focused learning is guided by flexibly constructed, prioritized, and scheduled explicit learning goals. (vi) Extended lifetimes allow compute-intensive learning during periods with low general demands, switching between domains, and incremental construction of user and self models. (vii) Natural interaction modalities enable a

natural and unmediated interplay of companions with their environment as a basis for learning and interaction on the envisioned scale.

Although ICARUS and the COMPANION cognitive systems are based on different paradigms, architectures, and mechanisms, they share an ambition for generality and a conscious emphasis on task-independent and capacity-unspecific approaches. Also, the central role of powerful learning mechanisms, together with the capability to reuse past experiences and general representations, supports domain independence and cross-domain functionalities.

5. Conclusions

In this essay, we advocated the need for generality in approaches to HLAI: While standard AI and cognitive modelling can meaningfully restrict their focus to (re-)implementing individual cognitive capacities, the goal of creating general intelligence with artificial means necessarily requires general mechanisms and capacities. As discussed, this need for flexibility and cross-domain validity goes back to the very nature of intelligence and cognition as many-facetted phenomena studied in psychology and, taking inspiration from results there, seems desirable. The specificity and domain-centricity of individual psychometric tests should not distract us from the general nature of intelligence.

As positive examples for systems that cut across domains and tasks, we discussed HDTP and IGOR2, sketching several applications that illustrate their coverage. In our opinion, the key properties common to these architectures, and shared with ICARUS, MICROPSI, and COMPANION, are a good starting point for future endeavours in HLAI. They operate at the symbol level, using expressive representations that allow considerable freedom in the choice of domains. They are based on a very general, domain-independent mechanism, namely anti-unification that operates on such representations, that is not bound to particular tasks, and that is involved in many different processes, such as generalization and representation alignment. Both approaches explicitly target high-level, domain-independent capacities, namely analogy making and inductive learning, conceptually excluding the specialization of the computational-level theory with respect to specific domain or task properties. Finally, both systems can take into account semantic aspects of the domain content without requiring a case-specific approach, namely the algebraic data types used in IGOR2 and the many-sorted representations in HDTP. And while also HDTP and IGOR2 remain only initial steps towards truly general systems, we believe that, when taken together, these four properties form a solid foundation for development of systems that exhibit general intelligence and cognitive capacities.

Acknowledgements

The first author acknowledges the financial support of the Future and Emerging Technologies programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 611553 (COINVENT).

References

Bach, J. (2009). *Principles of synthetic intelligence: PSI, an architecture of motivated cognition*. New York: Oxford University Press.

Besold, T. R. (2013). Human-level artificial intelligence must be a science. In K.-U. Kühnberger, S. Rudolph, & P. Wang (Eds.), *Artificial General Intelligence*, 174–177. Berlin: Springer.

Besold, T. R. (2014a). A note on chances and limitations of psychometric AI. *KI 2014: Advances in Artificial Intelligence* (pp. 49–54). Cham: Springer International Publishing.

Besold, T. R. (2014b). Sensorimotor analogies in learning abstract skills and knowledge: Modeling analogy-supported education in mathematics and physics. *Proceedings of the AAAI Fall 2014 Symposium on Modeling Changing Perspectives: Reconceptualizing Sensorimotor Experiences*. Menlo Park, CA: AAAI Press.

Bringsjord, S., & Schimanski, B. (2003). What is artificial intelligence? Psychometric AI as an answer. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* (pp. 887–893). San Francisco, CA: Morgan Kaufmann.

Choi, D., Konik, T., Nejati, N., Park, C., & Langley, P. (2007). A believable agent for first-person shooter games. *Proceedings of the Third Annual Artificial Intelligence and Digital Entertainment Conference* (pp. 71–73). Menlo Park, CA: AAAI Press.

Chomsky, N. (1959). Review of Skinner's 'Verbal behavior'. *Language*, *35*, 26–58.

Derthick, M., & Plaut, D. C. (1986). Is distributed connectionism compatible with the physical symbol system hypothesis? *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* (pp. 639–644). Hillsdale, NJ: Lawrence Erlbaum.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, *41*, 1–63.

Fauconnier, G., & Turner, M. (1998). Conceptual integration networks. *Cognitive Science*, *22*, 133–187.

Forbus, K. D., & Hinrichs, T. R. (2006). Companion cognitive systems: A step toward human-level ai. *AI Magazine*, *27*, 83–95.

Forbus, K. D., Klenk, M., & Hinrichs, T. (2009). Companion cognitive systems: Design goals and lessons learned so far. *IEEE Intelligent Systems*, *24*, 36–46.

Guhe, M., Pease, A., Smaill, A., Martinez, M., Schmidt, M., Gust, H., Kühnberger, K.-U., & Krumnack, U. (2011). A computational account of conceptual blending in basic mathematics. *Journal of Cognitive Systems Research*, *12*, 249–265.

Guhe, M., Pease, A., Smaill, A., Schmidt, M., Gust, H., Kühnberger, K.-U., & Krumnack, U. (2010). Mathematical reasoning with higher-order anti-unification. *Proceedings of the Thirty-Second Annual Conference of the Cognitive Science Society* (pp. 1992–1997). Austin, TX: Cognitive Science Society.

Gust, H., Kühnberger, K.-U., & Schmid, U. (2006). Metaphors and Heuristic-Driven Theory Projection (HDTP). *Theoretical Computer Science*, *354*, 98–117.

Hernández-Orallo, J., Martínez-Plumed, F., Schmid, U., Siebers, M., & Dowe, D. L. (2016). Computer models solving intelligence test problems: Progress and implications. *Artificial Intelligence*, *230*, 74–107.

Hofmann, J., Kitzelmann, E., & Schmid, U. (2014). Applying inductive program synthesis to induction of number series a case study with IGOR2. *KI 2014: Advances in Artificial Intelligence* (pp. 25–36). Cham: Springer International Publishing.

Hofmann, M., Kitzelmann, E., & Schmid, U. (2009). A unifying framework for analysis and evaluation of inductive programming systems. *Proceedings of the Second Conference on Artificial General Intelligence*. Paris: Atlantis Press.

Johnson-Laird, P. N. (1988). *The computer and the mind: An introduction to cognitive science*. London: Fontana Press.

Kitzelmann, E. (2009). Analytical inductive functional programming. *Proceedings of the Eighteenth International Symposium on Logic-Based Program Synthesis and Transformation* (pp. 87–102). Berlin: Springer.

Kitzelmann, E. (2010). *A combined analytical and search-based approach to the inductive synthesis of functional programs*. Doctoral dissertation, Fakultät Wirtschaftsinformatik und Angewandte Informatik, Otto-Friedrich Universität Bamberg, Bamberg. From `http://www.opus-bayern.de/uni-bamberg/volltexte/2010/280/`.

Kleene, S. C. (1952). *Introduction to metamathematics*. Amsterdam: North-Holland.

Kühnberger, K. U., & Hitzler, P. (2009). Facets of artificial general intelligence. *KI - Künstliche Intelligenz*, *23*, 58–59.

Langley, P., & Choi, D. (2006). A unified cognitive architecture for physical agents. *Proceedings of the Twenty-First National Conference on Artificial Intelligence* (pp. 1469–1474). Boston, MA: AAAI Press.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*, 141–160.

Martinez, M., Besold, T. R., Abdel-Fattah, A., Gust, H., Schmidt, M., Krumnack, U., & Kühnberger, K.-U. (2012). Theory blending as a framework for creativity in systems for general intelligence. In P. Wang & B. Goertzel (Eds.), *Theoretical Foundations of Artificial General Intelligence*, 219–239. Paris: Atlantis Press.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). Reprint of "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955". *AI Magazine*, *27*, 12–14.

Newell, A. (1980). Physical symbol systems. *Cognitive Science*, *4*, 135–183.

Nilsson, N. J. (2009). *The quest for artificial intelligence*. New York: Cambridge University Press.

Pylyshyn, Z. (1980). Computation and cognition: Issues in the foundation of cognitive science. *The Behavioral and Brain Sciences*, *3*, 111–132.

Raven, J. (2000). The Raven's Progressive Matrices: Change and stability over culture and time. *Cognitive Psychology*, *41*, 1–48.

Rosenbloom, P. S., Laird, J. E., & Newell, A. (1987). Knowledge level learning in Soar. *Proceedings of the Sixth National Conference on Artificial Intelligence* (pp. 499–504). Los Altos, CA: Morgan Kaufmann.

Rumelhart, D., & McClelland, J. (1986). On learning the past tenses of English verbs. In J. McClelland & D. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*. Cambridge, MA: Bradford Books.

Schmid, U., & Kitzelmann, E. (2011). Inductive rule learning on the knowledge level. *Cognitive Systems Research*, *12*, 237–248.

Schmidt, M., Krumnack, U., Gust, H., & Kühnberger, K.-U. (2014). Heuristic-driven theory projection: An overview. In H. Prade & G. Richard (Eds.), *Computational approaches to analogical reasoning: Current trends*, 163–194. Berlin, Heidelberg: Springer Berlin Heidelberg.

Schwering, A., Krumnack, U., Kühnberger, K.-U., & Gust, H. (2009). Syntactic principles of Heuristic-Driven Theory Projection. *Journal of Cognitive Systems Research*, *10*, 251–269.

Spearman, C. (1927). *The abilities of man*. London: Macmillan.

Sternberg, R., & Detterman, D. K. (Eds.). (1986). *What is intelligence? contemporary viewpoints on its nature and definition*. Norwood, NJ: Ablex.

Sternberg, R. J. (Ed.). (2000). *Handbook of intelligence*. Cambridge, UK: Cambridge University Press.

Veloso, M., & Carbonell, J. (1993). Derivational analogy in Prodigy: Automating case acquisition, storage, and utilization. *Machine Learning*, *10*, 249–278.

Wiese, E., Konerding, U., & Schmid, U. (2008). Mapping and inference in analogical problem solving – As much as needed or as much as possible? *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Sociecty* (pp. 927–932). Mahwah, NJ: Lawrence Erlbaum.