Explicit General Analogy for Autonomous Transversal Learning

Arash Sheikhlar ARASH19@RU.IS
Center for Analysis & Design of Intelligent Agents, Dept. Comp. Sci., Reykjavik U.

Kristinn R. Thórisson THORISSON@{RU.IS, IIIM.IS}
Center for Analysis & Design of Intelligent Agents, Dept. Comp. Sci., Reykjavik U.
and Icelandic Institute for Intelligent Machines, Reykjavik, Iceland

Jeff Thompson JEFF@IIIM.IS
Icelandic Institute for Intelligent Machines, Reykjavik, Iceland

Editor: Kristinn R. Thórisson

Abstract

Making analogies is a kind of reasoning where two or more things are compared, to highlight or uncover attributes of interest. Besides being useful for comparing what is known, analogy making can help a learning agent deal with tasks and environments not experienced before, where similarities and differences to known phenomena and their cause-effects relations can be a source for generating hypotheses about novel phenomena, which in turn can serve as a basis for exploration and experimentation. Artificial intelligence (AI) systems that can make use of explicit analogies are relatively rare, and those making general analogies are even rarer. This may be because most AI systems are targeted to well-known tasks, relying heavily on human programmers for knowledge creation, an approach that – besides being intractably slow, error-prone, and highly ineffective – precludes the use of analogies for enabling autonomous knowledge transfer between tasks, domains, and environments with common characteristics. The automation of explicit analogy making in the service of such knowledge transfer has, in our view, at least three prerequisites: (a) Compositional knowledge representation, (b) reasoning machinery, and (c) the ability of the agent to make experiments on its surroundings. For an agent's intelligence to be general, the methods chosen for these must be domain-independent and available on-demand at the agent's discretion. The agent would identify a target novelty, generate hypotheses about what the novelty is 'like' through analogies, generate a set of hypotheses with potential to disqualify these and select between competing hypotheses, and intervene on the environment through direct action to test them. Here we describe the design of an analogy mechanism that allows a learning agent with the above features to autonomously, using previously-learned causal knowledge, make analogies between a source and target task, hypothesize sets of new causal models for performing the new tasks, and to verify the validity of these through a set of autonomously generated actions. We describe how this general approach can be implemented in an existing cognitive system, the Autocatlytic Endogenous Reflective Architecture (AERA).

Keywords: General Machine Intelligence, Cumulative Learning, Machine Learning, Knowledge Representation, Autonomy, Generality, Autonomous Generality, Artificial General Intelligence

1. Introduction

In the broadest sense of the concept, *analogy making* is a cognitive process where selected pieces of information are mapped from a source phenomenon to a target phenomenon, through an analysis of similarities and differences. Humans often use analogy in familiar situations as a guide to take proper actions in order to achieve active goals. Analogy enables a human learner to bring acquired and verified knowledge to bear on new tasks and environments that have not been encountered before, in ways that help generalize the knowledge to make it more useful in light of uncertainty, in present and future situations. Similarly, an artificial intelligence (AI) system targeting general intelligence calls for a *general* analogy-making mechanism to steer its action-taking in novel situations and tasks, potentially providing increased flexibility and faster learning in partially unknown environments (cf. Eberding et al., 2022). Such analogy making capability should be available for any and all represented information, i.e. applicable transversally throughout the system's operation and knowledge.

We see analogies being of at least two kinds, (i) those relating to structure and form (e.g. the composition and shape of two objects), and (ii) those relating to transformations or actions (e.g. tasks that are similar in terms of the particular chain of actions required to achieve goals). For the purpose of planning, understanding (Thórisson et al., 2016), and generating explanations (Thórisson, 2021b), the latter kind requires causal knowledge and reasoning, which is a necessary foundation for all reliable actionable knowledge. The physical world – which we consider a prime target for future general machine intelligence – contains a lot of self-similarity at multiple levels of detail (cf. Henriksen, 2015) that can be exploited for goal achievement; for instance, hand-size objects can be grasped by pressing fingers around them—larger objects can be grasped by pressing opposite arms around them.

For robots doing physical tasks, action-oriented analogies could allow them to efficiently discover and learn causal relations, to transfer these to similar situations. However, how to make a machine perform such action-oriented analogies automatically, using the learned knowledge, is a challenge that seems far from being fully solved, as it requires a knowledge representation that (a) allows automatic acquisition, (b) captures part-whole relationships, (c) represents transformations, (d) allows for disruption-free updating in light of new evidence, and of course (e) allows comparison of arbitrary information sets, in support of explicit analogy making.[1]

Contemporary machine learning approaches, e.g. reinforcement learning and deep learning (cf. Dong et al., 2021), rely heavily on prior training, and require up-front knowledge of all relevant variables. This training is mostly performed by human designers, because the methods are not capable of self-supervised learning. Their behavior can also be somewhat unpredictable when facing novel tasks and situations. Reinforcement Learning (RL) addresses environments in relation to goals, and is thus more relevant to action-oriented learning. In RL, goals are usually weakly defined as fixed reward functions, however, which limits their flexibility severely. RL correlates actions, states and rewards. The entanglement of actions and goals commonly leads to negative knowledge transfer where prior knowledge

1. We define "explicit" here to mean that the learner has the ability to produce explicit arguments for and against a particular plan, prediction, or action being performed. In other words, the ability to perform explicit analogies implies an ability to produce explicit explanations that reference causal relations, based on verified experience (cf. Thórisson, 2021b).

hinders, rather than helps. By disentangling goal and action representation, knowledge is more easily stored in a task-independent way, making it more general. This can be done by explicitly representing causal relations of environments (Pearl, 2018), putting another requirement on the knowledge representation used.

Thus, the aims of this paper are to (a) present a new learning mechanism that uses action-oriented analogies to autonomously hypothesize reasonable causal relations for performing a new task, which it subsequently verifies through direct action, after which it (b) reviews the outcome and revises its knowledge accordingly, unifying the new knowledge with what is already known, resulting in increased actionable knowledge. The analogy mechanism is *general* in that it is not limited to a particular task or domain, it is *transversal* in that it can involve any explicitly-represented knowledge an agent may posses, and it is *explicit* in that it is not a side-effect of opaque processes but compositional and, like software, can be inspected, dissected, manipulated, and directly compared in whole or in part to other information.

2. Related Work

A primary focus of AI research for the past 20 years has been on Artificial Neural Networks (ANNs) (cf. Mitchell, 2021). Analogy can be approximated in ANNs through manipulation of high-dimensional information vectors. For instance, Mikolov et al. (2013) present an ANN that learns regularities from written text, allowing it to make (surface-level) linguistic analogy. A domain for studying visual analogies in AI systems is called Raven's Progressive Matrices (RPMs) (Hu et al., 2021; Mitchell, 2021). In an RPM problem, the system observes a set of geometrical objects, their attributes, and the relation between the objects in a sequence. It then tries to guess the attributes of the last object. Over the past few years, various RPM datasets have been generated by which ANNs can be trained (Wang and Su, 2015; Zheng et al., 2019). Visual analogy via ANNs has also been applied to other domains, such as detecting similarities between images (Lu et al., 2008). However, it has been shown that the trained ANNs are incapable of making correct analogical inferences when facing new RPMs and images that have not been in the training datasets (Hu et al., 2021). Besides a lack of generalizability, one limitation of ANNs in this respect is that they require their human designers to train them (via their own intuition), which implies that ANNs cannot autonomously make analogies.

A handful of symbolic approaches regarding analogical reasoning have been proposed. One example is structure mapping engine (SME) takes propositional descriptions and some task constraints as inputs and then produces mappings between the descriptions (Falkenhainer et al., 1989). However, structure mapping theory (Gentner, 1983), which is the theoretical foundation of SME, does not take object properties into account for analogies. Moreover, SME is not designed for an interactive agent that changes the state of environments through direct interventions. Also, this approach is fully focused on performing comparisons in symbolic spaces where no variables and related values are involved. Case-based reasoning (CBR) is another symbolic approach to analogical reasoning, which has four steps: retrieval (finding a case similar to current situation), reuse (computing the action), revision (guessing what the outcome of the action will be and revising it), and retention (storing the result of the experience) (Aamodt and Plaza, 1994). Although this approach is particularly designed for

interactive agents, like SME it suffers from not taking into account values when calculating similarity.

In the context of reinforcement learning (RL) several papers have been published on automatic knowledge transfer methods that can find inter-task mappings based on homomorphism (Sorg and Singh, 2009), sparse coding (Ammar et al., 2012), reconstruction error of Boltzmann machine (Ammar et al., 2014), and bisimulation (Wang et al., 2019). Nevertheless, such systems have significant limitations including that (1) RL agents learn reward-entangled policies and models that are not generalizable to new tasks where the goals (and thus the reward functions) are different, and (2) the automatic transfer methods have fixed mapping structures, provided at design time by the RL agent's designer, that cannot be learned.

Causal knowledge representation and inference go beyond these limitations, proposing a way to achieve task-independence, empirical testability, and potential of dealing with missing data (Pearl, 2018). The philosophy of causality goes a long history going back, to 460 BC to Democritus, and was only formalized in the 20th century. A notable attempt at formalizing causality is provided by Pearl (2009), whose causal diagrams reflects invariant, physical attributes of environments that do not change across variations of tasks. However, since the topic of causality has attracted the AI community's attention only very recently, causal discovery and generalization methods are still limited to offline learning algorithms, which rely on large amounts of data that are already in hand and are not designed for autonomous learning agents that have interaction with environments (Zeng et al., 2021; Rojas-Carulla et al., 2018; Bengio et al., 2019). Another way of finding causal relations (a.k.a. causal discovery/learning) is through interventions (systematic actions) done by an agent that can intervene on dynamic environments (Sheikhlar et al., 2021). Such agents can transfer relevant causal knowledge autonomously to novel situations via estimating similarities between different situations (Sheikhlar et al., 2020).

3. Methodological Framework

Our analogy-based generalization scheme is proposed within the methodological framework of constructivism (Thórisson, 2012), causal knowledge representation (Thórisson and Talbot, 2018; Pearl, 2009), and cumulative learning (Thórisson et al., 2019). Originally inspired by Piaget's theory (1950) of how the human mind develops, constructivist principles and methodology addresses how an artificial agent, e.g. a robot, can grow its knowledge through direct experience (Steunebrink et al., 2016; Nivel et al., 2014; Drescher, 1989). Constructivism assumes that a learner is endowed with sufficient autonomy to allow it to change their own cognitive configuration whenever required, producing appropriate control structures on-demand at run-time. The constructivist principles we propose include methods for how to achieve compositionality, knowledge transparency, temporal grounding, looped-back self-control, autocatalytic runtime operation, autonomous pattern matching, semantic and operational closure, and meta-control (Thórisson, 2012; see further explanation below).

Another pillar for our analogy-making mechanism is causal inference. Using Pearl's (2009) causal graphs directly as a basis for autonomous AI systems is not possible as the necessary mechanisms responsible for Pearl's causal graphs from scratch remains to be proposed by the author. One approach to that challenge, however, is Nivel and Thórisson's

Causal-Relational Models (CRMs) (Thórisson, 2021a; Thórisson and Talbot, 2018; Nivel et al., 2013b). CRMs are machine-created and -manipulated information structures that describe bidirectional relationships between cause and effect variables, designed specifically to support autonomous generation, manipulation, and reasoning over causal knowledge needed for cumulative learning (Thórisson and Talbot, 2018; Eberding et al., 2021). CRMs are typically used in sets, in accordance with the rules of their compositionality; each CRM contains preconditions which specify under what circumstances it is relevant, target variables that its knowledge makes claims about, and the kind of changes to these variables that it captures. Clusters of such CRMs makes an agent's knowledge independent from its goals and initial conditions (Sheikhlar et al., 2021) and provides thus a task-independent reasoning capability for an AI system.

Via this approach, we can build systems that autonomously acquire knowledge and use it in multiple different but similar situations/tasks, without the help of a designer or teacher. Our method meets the following requirements:

- **Compositional, transparent, explainable knowledge:** The agent autonomously divides its experience into discrete but connected causal relations (a.k.a. CRMs) which allow it not only to deal with a variety of tasks more flexibly but also to use task-invariant similarities between knowledge structures when facing new tasks. To do so, the CRMs are formed as peewee-size models that can represent both micro- and macro-relations in a task-environment (TE). A TE is an environment where one or more tasks are assigned and conducted.[2] The CRMs are compositional in the sense that they can be used together with other fractional models, which allows them to to be (re-)used for different tasks with different overall causal structures, defined within the same or different environment where some sub-structures are shared. This knowledge representation is *explainable* due to the traceability of causal model chains at multiple levels of detail (beyond the acquired knowledge, through generalizations generated from, and verified by, experience).

- **Transversal temporal grounding:** The knowledge representation and reasoning of an autonomous learner must have a built-in transversal conceptualization of time so that particular situations experienced can be temporally related to each other. This allows the agent to predict, given that it takes one action at a time, how the state of other variables changes in the future. This implies that if the agent has learned relevant models it will become aware of the time horizon that is required for performing a task.

- **Feedback loops:** Action-based learning occurs via feedback through interaction with the environment; the agent can perform causal experiments (interventions with the purpose of causal knowledge discovery) and observe how the experiments change the state of the world, allowing the generation of CRMs patterned after the observations. Another feedback loop allows for evaluating the effectiveness of

2. We use the term 'task-environment,' rather than simply 'task' or 'environment,' because the separation between the two is usually not obvious (several relevant elements can typically be classified as being part of either). This term is intended to capture the set of all relevant components an environment in which various tasks can be assigned to an agent, and neutralize any confusion that otherwise might arise.

predictions and plans based on existing CRMs. A third feedback loop allows for generating such CRMs hypothesized through a meta-control loop equipped with a concrete analogy-making mechanism.

- **Meta-control:** Meta-control in our approach has two key functions: (1) Estimating similarities between current observations and previously learned knowledge, and (2) detecting the relevance of variables to active goals. Both of these occur on-the-fly. Similarity estimation is done through pattern matching (explained in the following requirement), while relevance processes guide the learning process by selecting variables and values for processing.

- **Pattern matching:** Pattern matching allows detecting the similarity between the agent's current observations and prior experience. When preconditions are met, pattern matching occurs between the observed data and the previously learned CRMs, and then 1) new CRMs and/or preconditions can be hypothesized, 2) predictions can be made (deduction), and 3) backward reasoning from goals (planning) can occur.

A detailed description of our proposed analogy-based meta-controller, based on the following theory of autonomous cumulative transfer learning described below, is provided in section 5.

3.1. Theory of Autonomous Cumulative Transfer Learning

From our theory of autonomous cumulative transfer learning (ACTL; Sheikhlar et al., 2020) we can infer that if a task-environment (TE) Φ consists of elements $\{\varphi_1, \ldots \varphi_n\} \in \Phi$ of various kinds, including entities E_Φ, their attributes P_Φ, and relations $\Re_\Phi$ that couple entities and attributes of Φ with each other, then

> **when a cognitive agent can reliably predict particular selected aspects $\varphi_i \in \Phi$, $i \in 1, ..., n$, using its prior knowledge, φ_i is *familiar* to the agent, and non-novel.**

More precisely, *familiarity* is the level of similarity between of the agent's current observations and its prior knowledge with respect to φ_i. Sheikhlar et al. (2020) argued for different dimensions of similarity with respect to variables, values, relations, and transitions, as building blocks of a TE. Here, we put variables into two different classes, *entities* and *attributes*, where only attributes can have numeric or symbolic values. For instance, an *object* might have a color value 'green' and a *position* value '10.' Moreover, relations are assumed to be of three types: *causal*, *ontological* and *property*. Property relations are symbolic relationships between two or more different entities. For example, in *hand holding a pen*, *holding* is a property relationship between the *hand* and the *pen* entities. Ontological relations include non-temporal symbolic relationships, e.g. [*h essence hand*] represents an ontological relationship between the variable *h* and *hand*. Lastly, in transitions, which are temporal functions, the values of attribute changes can be represented by equations relating (antecedent) causes to (future) effects.

4. Knowledge Representation & Reasoning in AERA

The Autocatalytic Endogenous Reflective Architecture (AERA) is a control system with domain-independent self-supervised cumulative learning capabilities. The current implemented version of AERA, called OpenAERA,[3] has the following knowledge representation components (Nivel et al., 2013a):

- *Entities* and *ontologies* specify to what some knowledge is applied. E.g. [h *essence hand*] states that an entity h has the *essence* of *hand*. Both *essence* and *hand* are ontologies, determining the attributes of h.

- *Drives* are goal states that an AERA-based agent desires to reach; e.g. [c *position* 15] as a drive states that c must be at position 15.

- *Causal-relational models (CRMs)* represent transformations that relate a prior state and context to a future state and context. E.g. M:[*cmd grab*(h, t_0) h *holding* $X(t_1)$] means that after applying a *grab* command at time t_0, the hand h will be holding X at time t_1. The necessary context, e.g. that a graspable object would need to correctly positioned relative to the hand, would be represented with requirement models and composite states.

- *Composite states (CSTs)* specify under what circumstances a CRM holds. E.g. CST_1:[h *position* P, c *position* P], meaning that the entities h and c are the same position P.

- *Requirement models (M_{req}s)* relate CSTs and CRMs by instantiating them. E.g. M_{req}:[*icst* $CST_1(c,h,p_0)$ *imdl*$M(c)$], where *icst* and *imdl* stand for *instantiated composite state* and *instantiated model*, respectively. M_{req} means that if an instance of a cube c and a hand h are observed to be at the same position p_0 which is a requirement for grabbing the cube, the hand h will be holding the cube c after applying the 'grab' command.

OpenAERA builds its knowledge using CRMs, CSTs, and M_{req}s to perform deductive (forward chaining), abductive (backward chaining) and inductive (learning) reasoning. Through **deduction** an AERA agent predicts the outcome of an action—predicting states from causes. **Abduction** allows the agent to guess a set of potential causes that might have led to a state. Those causes are the commands that can manipulate the attributes of entities. At the end of an abductive reasoning process, the agent commits to the most plausible set of commands that lead to reaching the goal state. **Induction** (learning) currently occurs through three different learning mechanisms, *change-targeted pattern extractor* (CTPX), *prediction-targeted pattern extractor* (PTPX), and *goal-targeted pattern extractor* (GTPX). CTPX captures the changes in the attributes of entities when a command is applied and generates a CRM, a CST, and an M_{req}. PTPX comes into play when a prediction by a CRM fails, which then generates a new CST and a new M_{req} that is called

3. See http://www.openaera.org — accessed Oct. 9, 2022.

anti-requirement model (anti-M_{req}).[4] GTPX comes into play when a goal property is unexpectedly achieved, which then generates one or more new CRMs, CSTs and M_{req}s.[5] Since learning in AERA is defeasible, every learned model has a computed success rate (which, when combined with firing rate can be used to produce a "confidence" in a meta-model), which may change as the experience accumulates.

5. Explicit Analogy in AERA

In our approach, analogy guides learning and reasoning by providing a way to create defeasible hypotheses through informed guessing, in light of a new phenomenon and unexpected experiences. It is part of AERA's methodological foundation (Thórisson, 2012; Nivel et al., 2014) that the system shall continuously predict its immediate future and hypothesize new models when these predictions fail. Because the hypothesis pertain the unknown, they can only be produced to the best of the system's ability, as informed by its existing knowledge. This means that some of them, or even most of them, will be incorrect (Thórisson, 2021a), as the system verifies them empirically through direct intervention on the environment. Because AERA represents its experience of these activities explicitly in a high-resolution ('peewee-granularity'; (cf. Nivel and Thórisson, 2008) manner, the resulting knowledge comprises dissectable information structures composed of fractional information structures whose origins and properties can be analyzed – in whole and in part – through logical argumentation.

In this approach, analogy can work via comparisons at two different levels in relation to a goal. For this we have designed two analogy mechanisms that extend OpenAERA's existing inductive reasoning, providing it with the ability to autonomously perform explicit analogies.

- The first mechanism works by identifying sameness between attributes of entities, hypothesizing new CSTs and M_{req}s, and then testing the hypothesized M_{req}s.
- The second mechanism is based on identifying the sameness between the relations of two tasks, hypothesizing new CSTs and M_{req}s, and then testing the hypothesized M_{req}s.

5.1. Learning based on identical attributes

Knowledge transfer can occur between tasks having identical attributes. The attributes belong to entities, which are building components of a task-environment. Here, we use an example to show how the first analogy mechanism (learning via identical attributes) allows

4. In Replicode logic (Nivel et al., 2012), the programming language for AERA's knowledge representation, 'anti-' essentially means 'absence of,' and represents lack of knowledge.
5. The triad of CTPX, PTPX and GTPX fit the following three logical possibilities: An issued command unexpectedly explains the change of a value (CTPX), a model unexpectedly predicts the wrong value (PTPX), and a command unexpectedly results in a goal value being achieved (GTPX). These are detailed elsewhere (Nivel and Thórisson (2013); Nivel et al. (2012)) and can be found in the OpenAERA code (http://www.openaera.org).

the creation of new M_{req}s (and CSTs) and enables the AERA agent to detect relevant attributes after which it removes the irrelevant ones from its knowledge base.

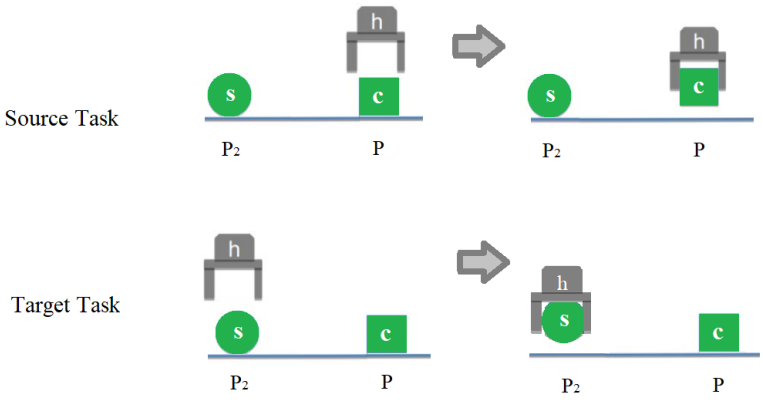

Figure 1: In the source task, AERA agent learns that by applying the *grab* command, hand h will be holding cube c. The shared attribute (green color) between sphere s and cube c allows AERA agent to hypothesize that the hand can also grab and then hold the sphere in the target task. In other words, it generates M_{req2}, which states that grab model M_{grab} also holds for the sphere.

Example 1: We assume there exist entities c, s, and h in the environment, each of which have attributes including *essence*, *position* and *color*. An illustration of this example is shown in Figure (1).

Initial conditions of the source task: Let us say we have
CST_c:[c *essence cube*, c *position* P, c *color green*],
CST_s:[s *essence sphere*, s *position* P_2, s *color green*],
CST_h:[h *essence hand*, h *position* P],
[h *Holding* [ ]]
as the initial state of the environment. For example, [h *Holding* [ ]] indicates that the hand is empty (the hand h is Holding nothing), and CST_c represents the current state of the entity c which has the *essence* of cube, *position* P, and *color* green. Note that the value of the *color* attribute is the same (green) for both the cube c and the sphere s.

Source Task: After applying a *grab* command (which might occur over learning through motor babbling) by the hand, the AERA agent will observe that the hand is holding the cube c, [h *Holding* c], in the next time frame. Since the agent sees a change in the state of the hand from *Holding* nothing to *Holding* c, CTPX generates the following CST, CRM, and M_{req}
CST_1:[CST_c, CST_h, h *Holding* [ ]],
M_{grab}:[*cmd grab*(h, t_0) h *Holding* $X(t_1)$],
M_{req1}:[*icst* CST_1 *imdl* M_{grab}],
which means that when the hand h and the cube c are at the same position P and the hand h is empty, the hand h will be holding the cube c after grabbing.

Target Task: After learning the above knowledge, assume that the target task is assigned to the agent, where the goal is to achieve the [h *Holding* s], that is, the hand h will have to be holding the sphere s. The initial conditions are that the hand's position is $P2$

(the sphere s is at P_2 as well) and the hand h is empty, [h *Holding* []]. Here is where the analogy-making mechanism comes into play. Since the cube c and the sphere s have an identical attribute value (green color), the following CST and M_{req} are hypothesized.

CST_2:[CST_s, CST'_h, h *Holding* [ ]],

M_{req2}:[*icst* CST_2 *imdl* M_{grab}],

where

CST'_h:[h *essence hand*, h *position* P_2].

The hypothesized M_{req2} states that M_{grab} also holds for the sphere s. In other words, the AERA agent hypothesizes that the hand h can use M_{grab} for both entities, the cube c and the sphere s, since both have an identical color. Since M_{req2} has not been tested in practice, it has a low confidence value (success rate) such that if it failed, it would be removed from the knowledge base immediately.

This mechanism also allows detecting the **relevance** of attributes via experience as follows:

- If M_{req2} succeeds in practice, the following M_{req} and CST are created
  M_{req3}:[*icst* CST_3 *imdl* M_{grab}],
  CST_3:[s *position* P, s *color green*, h *essence hand*, h *position* P],
  where CST_3 is the modified version of CST_2, in which the *essence* attribute has been removed, since the essence of the sphere s was proven to be irrelevant. Thus, the essence of objects should not be among the preconditions of the grab CRM.

- If M_{req2} fails in practice, it will be removed from the knowledge base. Then, the PTPX creates an anti-requirement model, $anti-M_{req}$: [*icst* CST_2 $\neg$ *imdl* M_{grab}]. The anti-M_{req} states that the hand h cannot use M_{grab} to grab the sphere s. Therefore, it finds the color property irrelevant and hypothesizes CST_4 and M_{req4}: [*icst* CST_4 *imdl* M_{grab}]. CST_4 is the modified version of CST_2, where the color property has been removed.

5.2. Learning based on identical relations

Knowledge transfer can also occur between tasks having identical relations. Here, we use another example to show how the second analogy mechanism (learning based on identical relations) allows the creation of new M_{req}s (and CSTs).

Example 2: Assume there exist three entities (c, s and h) in the environment, each of which has attributes including essence, position and color. Unlike Example 1, here, the entities c and s have different colors. An illustration of this example is shown in Figure (2).

Source Task: Assume that the source task was to achieve the goal [c *position* $P2$] by grabbing and moving the cube c with the hand h, where the blue items shown in Figure (3) were learned. So, we have

CST_1:[CST_c, CST_h],

CST_2:[CST_s, CST_h],

CST_3:[CST_c, h *Holding* c],

M_{move}:[*cmd* $move(h, DeltaP, P_0, t0)$ h *position* $P_1(t_1)$],

$M_{hMovesX}$:[*imdl* M_{move} c *position* Pc],

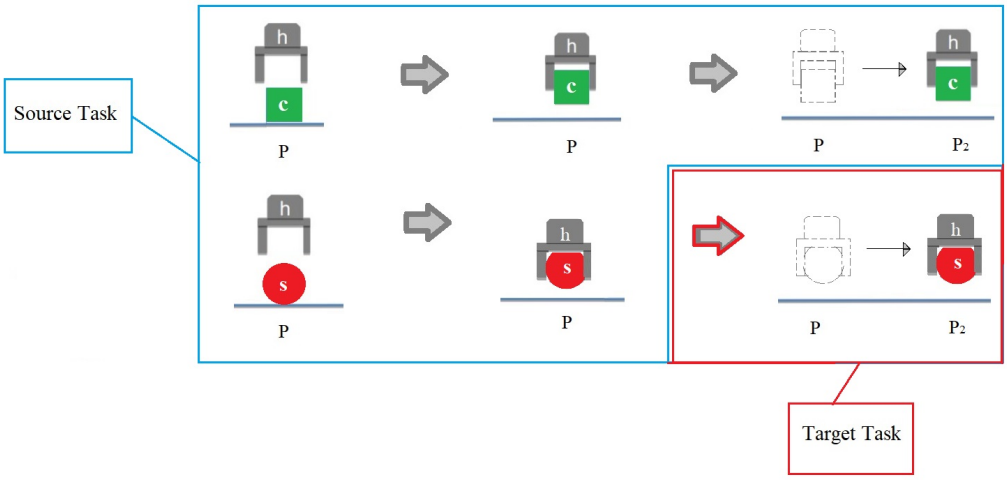

Figure 2: In the source task, AERA agent learns that the hand can grab the cube and then move it to another location when it is holding the cube. It also learns that it can grab a sphere. In the target task, it hypothesizes that the hand can also move the sphere when it is holding it. In other words, it generates $M_{req}^{CST_4}$, which states that move model $M_{hMovesX}$ also holds for the sphere.

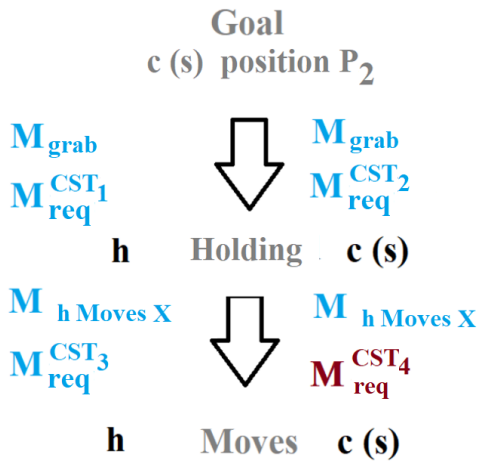

Figure 3: The sameness of relations between the task of moving objects c and s to position $P2$. See text for details.

$M_{req}^{CST_3}$:$[icst\ CST_3 \quad imdl\ M_{hMovesX}]$,
where
$CST_c = [c\ essence\ cube,\ c\ position\ P,\ c\ color\ green]$,
$CST_s = [s\ essence\ sphere,\ s\ position\ P,\ s\ color\ red]$,
$CST_h = [h\ essence\ hand,\ h\ position\ P]$.
$M_{hMovesX}$ states that after applying a *move* command to the hand h, the position of the object X that is held by the hand will be something different. Its preconditions, i.e. *CST3* and $M_{req}^{CST_3}$ state that $M_{hMovesX}$ only holds for the cube c.

Target Task: After learning the above knowledge, assume that the target task is to achieve the goal [*s position P2*], with [*h Holding s*] being the task's initial condition. Since all the causal and property relations are the same between the tasks of (grabbing and) moving *c* and *s*, the following CST and M_{req} are hypothesized once the goal property [*s position P2*] is given to the AERA agent

CST_4: [CST_h, *h Holding s*],

$M_{req}^{CST_4}$:[*icst* CST_4 *imdl* $M_{hMovesX}$].

The hypothesized $M_{req}^{CST_4}$ states that the AERA agent can use $M_{hMovesX}$ to move the sphere *s* with the hand *h* as well. Note that $M_{req}^{CST_4}$ has a low confidence value (success rate) such that if it failed, it would be removed immediately.

6. Conclusions

We have presented two proposals for concrete analogy-making that meet the requirements of general self-supervised learning. The mechanisms, in short, rest on the premise that entities having various attributes build task-environments and can have causal or property relationships. If an AERA agent captures identical attributes and/or relationships between two different task entities in a particular task-environment, it can transfer its knowledge autonomously in relation to a given goal, even one that it has not done before.

The approach has been outlined in the OpenAERA framework; the two analogy algorithms are currently being implemented within the current inductive reasoning and learning mechanisms in OpenAERA.[6] In future work we plan to further formalize the mechanisms put forth here.

Acknowledgments

This work was supported in part by the Department of Computer Science at Reykjavik University, the Icelandic Institute for Intelligent Machines, and a grant from Cisco Systems Inc. We would like to thank Leonard Eberding for his useful comments on the paper.

References

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

Haitham B. Ammar, Karl Tuyls, Matthew E. Taylor, Kurt Driessens, and Gerhard Weiss. Reinforcement learning transfer via sparse coding. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 383–390. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

Haitham Bou Ammar, Eric Eaton, Matthew E Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. An automated measure of mdp similarity for transfer in reinforcement learning. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

6. http://www.openaera.org – accessed Oct. 9, 2022.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

Shi Dong, Ping Wang, and Khushnood Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021.

Gary L. Drescher. *Made-up minds: a constructivist approach to artificial intelligence*. PhD thesis, Massachusetts Institute of Technology, 1989.

Leonard M. Eberding, Matteo Belenchia, Arash Sheikhlar, and Kristinn R. Thórisson. About the intricacy of tasks. In *International Conference on Artificial General Intelligence*, pages 65–74. Springer, 2021.

Leonard M. Eberding, Arash Sheikhlar, and Kristinn R. Thórisson. Comparison of machine learners on an ABA experiment format of the cart-pole task. *Proceedings of Machine Learning Research, International Workshop on Self-Supervised Learning*, 159:49–63, 2022.

Brian Falkenhainer, Kenneth D Forbus, and Dedre Gentner. The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1):1–63, 1989.

Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170, 1983.

Richard N. Henriksen. *Scale Invariance: Self-Similarity of the Physical World*. Wiley, 2015.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1567–1574, 2021.

Shan Lu, Soyeon Park, Eunsoo Seo, and Yuanyuan Zhou. Learning from mistakes: a comprehensive study on real world concurrency bug characteristics. In *ACM Sigplan Notices*, volume 43, pages 329–339. ACM, 2008.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

E. Nivel, K. R. Thórisson, B. R. Steunebrink, H. Dindo, G. Pezzulo, M. Rodriguez, C. Hernandez, D. Ognibene, J. Schmidhuber, R. Sanz, Helgi Páll Helgason, A. Chella, and G. K. Jonsson. Bounded Recursive Self-Improvement. Technical RUTR-SCS13006, Reykjavik University Department of Computer Science, Reykjavik, Iceland, 2013a.

Eric Nivel and Kristinn R. Thórisson. Self-Programming: Operationalizing Autonomy. In *Proceedings of the 2nd Conf. on Artificial General Intelligence*, 2008.

Eric Nivel and Kristinn R. Thórisson. Replicode: A constructivist programming paradigm and language. Technical RUTR-SCS13001, Reykjavik University School of Computer Science, 2013.

Eric Nivel, Nathaniel Thurston, and Yngvi Bjornsson. Replicode Language Specification. Technical report, Technical report available at http://wiki. humanobs. org/_media/public: publications: projdocs: d3-1-specification-replicode. v1. 0. pdf, 2012.

Eric Nivel, Kristinn R. Thórisson, H. Dindo, G. Pezzulo, M. Rodriguez, C. Corbato, B. Steunebrink, D. Ognibene, A. Chella, Jürgen Schmidhuber, Ricardo Sanz, and Helgi Páll Helgason. Autocatalytic Endogenous Reflective Architecture. Technical RUTR-SCS13002, Reykjavik University School of Computer Science, Reykjavik, Iceland, 2013b.

Eric Nivel, Kristinn R. Thórisson, Bas R. Steunebrink, Haris Dindo, Giovanni Pezzulo, Manuel Rodríguez, Carlos Hernández, Dimitri Ognibene, Jürgen Schmidhuber, Ricardo Sanz, and others. Bounded Seed-AGI. In *Artificial General Intelligence*, pages 85–96. Springer, 2014.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009. ISBN 052189560X, 9780521895606.

Judea Pearl. Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution. *arXiv:1801.04016 [cs, stat]*, January 2018. arXiv: 1801.04016.

Jean Piaget. *The Psychology of Intelligence*. Routledge and Kegan Paul, London, England, 1950.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1): 1309–1342, 2018.

Arash Sheikhlar, Kristinn R. Thórisson, and Leonard M. Eberding. Autonomous cumulative transfer learning. In *International Conference on Artificial General Intelligence*, pages 306–316. Springer, 2020.

Arash Sheikhlar, Leonard M Eberding, and Kristinn R. Thórisson. Causal generalization in autonomous learning controllers. In *International Conference on Artificial General Intelligence*, pages 228–238. Springer, 2021.

Jonathan Sorg and Satinder Singh. Transfer via soft homomorphisms. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 741–748, 2009.

Bas R. Steunebrink, Kristinn R. Thórisson, and Jürgen Schmidhuber. Growing recursive self-improvers. In *Proceedings of Artificial General Intelligence*, pages 129–139, 2016.

Kristinn R. Thórisson. A New Constructivist AI: From Manual Methods to Self-Constructive Systems. In Pei Wang and Ben Goertzel, editors, *Theoretical Foundations of Artificial General Intelligence*, volume 4 of *Atlantis Thinking Machines*, pages 145–171. Atlantis Press, Amsterdam, The Netherlands, 2012.

Kristinn R. Thórisson. Seed-programmed autonomous general learning. In *Proceedings of Machine Learning Research*, volume 131, pages 32–70, 2021a.

Kristinn R. Thórisson. The 'explanation hypothesis' in autonomous general learning. In *Proceedings of Machine Learning Research*, volume 159, pages 5–27. Springer, 2021b.

Kristinn R. Thórisson and Arthur Talbot. Cumulative learning with causal-relational models. In *Artificial General Intelligence*, pages 227–237, Cham, 2018. Springer International Publishing.

Kristinn R. Thórisson, David Kremelberg, Bas R. Steunebrink, and Eric Nivel. About understanding. In *Proceedings of AGI-16*, pages 106–117, New York, NY, USA, 2016. Springer-Verlag.

Kristinn R. Thórisson, Jordi Bieger, Xiang Li, and Pei Wang. Cumulative learning. In *International Conference on Artificial General Intelligence*, pages 198–208. Springer, 2019.

Hao Wang, Shaokang Dong, and Ling Shao. Measuring structural similarities in finite mdps. In *IJCAI*, pages 3684–3690, 2019.

Ke Wang and Zhendong Su. Automatic generation of raven's progressive matrices. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

Shuxi Zeng, Murat Ali Bayir, Joseph J. Pfeiffer III, Denis Charles, and Emre Kiciman. Causal transfer random forest: Combining logged data and randomized experiments for robust prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 211–219, 2021.

Kecheng Zheng, Zheng-Jun Zha, and Wei Wei. Abstract reasoning with distracting features. *Advances in Neural Information Processing Systems*, 32, 2019.