ACS Poster Collection (2018) 1–15 Published 8/2018; Paper X

MacGyver Problems: AI Challenges for Testing Resourcefulness and Creativity

Vasanth Sarathy VASANTH.SARATHY@TUFTS.EDU
Matthias Scheutz MATTHIAS.SCHEUTZ@TUFTS.EDU
Department of Computer Science, Tufts University, Medford, MA 02155 USA

Abstract

When faced with real-world problems that seem unsolvable, humans display an exceptional degree of flexibility and creativity, improvising solutions with the limited resources available. In this essay, we propose a class of domain-independent AI challenge tasks - MacGyver Problems - that target these capabilities. We present a formal framework for generating these problems and outline a set of independent subtasks that will let the research community make progress. We also consider, informally, ways in which researchers can measure progress and evaluate agents under the proposed framework.

1. Introduction

How should we evaluate machine intelligence? This a long-standing problem in AI and robotics. From Alan Turing's original question about whether machines can think (Turing, 1950) to today's plethora of robotics and AI challenges (Levesque et al., 2012; Feigenbaum, 2003; Boden, 2010; Bringsjord & Sen, 2016; Cohen, 2005; Harnad, 1991; Riedl, 2014) and data sets (Johnson et al., 2017; Weston et al., 2015) , the question of what makes a suitable test is still open, relevant, and crucial to judging progress in AI while guiding its research and development.

The crux of this question is a choice about what we should measure. The Turing Test focused on natural language interactions; its progenies have since expanded to include vision, planning, game playing, localization and mapping and many others. Research in various AI subfields is often guided by these sorts of targeted data sets and challenges. Since research questions within AI subfields are quite rich and complex, we might argue there is no need to pick one behavior, but rather pursue all of them, separately and in parallel. However, psychological and zoological studies in human and animal intelligence suggest the existence of general capabilities that transcend these types of targeted abilities (Ackerman, 2016).

We introduce the idea that general intelligence is encapsulated in notions of resourcefulness, improvisation, and creative problem solving that we humans use everyday and that we celebrate in movies like "The Martian" and television shows like "MacGyver." We thus ask a different question: *can machines improvise when they become stuck on a problem?*

As the first step towards formalizing intuitions of creative problem solving and improvisation, we define a new class of - *MacGyver problems* - using the language of classical planning. The

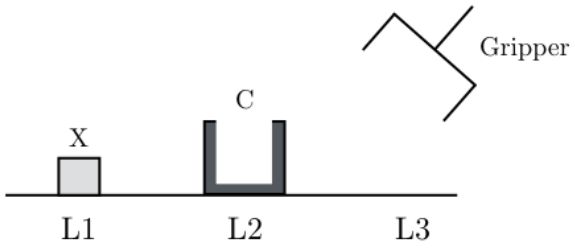

Figure 1. Cup world: contains a block X and a cup C. It is easy to see how the block can be moved from location $L1$ to $L3$. But, what if the gripper is not allowed to touch the block? Details in Section 4.

idea here is to let an agent begin with a seemingly unsolvable problem and observe how it fares in attempting to find a solution. If the agent can take actions and incorporate relevant knowledge from the environment to make the problem more tractable (or at least computable in some cases), then we might be able to predict the agent's capabilities in solving novel problems in the open world. We present these ideas in the context of an illustrative blocks world variant called *cup world* (shown in Figure 1), where we define an initial instance that is unsolvable. The agent must then transform and modify its initial domain representations to learn and incorporate new concepts and actions that let it solve the problem.

In the following pages, we define the MacGyver framework more formally (Section 3) and provide some complexity results. In Section 4, we return to the cup world domain and outline how a MacGyver problem in cup world could be solved, while noting the underlying challenges in solving these problems, more generally. We then discuss, in Section 5, the space of capabilities and component subtasks that an agent might need to solve MacGyver problems. In Section 6, we discuss how we might evaluate these agents and how the research community can evaluate its progress. Finally, Section 7 concludes and discusses avenues for future research. But first we briefly review the history of machine intelligence tests and highlight some of their missing aspects.

2. The Turing Test and its Progeny

Alan Turing (1950) asked whether machines could produce observable behavior (e.g., natural language) that we would say required thought in people. He suggested that if interrogators were unable to tell, after having long free-flowing conversations with a machine whether they were dealing with a machine or a person, then they could conclude that the machine was "thinking". Turing did not actually intend for this to be a test, but rather a prediction of what could be achieved, technologically, in fifty years (Cooper & Van Leeuwen, 2013). Nevertheless, others have since developed tests for machine intelligence that were variations of the so-called Turing Test to address a common criticism that it was easy to deceive the interrogator.

For example, Levesque et al. (2012) designed a reading comprehension test, entitled the *Winograd Schema Challenge*, in which the agent is presented a question having some ambiguity in the referent of a pronoun or possessive adjective. The question asks the reader to determine the referent of this ambiguous pronoun or possessive adjective by selecting one of two choices. Feigenbaum

(2003) proposed a variation of the Turing Test in which a machine can be evaluated by a team of subject matter specialists through natural language conversation. Other proposed tests have attempted to study a machine's ability to produce creative artifacts and solve novel problems (Boden, 2010; Bringsjord et al., 2001; Bringsjord & Sen, 2016; Riedl, 2014; Wiggins, 2006).

Extending capabilities beyond the linguistic and creative ones, Harnad (1991) suggested a *Total Turing Test* (T3) that expanded the range of capabilities to a full set of robotic capacities found in embodied systems. Schweizer (2012) extended the T3 to incorporate species evolution and development over time, proposing the *Truly Total Turing Test* (T4) to test not only individual cognitive systems but whether the candidate cognitive architecture is capable, as a species, of long-term evolutionary achievement.

Finding that the Turing Test and its variants were not helping guide research and development, others proposed a task-based approach. Specific task-based goals were couched as toy problems that were representative of a real-world task (Cohen, 2005). The research communities benefited greatly from this approach and focused their efforts towards specific machine capabilities like object recognition, automatic scheduling and planning, scene understanding, localization and mapping, and even game playing. Many competitions and challenges emerged that tested the machine's performance in applying these capabilities. Some competitions even tested embodiment and robotic capacities that combined multiple tasks. For example, the DARPA *Robotics Challenge* evaluated a robot's ability to conduct remote operations including turning valves, using a tool to break through a concrete panel, opening doors, and remove debris from entryways.

Unfortunately, neither the Turing Test variants nor the task-based challenges captured the intuitive notions of resourcefulness that is at the core of creative problem solving. Creative agents situated in the real world must be able to solve problems with limited resources. This means they must be able to improvise, challenge prior assumptions, generate and test ideas, make new observations, and acquire new knowledge from their environments.

3. The MacGyver Framework

Our core proposal is to present an agent with a problem that is unsolvable with its initial knowledge and observe its problem-solving processes to estimate the degree to which it is creative. If the agent can think outside of its current context, take exploratory actions, incorporate relevant environmental cues, and learn knowledge to make the problem tractable (or at least computable), then it has the general ability to solve open-world problems.

This notion of expanding one's current context builds on Boden's (2010) seminal work on creativity and, specifically, her distinction between *exploratory* and *transformational* creativity. Boden introduced the notion of a conceptual space and proposed that exploratory creativity involves discovering new objects in this space. Transformational creativity involves redefining that conceptual space and producing a paradigm shift. There have been a few attempts to formalize and conceptualize these ideas, including Wiggins (2006) Creative Systems Framework. He formalized Boden's notion of a conceptual space $\mathscr{C}$ that is a subset of a universe $\mathscr{U}$, and that in turn contains every possible concept. A conceptual space, according to Wiggins, is defined by a set of rules $\mathscr{R}$ for constraining the space and a set of rules $\mathscr{T}$ for traversing it. Exploratory creativity occurs when the rule

sets are used within a concept space to discover new concepts. Transformational creativity occurs when the conceptual space is redefined by modifying rule sets. Using this abstract formalism, he began the task of characterizing the behavior of creative systems and, importantly, distinguishing between object-level search within a conceptual space and meta-level searches of conceptual spaces.

However, more work was still needed to operationalize these concepts, which, while more formal than Boden's original proposal, lacked depth and connections to work in AI and robotics formalisms. More recently, Colin et al. (2016) built on Wiggins' basic ideas and proposed that the dual searches - exploratory and transformational - can be set up as a hierarchical reinforcement learning problem formalized in terms of Markov decision processes. While this is a promising approach to formalizing these dual searches, we believe that it is just one piece of a larger puzzle. In response, we propose an approach that unifies several AI research traditions including not only reinforcement learning but also planning, belief revision, and others. We can now start to formalize the MacGyver framework.

3.1 Formal Definition of a MacGyver Problem

We define $\mathcal{L}$ to be a first-order language with atoms $p(t_1, \ldots, t_n)$ and their negations $\neg p(t_1, \ldots, t_n)$, where t_i represents terms that can be variables or constants. An atom is grounded if and only if all of its terms are constants. A planning domain in $\mathcal{L}$ can be represented as $\Sigma = (S, A, \gamma)$, where S denotes the set of states, A the set of actions, and γ the transition functions. A planning problem $\mathcal{P} = (\Sigma, s_0, g)$ consists of the domain, the initial state s_0, and the goal state g. A plan π is any sequence of actions. A plan π is a solution to the planning problem if $g \subseteq \gamma(s_0, \pi)$. We also consider the notion of state reachability and the set of all successor states $\hat{\Gamma}(s)$, which defines the set of states reachable from s.

To formalize a MacGyver problem, we define a universe and a world within this universe. The world describes the full set of abilities of an agent and includes those abilities that the agent knows about and those of which it is unaware. We can then define an agent subdomain as representing a proper subset of the world that is within the agent's awareness. A MacGyver problem then becomes a planning problem that is defined in the world, but outside the agent's current subdomain, as depicted in Figure 2.

Definition 1 (Universe). *A universe $\mathbb{U} = (S, A, \gamma)$ as a classical planning domain that represents all aspects of the physical world perceivable and actionable by any and all agents, regardless of capabilities. This includes all allowable states, actions, and transitions in the physical universe.*

Definition 2 (World). *A world $\mathbb{W}^t = (S^t, A^t, \gamma^t)$ as a portion of the Universe $\mathbb{U}$ corresponding to those aspects that are perceivable and actionable by a particular species t of agent. Each agent species $t \in T$ has a particular set of sensors and actuators allowing agents in that species to perceive a proper subset of states, actions or transition functions. Thus, a world can be defined as*

$$\mathbb{W}^t = \{(S^t, A^t, \gamma^t) \mid ((S^t \subseteq S) \vee (A^t \subseteq A) \vee (\gamma^t \subseteq \gamma)) \wedge \neg((S^t = S) \wedge (A^t = A) \wedge (\gamma^t = \gamma))\}.$$

Definition 3 (Agent Subdomain). *An agent subdomain $\Sigma_i^t = (S_i^t, A_i^t, \gamma_i^t)$ of type t is a planning subdomain that corresponds to the agent's perception and action within its world $\mathbb{W}^t$. In other*

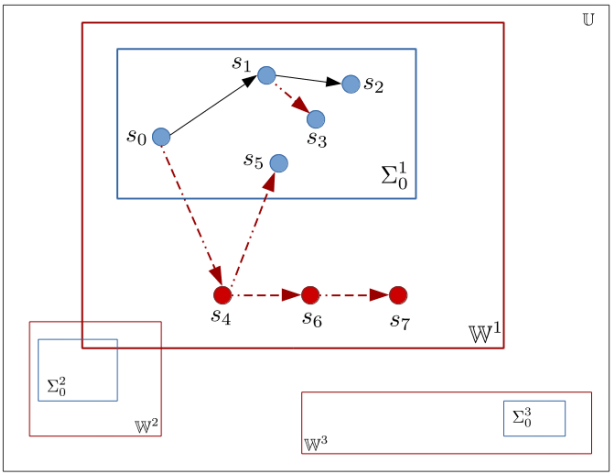

Figure 2. Conceptual diagram showing several exemplary MacGyver problems and how they relate to classical planning tasks. Three agents ($t = 1, 2, 3$) are depicted along with their starting domains (Σ_0^t) and their worlds ($\mathbb{W}^t$). Operators and states not in the agent's initial domain are shown in red. Thus, states s_3, s_4, s_5, s_6, s_7 are unreachable from s_0 in domain Σ_0^1, and consequently could be characterized as goal states of MacGyver problems, albeit of different difficulties.

words, the agent is not fully aware of all of its capabilities at all times, and the agent domain Σ_i^t corresponds to the portion of the world that it is perceiving and acting at time i.

$$\Sigma_i^t = \{(S_i^t, A_i^t, \gamma_i^t) \mid ((S_i^t \subset S^t) \vee (A_i^t \subset A^t) \vee (\gamma_i^t \subset \gamma^t)) \wedge \neg((S_i^t = S^t) \wedge (A_i^t = A^t) \wedge (\gamma_i^t = \gamma^t))\}$$

Definition 4 (MacGyver Problem). *A MacGyver problem with respect to an agent t is a planning problem in the agent's world $\mathbb{W}_t$ that has a goal state g that is currently unreachable by the agent. Formally, a MacGyver problem $\mathcal{P}_M = (\mathbb{W}^t, s_0, g)$, where*

- *$s_0 \in S_i^t$ is the initial state of the agent*
- *g is a set of ground atoms*
- *$S_g = \{s \in S | g \subseteq s\}$*

Where $g \subseteq s', \forall s' \in \hat{\Gamma}_{\mathbb{W}^t}(s_0) \setminus \hat{\Gamma}_{\Sigma_i^t}(s_0)$.

3.2 Complexity Results and Importance of Heuristics

It naturally follows that, in the context of a world $\mathbb{W}_t$, the MacGyver problem $\mathcal{P}_M$ is a classical planning task which from the agent's current perspective is unsolvable. We can reformulate the MacGyver problem in terms of language recognition to analyze its complexity. This will clarify the difficulty of the problem, and, importantly, establish the key role that heuristics play when solving MacGyver problems.

Definition 5 (MGP-EXISTENCE). *Given a set of statements D of planning problems, let* MGP-EXISTENCE(D) *be the set of all statements $P \in D$ such that P represents a MacGyver problem $\mathcal{P}_M$, without any syntactical restrictions.*

Theorem 1 *MGP-EXISTENCE is decidable.*

Theorem 2 *MGP-EXISTENCE is EXPSPACE-complete.*

Theorems 1 and 2 show that the question of knowing whether a given problem is a MacGyver problem, although computable (in the finite case), is still intractable.[1] Thus, the agent will necessarily need heuristics to explore its own search space. We argue that the role of heuristics in solving these problems is paramount. Indeed, there is no single heuristic in any sense of the word (Langley, 2017) that can guide an agent in all MacGyver problems. Even for a given MacGyver task, it is unlikely that a single heuristic will be sufficient. This means that the agent must use and reason with a family of heuristics as it searches for a solution. This is good news for cognitive systems research that aims to leverage the power of heuristics in problem solving and align it with parallel results in humans, and, conversely, uncover heuristics that people use when solving these problems to better inform agent design. In fact, we might flip how we think about heuristics for MacGyver problems: from a weak method that either produces a suboptimal solution or does not guarantee a solution at all, to a necessary step towards achieving representational change or domain transformation, thereby eliciting satisfactory solutions.

3.3 MacGyver Solution via Domain Modification

By definition, MacGyver problems require that the initial domain be transformed in some way (e.g., by adding a state, transition function, or action) for the goal state to be reachable. Here, we provide some formal specifications for what a solution strategy might look like for a MacGyver agent, which must keep modifying its domain representation until it can reach the goal.

Definition 6 (Agent Domain Modification). *A domain modification Σ_j^{t*} involves either a domain extension or contraction (for brevity, we only consider extensions here). A domain extension Σ_j^{t+} of an agent is an Agent-subdomain at time j that is in the agent's world $\mathbb{W}^t$ but not in the agent's subdomain Σ_i^t in the previous time i, such that $\Sigma_i^t \preceq \Sigma_j^t$. The agent extends its subdomain through sensing and perceiving its environment and its own self. E.g., the agent can extend its domain by making an observation, receiving advice or an instruction or performing introspection. Formally,*

$$\Sigma_j^{t+} \;=\; \{(S_j^{t+}, A_j^{t+}, \gamma_j^{t+}) \;\mid\; (S_j^{t+} \;\subset\; S^t \setminus S_i^t) \vee (A_j^{t+} \;\subset\; A^t \setminus A_i^t) \vee (\gamma_j^{t+} \;\subset\; \gamma^t \setminus \gamma_i^t)\}.$$

The agent subdomain that results from a domain extension is $\Sigma_j^t = \Sigma_i^t \cup \Sigma_j^{t+}$. A domain modification set $\Delta_{\Sigma_i^t} = \{\Sigma_1^{t*}, \Sigma_2^{t*}, \ldots, \Sigma_n^{t*}\}$ is a set of n domain modifications on subdomain Σ_i^t. Let Σ_Δ^t be the subdomain resulting from applying $\Delta_{\Sigma_i^t}$ on Σ_i^t.

1. The proofs for these theorems are straightforward applications of complexity-theoretic techniques and have been omitted here for brevity.

Definition 7 (Strategy and Domain-Modifying Strategy). *A* strategy *is a tuple* $\omega = (\pi, \Delta)$ *of a plan* π *and a set* Δ *of domain modifications. A domain-modifying strategy* ω^C *involves at least one domain modification, i.e.,* $\Delta \neq \emptyset$.

Definition 8 (Context). *A* context *is a tuple* $\mathbb{C}_i = (\Sigma_i, s_i)$ *that represents the agent's subdomain and state at time* i.

We are now ready to define an insightful strategy as a set of actions and domain modifications that the agent must perform for the problem's goal state to be reachable.

Definition 9 (Insightful Strategy). *Let* $\mathbb{C}_i = (\Sigma_i^t, s_0)$ *be the agent's current context. Let* $\mathcal{P}_M = (\mathbb{W}^t, s_0, g)$ *be a MacGyver problem for the agent in this context. An insightful strategy is a domain-modifying strategy* $\omega^I = (\pi^I, \Delta^I)$ *that, when applied in* $\mathbb{C}_i$*, results in a context* $\mathbb{C}_j = (\Sigma_j^t, s_j)$*, where* $\Sigma_j^t = \Sigma_{\Delta^I}^t$ *such that* $g \subseteq s', \forall s' \in \hat{\Gamma}_{\Sigma_j^t}(s_j)$.

Formalizing the insightful strategy in this way is analogous to the moment of insight reached when a problem becomes tractable (or in our definition computable) or when solution plan becomes feasible. Specifically, solving a problem involves creative exploration, domain extensions, and domain contractions until the agent has the information it needs within its subdomain to solve it as a classical planning task, and it does not need any further domain extensions.

Note, however, the definitions stated here do not require a multitude of domain transformations. The agent might achieve a flash of insight about the appropriate representation in a single transformation. Alternatively, the agent may work through several minor transformations. So the definitions here are meant to capture creative problem solving more generally, encompassing both single-step and multi-step insights.

Moreover, the definition of an insightful strategy does not require using a single strategy for all MacGyver problems, as no such universal strategy exists. Instead, definition 9 merely serves as a formalism for capturing notions of context and representational change that apply to any solution. That is, the definition is not tied to a particular algorithm or solution plan, and accordingly does not speak to the quality of the solution reached. An insightful strategy is just meant to move the agent to a state from which the goal state is reachable. Achieving the goal state might still be intractable, as is any classical planning problem.

3.4 Connections to Insight Problem Solving in Humans

The MacGyver problem formulation proposed in this essay draws substantial inspiration from decades-old line of psychological research on insight problem solving. There are numerous theories of insight and its links to creativity, one of the earliest being Wallas' (1926) four-stage framework of preparation, incubation, illumination, and verification. Many subsequent psychological theories have addressed what solvers were doing during each of these stages. What these theories shared was that insight problem solvers encountered an impasse and later had an insight that let them solve the task (Ohlsson, 1992). In this essay, we consider the problem-space approach to insight problem solving, in which impasses occur due to an incorrectly chosen problem representation, such as

wrong propositions or operators) (Kaplan & Simon, 1990; Ohlsson, 1992). Impasses can be broken by revising the problem representation, which in turn can be viewed as a meta-level search over the space of representations. An alternative approach to thinking about insight is to consider it not as a search through a problem space and then a meta-level space, but as a memory retrieval task of indexing, retrieving, and applying an analogous knowledge structure (Langley & Jones, 1988).

Notwithstanding their differences, all the theories agree about the importance of cues and heuristics in the search (or as appropriate, analogical retrieval) process. These cues and heuristics help the solver consider previously ignored knowledge structures, propositions, relations, and operators, as well as consider new ones. The heuristics themselves may be strategies for navigating through or modifying the knowledge structures (e.g., problem representations) that contain the problem or solution. In addition, some heuristic strategies might actively or passively participate with the surrounding environment and observe what happens (Sarathy, 2018). The purpose of these cues and heuristics is to arrive at a representation that allows a full or partial insight. We preview a smattering of such heuristics in the next section, where we provide a blueprint for solving a MacGyver problem in the context of the cup world domain introduced in Section 1. Overall, we believe the connections between the MacGyver problem solving and insight problem solving in humans to be deeply interesting and worthy of further research.

4. A Conceptual Blueprint for Solving Cup World

We are now ready to operationalize the MacGyver framework in the cup world domain, outline the domain transformations needed to solve this problem, and discuss some domain-modifying strategies. This will highlight the sorts of computational, perceptual, and action capabilities needed to execute the appropriate strategy, assimilate the necessary knowledge, and make the needed domain transformations.

The cup world domain consists of a block and cup on a table, as shown in Figure 1. Consider a start domain Σ_0^t for an agent t with a gripper. Let us say the agent has the ability to perceive and reason about relational aspects of its environment (e.g., whether an object is visible, whether it has been touched, and spatial relationships between objects). The domain is discrete and finite, with actions being represented symbolically. Let us also assume that the agent is a pick-and-place robot with a single operator for picking and placing objects; this $pick\text{-}and\text{-}place$ action has an associated script composed of several sub-actions ($reach\text{-}for$, $grasp$, $lift$, $move\text{-}over$, $set\text{-}down$, $release$), that form the set of primitive actions.

The agent is primarily tasked with picking and placing objects, and accordingly has chunked subactions into a single $pick\text{-}and\text{-}place$ macro-operator that increases planning efficiency. That said, the primitive actions can be directly coupled to robotic actuators with precise continuous values. For example, the $lift$ action can be parametrized to 3D point clouds that represent objects in space and real-valued vertical displacements. Later in this section, we comment on extensions to infinite and continuous space in which new primitive actions can be generated from parameterizations of actuator controls. Thus, we assume here that the robot can process 3D point cloud information from the environment and determine whether an object is a cup or a block, and whether it is upright.

In Σ_0^t, we can solve traditional planning problems such as moving the block x from l_1 to l_3, with the plan $\pi = \{pick\text{-}and\text{-}place(x, l_1, l_3)\}$. We can define a MacGyver problem, $\mathcal{P}_M$, for this agent t given a start domain Σ_0^t by requiring that the block x be moved to l_3 **without touching it**, that is $touched(x) = False$. To handle this problem, the agent must extend its start domain by using heuristics and cues. Although there may be many different ways to solve this MacGyver problem, let us consider one particular approach and analyze more closely what sorts of heuristics and cues might be activated by an agent conforming to it. Consider a set of two domain extensions $\Delta_{\Sigma_0^t} = \{\Sigma_1^{t+}, \Sigma_2^{t+}\}$. The first extension Σ_1^{t+} includes a $nudge$ action and the second domain extension Σ_2^{t+} includes several new actions and the fluent $enclosed$.

We can decompose the $pick\text{-}and\text{-}place$ action into its primitive actions (*Heuristic 1:* decompose chunks) and then attempt to replan with these primitive operators. However, this will fail, as any action that requires grasping the block will trigger the $touched$ event. Thus, the agent may instead attempt to solve a simpler problem of moving the block without picking it up (*Heuristic 2:* reformulate goals and define simpler goals). Langley et al. (2016) have reported an architecture for hierarchical problem solving that searches through a space of problem decompositions, which might be a fruitful approach here. The *lift* action triggers the $pickedup$ event, so the agent must avoid this action. This means that the *lift* action's postcondition of $holding$ must be relaxed as a requirement from the $move\text{-}across$ action (*Heuristic 3:* relax constraints and preconditions). The agent then defines a new action, $nudge$, which is essentially the $move\text{-}across$ action without the $holding$ requirement. Upon performing this action in the environment, the agent notices that the cup c can be moved from one location to another without picking it up (*Heuristic 4:* notice invariants). This is a major milestone in the agent's problem solving. It can now solve the intermediate problem (originally a MacGyver problem, as well) by following the plan

$$\pi = \langle reach\text{-}for(c), grasp(c), nudge(c, l_3), release(c), reach\text{-}for(x), grasp(x), nudge(x, l_2), \\ release(x), pick\text{-}and\text{-}place(c, l_3, l_1), reach\text{-}for(x), nudge(x, l_3)\rangle.$$

The agent might not know it yet, but it is much closer to solving the more difficult "not-touched" problem. Next, the agent might attempt to relax constraints on other primitive actions and discover that it can pick up, flip, and set down the cup on top of the block, as $set\text{-}down$ otherwise requires a clear location. In doing so, it might notice something unexpected, namely that the block disappears (*Heuristic 5:* notice anomaly). The agent can then hypothesize a new relation called $P(x, c, l_1)$ to account for this behavior and generate a new action, $cover$, to capture the dynamics (*Heuristic 6:* hypothesize predicates). After experimenting with this process, and through language-based human assistance, it might rename the predicate P to be $enclosed$.

Finally, with additional experimentation, the agent must learn a $scoot$ action, which is a variant of $nudge$ with added knowledge that there will be co-movement of the block and cup if the cup encloses the block. The purpose of this example is not to demonstrate a generalizable solution to a MacGyver problem, which no one has achieved to date. However, it does provide a rough sense of the types of knowledge and heuristics that could support creative problem solving in this domain.

Thus far, we have treated perceptual capabilities and primitive actions as symbols in cup world. However, in real-world settings, these primitive actions will be grounded in robotic actions that are parametrized as points in continuous 3D space, with numeric color and depth values. We can extend

our formulation to infinite MacGyver problems that, informally, require the agent to synthesize new symbols from real-valued grounded actions and percepts (e.g., learning to detect $maroon$ when it has a color detector and knowledge of the symbol red). We can pursue such questions in conjunction with research on integrating task and motion planning, and on learning symbolic operators from continuous spaces (Mourao et al., 2012; Konidaris et al., 2014; Kaelbling & Lozano-Pérez, 2013; Srivastava et al., 2014; Garrett et al., 2017).

5. Agent Capabilities and Subtasks

As we have seen, solving even simple MacGyver problems can be a challenging endeavor, requiring the ability to track and maintain cues and heuristics while remaining cognizant of the current state of problem solving. Nevertheless, we are optimistic that research progress can be made in this regard if we can leverage results from relevant subfields in AI and cognitive systems. Below is short, nonexhaustive list of subtasks that MacGyver problems appear to require:

1. Impasse Detection. As noted earlier, determining the existence of a MacGyver problem is intractable. Thus, the agent must use heuristics in an attempt to solve a MacGyver problem as a traditional planning task. It must also be equipped to detect unsolvability or at least intractability (Bäckström et al., 2013; Lipovetzky et al., 2016). Recent efforts by the planning community have started to address this issue through the "First Unsolvability IPC" at ICAPS 2016.

2. Domain Transformation and Problem Restructuring. Chunk decomposition and constrain relaxation have been well studied in psychology (Knoblich, 2009). The extensive literature in plan task revision has formally studied the effects of changing state, goal, and operators (Herzig et al., 2014; Göbelbecker et al., 2010).

3. Experimentation. The agent cannot know that scoot and pick-and-place have similar effects, so it must learn this through embodied interaction with the world. We can apply research on learning from exploration and intrinsically motivated reinforcement learning (curiosity, exploration, and play) to enable an agent to explore a problem space in search of operator variants (Pathak et al., 2017; Hester & Stone, 2017; Chentanez et al., 2005; Colin et al., 2016).

4. Discovery Detection. For the agent to have learned the $enclosed$ predicate, it must detect unexpected events, which in this case was an unexpected change in a fluent. Appropriate mechanisms for tracking uncertainty using probabilistic approaches (e.g., Bayesian and information-theoretic methods), in combination with salience and attention mechanisms, can help detect these events. From the real-world standpoint, the agent must possess capabilities to be able to execute this exploration and discovery process, including grasping and manipulating unfamiliar objects. These practical abilities are not trivial, and, in combination with intelligent reasoning, will provide a clear demonstration of agent autonomy while solving real-world problems.

5. Domain Extension. Finally, the agent must know how and when to assimilate new knowledge about its domain. This involves more than simply adding new domain elements; it must determine if the new knowledge will be consistent with existing content. Here, we can turn to research in belief revision that offers a formal approach to addressing these sorts of knowledge representation issues (Herzig et al., 2014).

In addition, the agent must posses a set of crucial supporting skills, such as abilities to (1) dynamically invoke and use families of heuristics, (2) consider internal and external cues at varying levels of abstractions, (3) operate in a 'problem-stewing' or cue-monitoring mode, (4) formulate and redefine symbolic knowledge, including propositions, operators and goals, (5) perform meta-level searches over a problem space, and (6) learn and acquire knowledge from different domains. We believe that, to design agents that can solve MacGyver problems, it is essential that independent lines of research be merged. Importantly, we must combine the subtasks mentioned above and the supporting skills into an integrated cognitive system that can be embodied and situated in physical or virtual environments.

6. Evaluating Agents and Measuring Research Progress

Any proposed framework for intelligent agents must be accompanied by a discussion of how to evaluate those agents. For the Turing Test and many of its variants, one can only measure agents by the ultimate but subjective human judgment. This involves the question of being able to identify the source of behavior as human or artificial. Essentially, this was an all or nothing proposition that was highly subjective. We argue that this is insufficient. The MacGyver formulation offers subjective and objective options that were not available in many previous approaches. We propose three subclasses of measures: problem-centric , solution-centric measures, and agent-centric.

Problem-centric measures are based on a MacGyver problem's inherent difficulty. Because such tasks have a domain-independent formulation, we can consider measures like the reachability of goal states and distance from start state, the size of the initial domain, the size of the minimal domain that contains both the start and goal state, and the existence and number of dead-end states (from which the agent can never solve the problem). Problem-centric measures, at their core, focus on the difficulty of MacGyver problems themselves, independent of the specific solution taken by the agent.

Solution-centric measures consider the solution found on a given MacGyver problem. Humans place high value on elegant and clever solutions to complex problems. We might quantify elegance and cleverness as complexity of an insightful strategy, the nature and number of domain transformations needed, the nature and number of heuristics and cues used in solving the problem, the time taken to solve a problem, the length of the solution plan, and in many other ways. Solution-centric metrics, when combined with problem-centric ones could even capture partial progress. For example, an agent making even limited progress on a very difficult problem might be scored higher than an agent that fully solves easier problems. Solution-centric measures could also capture subjective judgments of a human arbiter, such as judging whether a human or an artificial agent generated a solution.

Neither problem-centric nor solution-centric measures capture the inherent resource limitations of the agent and its environment. The agent's sensory-motor and cognitive capabilities limit how it initially represents problems, and which problems and strategies are viable. We would not expect a Roomba to pick up a block in cup world because it has no arms. Agent-centric measures might consider these resource limitations and therefore acknowledge that a task may be a MacGyver problem for one agent but not for another. Agent-centric measures could let us track and update the

knowledge an agent has acquired during its lifetime. We can ask if the agent has improved its overall problem-solving capabilities and used knowledge gained from solving one task in another one. Finally, agents could be placed in adversarial games with one agent being tasked to design MacGyver problems for the other. This would let us evaluate an agent's creative ability not only to solve MacGyver problems, but also to generate them.

We do not advocate for any one specific measure. Indeed, it might be best to take a hybrid approach that incorporates all three. The different metrics could let us study whether agents solving one MacGyver problem can solve similarly difficult (or easier) ones. We could also see whether the types of knowledge and heuristics that work with one class of MacGyver problems also work with another class.

Thus far, we have discussed evaluating problem-solving agents. However, we would also like to be able to track how the overall research community designs creative AI. To this end, we can track the progress on the independent subtasks identified earlier, which map well onto existing research agendas in AI subfields. In addition, we urge the community to invest time and funds to develop support skills needed for effective problem solving, such as the ability to draw on heuristics, to manipulate symbolic knowledge, and to play a role in integrated systems.

7. Conclusion

In this essay, we introduced a new class of AI challenges - *MacGyver problems* - defined as planning problems that are seemingly unsolvable, but where an agent can widen its representation of the domain, and maybe its understanding of the world, to discover solutions. Inspired by work in human insight problem solving and using the language of classical planning, we defined this class of tasks in a domain-independent framework that can generate domain-specific problem instances. We described one such instantiation using a toy domain called "cup world." We showed that recognizing and solving MacGyver problems are computationally intractable. Thus, a cognitive system must leverage families of heuristics, strategies, cues, and experiments to find solutions.

We walked through a heuristic approach to solving cup world, which also let us discuss the various agent capabilities needed to solve these problems. It is especially interesting to note the parallels between MacGyver problems and human insight problems, and we suggest that much can be learned from how humans solve them. We also provided a short and nonexhaustive list of research subtasks, that is crucial to building cognitive systems that can solve MacGyver problems. Finally, we discussed possible ways to measure and evaluate agents that attempt to solve such problems, as well as research progress as a whole. The formulation does not require specific internal mechanisms or particular solution strategies, but instead focuses on general features of creative problem solving.

We believe the formulation of such challenge problems is only the first step, albeit a fruitful one, towards designing creative cognitive systems. We hope the formalism will help guide research by providing a set of formal specifications for measuring AI progress based on the ability to solve increasingly difficult MacGyver problems. We believe the cognitive systems community is ideally suited to tackle these challenge problems, and we encourage the community to use the framework and pursue research on (a) designing additional domain-specific instances of MacGyver problems, (b) updating cognitive architectures to solve these problems, and (c) conducting studies that compare

performance of artificial agents to humans on these problems; even ones that are simple for humans like cup world might shed light on humans' heuristics.

Acknowledgements

We would like to thank Pat Langley for reading and commenting on earlier drafts of this essay.

References

Ackerman, J. (2016). *The genius of birds*. New York, NY: Penguin.

Bäckström, C., Jonsson, P., & Ståhlberg, S. (2013). Fast detection of unsolvable planning instances using local consistency. *Proceedings of the Sixth Annual Symposium on Combinatorial Search* (pp. 29–37). Leavenworth, Washington, USA.

Boden, M. A. (2010). The Turing test and artistic creativity. *Kybernetes*, *3*, 409–413.

Bringsjord, S., Bello, P., & Ferrucci, D. (2001). Creativity, the Turing test, and the (better) Lovelace test. *Minds and Machines*, *1*, 3–27.

Bringsjord, S., & Sen, A. (2016). On creative self-driving cars: Hire the computational logicians, fast. *Applied Artificial Intelligence*, *8*, 758–786.

Chentanez, N., Barto, A. G., & Singh, S. P. (2005). Intrinsically motivated reinforcement learning. *Advances in Neural Information Processing Systems* (pp. 1281–1288). Vancouver, Canada: MIT Press.

Cohen, P. R. (2005). If not Turing's test, then what? *AI Magazine*, *26*, 61–67.

Colin, T. R., Belpaeme, T., Cangelosi, A., & Hemion, N. (2016). Hierarchical reinforcement learning as creative problem solving. *Robotics and Autonomous Systems*, *86*, 196–206.

Cooper, S. B., & Van Leeuwen, J. (2013). *Alan Turing: His work and impact*. Waltham, MA: Elsevier.

Feigenbaum, E. A. (2003). Some challenges and grand challenges for computational intelligence. *Journal of the ACM*, *50*, 32–40.

Garrett, C. R., Lozano-Pérez, T., & Kaelbling, L. P. (2017). Strips planning in infinite domains. *arXiv preprint arXiv:1701.00287*.

Göbelbecker, M., Keller, T., Eyerich, P., Brenner, M., & Nebel, B. (2010). Coming up with good excuses: What to do when no plan can be found. *Proceedings of the International Conference on Automated Planning and Scheduling* (p. 81–88). Toronto, Canada: AAAI Press.

Harnad, S. (1991). Other bodies, other minds: A machine incarnation of an old philosophical problem. *Minds and Machines*, *1*, 43–54.

Herzig, A., de Menezes, M. V., de Barros, L. N., & Wassermann, R. (2014). On the revision of planning tasks. *Proceedings of the Twenty-First European Conference on Artificial Intelligence* (pp. 435–440). Prague, Czech Republic: EurAI.

Hester, T., & Stone, P. (2017). Intrinsically motivated model learning for developing curious robots. *Artificial Intelligence*, *247*, 170–186.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., & Girshick, R. (2017). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1988–1997). Honolulu, HI.

Kaelbling, L. P., & Lozano-Pérez, T. (2013). Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, *32*, 1194–1227.

Kaplan, C. A., & Simon, H. A. (1990). In search of insight. *Cognitive psychology*, *22*, 374–419.

Knoblich, G. (2009). Psychological research on insight problem solving. In H. P. Harald Atmanspacher (Ed.), *Recasting reality*, 275–300. Springer.

Konidaris, G., Kaelbling, L. P., & Lozano-Perez, T. (2014). Constructing symbolic representations for high-level planning. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (pp. 1932–1938). Quebec City, Canada.

Langley, P. (2017). Heuristics and cognitive systems. *Advances in Cognitive Systems*, *5*, 3–12.

Langley, P., & Jones, R. (1988). A computational model of scientific insight. In R. J. Sternberg (Ed.), *The nature of creativity: Contemporary psychological perspectives*, 177–201. New York, NY: Cambridge University Press.

Langley, P., Pearce, C., Bai, Y., Barley, M., & Worsfold, C. (2016). Variations on a theory of problem solving. *Proceedings of the Fourth Annual Conference on Cognitive Systems*. Evanston, IL.

Levesque, H., Davis, E., & Morgenstern, L. (2012). The winograd schema challenge. *Proceedings of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning* (pp. 552–561). Rome, Italy: AAAI Press.

Lipovetzky, N., Muise, C. J., & Geffner, H. (2016). Traps, invariants, and dead-ends. *Proceedings of the International Conference on Automated Planning and Scheduling* (pp. 211–215). London, UK: AAAI Press.

Mourao, K., Zettlemoyer, L. S., Petrick, R., & Steedman, M. (2012). Learning STRIPS operators from noisy and incomplete observations. *arXiv preprint arXiv:1210.4889*.

Ohlsson, S. (1992). Information-processing explanations of insight and related phenomena. *Advances in the psychology of thinking*, *1*, 1–44.

Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. *Proceedings of the Thirty-fourth International Conference on Machine Learning* (pp. 2778–2787). Sydney, Australia: PMLR.

Riedl, M. O. (2014). The Lovelace 2.0 test of artificial creativity and intelligence. *arXiv preprint arXiv:1410.6142*.

Sarathy, V. (2018). Real world problem-solving. *Frontiers in Human Neuroscience*, *12*, 1–14.

Schweizer, P. (2012). The externalist foundations of a truly total turing test. *Minds and Machines*, *22*, 191–212.

Srivastava, S., Fang, E., Riano, L., Chitnis, R., Russell, S., & Abbeel, P. (2014). Combined task and motion planning through an extensible planner-independent interface layer. *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 639–646). Hong Kong, China: IEEE.

Turing, A. M. (1950). Computing machine and intelligence. *MIND*, *59*, 433–460.

Wallas, G. (1926). *The art of thought*. London, UK: Jonathan Cape.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., & Mikolov, T. (2015). Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Wiggins, G. A. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, *19*, 449–458.