

The man behind the curtain: Overcoming skepticism about creative computing

Martin Mumford and Dan Ventura

Computer Science Department
Brigham Young University
Provo, UT 84602 USA
martindm@byu.edu, ventura@cs.byu.edu

Abstract

The common misconception among non-specialists is that a computer program can only perform tasks which the programmer knows how to perform (albeit much faster). This leads to a belief that if an artificial system exhibits creative behavior, it only does so because it is leveraging the programmer's creativity. We review past efforts to evaluate creative systems and identify the biases against them. As evidenced in our case studies, a common bias indicates that creativity requires both intelligence and autonomy. We suggest that in order to overcome this skepticism, separation of programmer and program is crucial and that the program must be the responsible party for convincing the observer of this separation.

Introduction

Demonstrations of computational creativity are often viewed with intense skepticism – much like a Victorian-era magician's trick full of smoke and mirrors. After all, creativity is regarded in many circles as a uniquely human characteristic, and so the claim of a *creative computer* invites immediate and often passionate skepticism. Even when an artificial system exhibits convincing creative behavior, the credit usually rests on the programmer as the true creative individual behind the act.

What can be done to convince the audience that there are no strings attached – that a program is being creative independently from its programmer? How far should they be allowed to probe, to test, and to know about the system's workings to be convinced?

It is important to motivate the separation of programmer and program in computational creativity applications. Consider a piece of software designed to monitor the landing gear on an aircraft. This software likely utilizes planning or decision-making algorithms, based on relevant conditions. If the software malfunctions in-flight, the aircraft may be damaged. Complex though the software may be, it cannot take the blame for following the instructions of its programming. Now consider a creative joke generator which tweets a new joke each day. One day, a generated joke happens to be highly offensive, and sparks criticism. This criticism cannot be targeted at the program, but at the programmer instead, for it is perceived to be following complex coded

instructions. This is especially important as technology becomes more complex, and the general public becomes less aware of the specific details of its implementation. After all, as the science fiction author Arthur C. Clarke puts it:

“Any sufficiently advanced technology is indistinguishable from magic.”

This leaves us with a powerful motivator to understand how people perceive the division of creativity between creator and creation. Because computers are currently perceived as incapable of autonomy and thought, as programmers, we will be credited for *and* be held accountable for what our programs do.

In this paper we focus on the issues of perception and skepticism regarding artificial creativity. This discussion is hardly new but rather a modern revival motivated by recent progress in the field. As creative systems become more advanced, exhibiting more compelling creative behaviors, and applications begin to appear in the wild, the discussion becomes relevant again.

We outline a high-level review of suggested properties of creative systems, as well as previously proposed tests for evaluating the creativity of a system. We then report on a brief case study illustrating the impact of interactivity on perception. This is supplemented by a survey taken by software engineers, computer scientists, as well as non-specialists, which exposes some of the primary obstacles in the public perception of artificial creativity. We also offer an example from popular culture which highlights the issue of perceived autonomy as it relates to creativity. Finally, we discuss the impact of these perceptions on the potential direction and progress of the field.

A History of Skepticism

The Lady Lovelace, upon hearing about the possible creativity of Charles Babbage's Analytical Engine, put forth the same argument that is still used today – as quoted in (Dartnall 1994), that “[Machines] have no pretensions whatever to *originate* anything,” having no autonomous thought, and thus cannot be considered creative.

Nearly two hundred years later, despite significant advances in machine learning and computational creativity, this remains the dominant perception, with some degree of truth. In an attempt to address the question of whether or not

computers have the capacity for creative acts, several characteristics of creativity have been put forth by behavioral and computer scientists.

Necessary and Sufficient Conditions

The search for qualities of creative systems is rooted in the question “What is creativity?” While an ill-formed and hotly contested question, it has nevertheless motivated scholars to seek out some of the necessary conditions to determine whether a system should be considered creative. The properties put forth so far are still subject to debate, and far from sufficient or exhaustive, but offer a guiding set of characteristics by which to begin judging the creativity of a system. An artificial system possessing many of these characteristics could be persuasively argued to be creative, because it shares those attributes with creative humans.

Properties of the Artefact The most straightforward way to judge a system is by the artefacts it produces. This requires no knowledge of system process, and success is often measured by comparing human-generated and computer-generated artefacts side-by-side or in a blind preference test.

Creative qualities artefacts should exhibit have included quality (Wiggins 2006; Colton 2008b), novelty or imagination (Ritchie 2007; Wiggins 2006; Colton 2008b), robustness or variability, and typicality (Ritchie 2007; Colton 2008b).

Properties of the System In addition to the artefacts, the process of creation itself has been suggested as a major factor in judging creative acts. Some of the aspects of the process include: appreciation or aesthetics (Colton 2008b; Colton, Pease, and Charnley 2011), individual style, intentionality, the ability to explain or justify decisions (Colton, Pease, and Charnley 2011), social context in a larger community of creators (Saunders and Gero 2001; Jennings 2010), and taking the audience into account (Maher, Brady, and Fisher 2013). Recent work has even been done on meta-evaluation – the evaluation of creative evaluation frameworks (Jordanous 2014).

Furthermore, we understand that the ability to learn is intertwined with the ability to create. A system that can learn its own fitness function for an aesthetic measure, for example, is arguably more creative than one that must have it explicitly specified by the developer, and some work has been done on automatically learning aesthetics (Colton 2008a).

Tests of Computational Creativity

A few general psychological creativity tests exist but are often in a format inaccessible to computers. For example, the Torrance Tests of Creative Thinking (TTCT) involve many verbal and drawing tasks which are beyond the abilities of modern computer vision and natural language processing. And so, in addition to a set of essential qualities for creativity, academics have sought to define a “Turing Test” for creativity more suited to computers.

Even if a convincing, well-defined test existed, the concept itself has been criticized (Pease and Colton 2011) as limiting the potential style and variety of creativity in

computers, much as the original psychological counterparts (Kim 2006) have been criticized.

Turing Tests have been subject to scrutiny by the Chinese Room argument (Searle 1980), which appears to coincide with the most common criticism of creative systems – that no matter how creative they may seem, their internal workings could still comprise some form of Searle’s rule-book. The Lovelace Test (Bringsjord, Bello, and Ferrucci 2003) tries to address this issue by dealing with the separation of programmer and program, rather than focusing on the system exclusively. Specifically, one of the requirements of the Lovelace Test is that the programmer cannot explain how an artefact was generated by the system, even when given ample time to do so.

Notably, Bringsjord implies that the Lovelace Test can only essentially be passed when a system is perceived of as ‘thinking for itself’, and ‘having a mind’. The perception of creativity is thoroughly entangled with the perception of intelligence and autonomy. While programmer surprise and inability to explain can help to establish the system as a separate entity, such surprise can be faked. Overcoming residual skepticism may require methods that establish the autonomy of a system without the need to rely on programmer reactions.

Modern Skepticism

In its current state, the field of Computational Creativity continues to face heavy skepticism from non-specialists. This is actually quite healthy for our field, as such skepticism provides a motivation to build systems that are not only theoretically sound, but convincingly demonstrable and socially acceptable. We explored the primary complaints and biases against the notion of creative computers, with the intent to discover the core issues that need to be addressed. This exploration revolved around the question, “What would it take to subjectively convince someone of a system’s creativity?”

Man behind the curtain: A case study

In order to explore what it would take to alter people’s perception, we created a simple analogy-making program, the output of which might be considered creative. This program was presented in three stages to 35 participants who were told that it was powered by a creative artificial intelligence.

- Stage one: No interactivity. The user presses a button and the computer produces a random analogy.
- Stage two: Selective interactivity. The user selects two nouns from a short list, and the computer produces an analogy between them.
- Stage three: Full interactivity. The user inputs any two concepts, and the computer produces an analogy.

The first two stages only *appear* to be creative – but in reality the computer is selecting from a pool of pre-generated analogies. Although the analogies could have been retrieved nearly instantly, a loading screen was presented to give the appearance of processing happening ‘behind the curtain.’

The pre-generated analogies were created by hand using two seemingly unrelated concepts, and connected in a clever

and humorous way using similar properties between the two. For example, ‘cats are like lawnmowers: temperamental and destructive.’ For stage two, items could be selected from two lists of five, making 25 possible analogies in the pool.

Since we did not actually construct a creative analogy generator, the only way to provide full interactivity was to utilize a human operator using a networked device to ‘respond’ to analogy requests. In our case we actually placed a man behind a curtain – the operator was sitting behind a partition nearby as users participated. In order to ensure consistent quality and style of analogies between different stages, the writer of the analogies for the first two stages also served as the operator for the third.

Users were asked at random to either participate in one of the three tiers, or to move through all three consecutively. After observing the analogies that were ‘generated’ by the computer, they were asked to evaluate the creativity of the task in general, as well as to determine where they felt the attribution of creativity belonged on a 5-point Likert scale from programmer to program.

First, we observed that as the degree of allowed interactivity increased, the users were more inclined to test the system for patterns or trickery. When asked to split the attribution of creativity between programmer and program, a 1.0 on the scale represented ‘all programmer’ and 5.0 represented ‘all program’, where 3.0 represented an equal responsibility between the two. The average placement was 2.25 for stage 1, 2.46 for stage 2, and 3.1 for stage 3, showing an improved willingness to attribute creativity to the computer.

Second, we observed that those who tried successively more interactive levels attributed dramatically more creativity to the system (more so than those participating in individual tests). This is likely because they had to revise their own assessments multiple times.

Finally, among the highly skeptical, we found that a clear, repeated input-output pattern caused any and all creativity of the system to be discounted. Because the first two tests simulated a creative system by drawing from a pool of pre-generated analogies, and that pool was not particularly deep, astute users would probe the system until it eventually produced a duplicate. Each user who discovered a duplicate would invariably rate the system as having low creativity.

Conflicts

There also exist a few ‘double-edged swords’ in a creative system that can subjectively decrease or increase the perception of creativity.

Knowledge of System Keeping the system as a black-box (no knowledge) forces the user to evaluate the system based on the artefacts alone. Unfortunately this can mask the true creativity or lack of creativity in a system. For some individuals, keeping the system internals unknown is crucial, based on the notion that creative people produce artefacts *ex nihilo*, or that the creative process is fundamentally mysterious and cannot be explained. To expose the process might disrupt the appearance of creativity for these individuals.

For example, it is trivial to implement a genetic algorithm to *evolve* a painting of the Mona Lisa, simply by setting the

fitness function to be a pixel-by-pixel comparison between the phenotype and a picture of the Mona Lisa. Yet watching the painting evolve and take form in real time, it is easy for an outside observer to attribute to the program some level of intelligence and creativity. Of course, had the curtain been pulled back and the process exposed to the observer, they would have been disappointed at the naive way in which the system randomly combines and mutates.

Exposing the high-level workings of the system allows the observer to make judgments about the process itself. However, exposing all of the system’s process could remove the mystery of the process, leading to the perception that the program is ‘merely following instructions,’ no matter how complex they may be.

In our analogy-making experiment, several technically-minded users attempted to discover the internal workings, inventing progressively harder requests meant to probe for templates and patterns. These individuals were impressed if they could not determine a consistent pattern, and remained unconvinced if they could imagine a clear process by which the artefacts were generated.

Humanized Process People tend to project human emotions and behaviors onto non-human objects. A process that seems more ‘human’ (pausing as if in thought, backtracking, slight errors, etc.) can improve the perception of creativity. As Colton observes (2008b),

“...it is apparent that being able to watch The Painting Fool create its paintings means that people project more value onto them than they would if the paintings were rapidly generated through, say, an image filtering process. This seems to be because they can project critical thought processes onto the software, and empathise with it more.”

On the other hand a process with elements that appear highly computer-like (superhuman speed, enormous scale, lack of mistakes, logical explanations, etc.) can sometimes lend strength to the perception that a computer is doing all the work. Ultimately, the most persuasive portrayal might incorporate aspects of both philosophies.

The Creative Threshold

We conducted three surveys among different audiences asking about computers and creativity. Each participant was asked to rate whether computers were currently capable of creativity, and whether they will someday be capable of creativity, on a Likert scale from 0 to 10. They were then asked to define what they thought were essential requirements or characteristics of creativity. Finally, they were asked to describe what behavior or characteristics a system should have to convince them that it was creative. The exact questions and selected responses can be found in Appendix A.

We first sought to understand the opinion of those that were technologically literate, but unfamiliar with programming and code. This survey was conducted on Reddit (a social bulletin board website) and had 75 respondents. We did not collect demographic information, but general statistics of Reddit users are can be found elsewhere (Duggan and Smith 2013) for those interested.

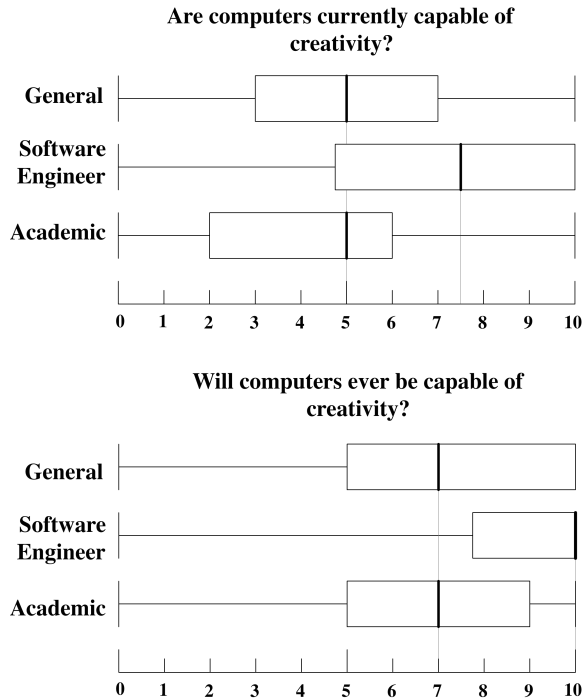


Figure 1: Quantitative analysis of responses by group: each boxplot shows the first quartile (left), median (bold) and third quartile (right).

For comparison, the same survey was given to a group of 26 software engineers working in the industry, and again to a group of 37 computer science professors and graduate students at Brigham Young University.

We originally anticipated that people familiar with programming or AI would have a deeper understanding of its potential, and thus show less skepticism at the concept of computational creativity. Academics, being the most familiar with current research and progress were expected to show the strongest optimism. However, the academics surveyed displayed somewhat more skepticism than any other group. More surprising still, the programmers demonstrated a disproportionately high level of confidence.

Among the open-ended responses in all three groups about the requirements for creativity, eight broad classes emerged:

- **Lateral Thinking:** Often described as ‘outside the box’, including methods of thinking that ‘do not rely on logic,’ going beyond formal inductive and deductive reasoning.
- **Flexibility:** The ability to work within arbitrary constraints and handle many kinds of tasks.
- **Aesthetics:** Taste, or the ability to judge quality and discern good artefacts from bad ones.
- **Novelty:** Producing artefacts which are original, unique, or different from what has been seen before.

- **Analogy:** The ability to make interesting analogies between seemingly unrelated concepts, or to combine or otherwise transform old concepts into something new.
- **Self-Improvement:** The ability to learn from experience over time.
- **Autonomy:** Often described as ‘independent thought’, ‘unique intelligence’, or emphasizing a lack of pre-defined rules.
- **Human Emotions:** bravery and curiosity were the most common human emotions listed.

Particularly among the most skeptical participants (those who rated it unlikely that computers are or ever will be creative), autonomy was the top priority for creativity. Responses such as, ‘agency’, ‘choose for itself’, ‘independent intellectual ability’, and ‘independent thought’ suggested that the system must be autonomous to convince them. Consider the following responses specifically about code: ‘not based on algorithms’, ‘not a result of programming’, ‘create its own programs’, ‘no explicit code detailing what to do’, and ‘write the program on its own’.

Of course, computer programs can already exceed their original programming, through machine learning for example. Decades ago, classical AI algorithms were already capable of learning things that their creators did not know, and acquiring skills that their creators did not possess. The observed unwillingness to acknowledge a program as an independent entity appears to stem from a philosophical standpoint, even among other computer scientists, that code merely follows instructions (albeit extremely complex ones). This is a valid point of debate, though a particularly fuzzy one, since even creative humans could be argued to be following a complex set of chemical and psychological instructions.

This need for an intelligent autonomous entity separate from the programmer sparks interesting questions. Is it possible for a computer system to possess all of the creative attributes typically outlined in our field (appreciation, skill, novelty, typicality, intentionality, learning, individual style, curiosity, accountability), and yet *still* not be creative? Alternatively, can a machine be creative without being intelligent? More broadly, is *general* or *strong* artificial intelligence necessary before people become comfortable with ascribing creativity to a machine?

We are not prepared to claim that general intelligence is required for creative behavior, but instead observe that people are generally unwilling to *attribute* creativity to a system until it appears to be a separate, intelligent entity.

In popular culture

We turn to a portrayal of creative computing in popular culture to demonstrate the perception that in order to be creative, a computer must have autonomous thought and exceed its programming.

In an episode of the television series *Star Trek: Voyager*, a trial is conducted to determine whether a computer program (the holographic doctor) should retain the rights to the creative work (holonovel) which he created. Part of the trial

appeals to the argument that attributes of the artefact are enough to deem a computer creative:

BROHT: A replicator created this cup of coffee. Should that replicator be able to determine whether or not I can drink it?

TUVOK: But I have never encountered a replicator that could compose music, or paint landscapes, or perform microsurgery. Have you? Would you say that you have a reputation for publishing respected, original works of literature?

BROHT: I'd like to think so.

TUVOK: Has there ever been another work written about a hologram's struggle for equality?

BROHT: Not that I know of.

TUVOK: Then in that respect, it is original.

BROHT: I suppose so.

TUVOK: Your honour, Section seven ... defines an artist as a person who creates an original artistic work. Mister Broht admits that the Doctor created this programme and that it is original. I therefore submit that the Doctor should be entitled to all rights and privileges accorded an artist under the law.

However, the appeal to originality was ultimately not enough evidence to convince the judges. The winning argument rested on the doctor's autonomy and independent thought:

KIM: He decided it wasn't enough to be just a doctor, so he added command subroutines to his matrix and now, in an emergency, he's as capable as any bridge officer.

ARBITRATOR: That only proves the Doctor's programme can be modified.

KIM: Your honour, I think it shows he has a desire to become more than he is, just like any other person.

JANEWAY: Starfleet had programmed him to follow orders. The fact that he was capable of doing otherwise proves that he can think for himself.

In this fictional case, as with the personal biases discovered in the survey, the deciding factor is intelligent, autonomous thought. This gives rise to several open questions for discussion:

- In what way are different aspects of intelligence interrelated with different aspects of creativity?
- Is intelligence necessary for creativity?
- If so, is artificial general intelligence necessary for general creativity?
- Is the threshold of evaluating creativity arbitrarily lower for humans or living beings such as crows (which have been shown to solve problems creatively) than for inanimate systems like programs?
- Though increasing the *intelligence* of our creative programs could boost creative perception, would it necessarily have a positive impact on the true creativity of the system, or the quality of artefacts it produces?
- How best can we convincingly demonstrate the autonomy of a creative system?

Future Skepticism

There are many current approaches we can utilize to overcome some of the perceptual barriers, one of which is the capacity for a program to code parts of itself. Work is already being conducted in *creative code generation* (Cook 2013), which could boost the perception of autonomy by non-specialists. Metaprogramming (writing code that writes code) does not necessarily translate to more creative programs, but it certainly lends credence to the idea that the program is separate from the programmer. This in turn provides an entity other than the programmer to which creativity can be attributed. Additionally, using machine learning methods to improve a system's aesthetic sense, cognitive ability, or skill level strengthens the claim that it is able to 'exceed its original programming'.

More broadly, we need to consider the impact of these perceptual issues on the goals of our field as a whole. To what extent should public opinion factor into our goals? Several of the requirements for creativity are already shared by both public opinion and computational creativity researchers. A heavier emphasis on boosting perception may only serve as a motivation for trickery and selective methods of presentation, which would not necessarily increase the creativity of our systems or the quality of artefacts they produce.

Consider the difference between the aircraft landing gear software and the joke generator in the introduction. We understand there is a creative difference between aircraft software and a joke generator. Aircraft software was designed to be predictable and react to very particular situations in very particular ways – a clear mapping from inputs to outputs. Thus a software failure is likely to be the fault of the programmer. However, a joke generator is ideally unpredictable – *that's the point*. Its creator may be surprised at the jokes it generates, but the audience cannot necessarily ascribe this to the generator program being an autonomous entity. It could then be argued that the programmer is indeed responsible for the offensive joke, but unknowingly so, because the programmer was unaware of the range of possible jokes that the program could generate.

A parent is socially responsible for the behavior of their child, but they cannot take credit for the child's creative acts or creative capacity, and nor can a mentor or teacher. However this relationship changes dramatically in software, where the programmer is not merely training an existing system, but making architectural decisions about the way it should think. If we could manipulate or condition the human brain to be more creative, or to deliberately specify how the thought process works, would the credit for the individual's creative acts rest partly on us?

A primary goal of our field is to shift the burden of creativity from ourselves to our programs. However, our level of direct involvement in the minds of our machines makes this transference difficult, despite our best efforts to facilitate it. The philosophical question to ask is whether this difficulty is entirely a matter of perception, in which case it is a problem of persuasion, or whether more of ourselves resides in the machine than we would like to admit. This entanglement between creator and creation may be unavoidable.

able, until our creative systems can be considered separate, intelligent entities with *independent thought*, at which point we open an entirely different can of worms.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

References

Bringsjord, S.; Bello, P.; and Ferrucci, D. 2003. Creativity, the Turing test, and the (better) Lovelace test. In *The Turing Test*. Springer. 215–239.

Colton, S.; Pease, A.; and Charnley, J. 2011. Computational creativity theory: The face and idea descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*, 90–95.

Colton, S. 2008a. Automatic invention of fitness functions with application to scene generation. In *Applications of Evolutionary Computing*. Springer. 381–391.

Colton, S. 2008b. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20.

Cook, M. 2013. Creativity in code: generating rules for video games. *ACM Crossroads* 19(4):40–43.

Dartnall, T. 1994. *Artificial Intelligence and Creativity: An Interdisciplinary Approach*, volume 17. Springer Science & Business Media.

Duggan, M., and Smith, A. 2013. 6% of online adults are Reddit users. *Pew Internet & American Life Project* 3.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Jordanous, A. 2014. Stepping back to progress forwards: Setting standards for meta-evaluation of computational creativity. In *Proceedings of the 5th International Conference on Computational Creativity*.

Kim, K. H. 2006. Can we trust creativity tests? a review of the Torrance tests of creative thinking (TTCT). *Creativity Research Journal* 18(1):3–14.

Maher, M. L.; Brady, K.; and Fisher, D. H. 2013. Computational models of surprise in evaluating creative design. In *Proceedings of the 4th International Conference on Computational Creativity*, 147–151.

Pease, A., and Colton, S. 2011. On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB Symposium on AI and Philosophy*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Saunders, R., and Gero, J. S. 2001. Artificial creativity: A synthetic approach to the study of creative behaviour. In *Computational and Cognitive Models of Creative Design V*. Key Centre of Design Computing and Cognition, University of Sydney. 113–139.

Searle, J. R. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences* 3(3):417–424.

Appendix A: Survey Responses

- Question 1: Do you think that computers are currently capable of being creative?
- Question 2: Do you think computers will ever be capable of creativity?
- Question 3: Name a couple of capabilities or traits required for someone to be considered ‘creative’
- Question 4: Briefly, what would a computer program have to do to convince you that it (not the programmer) was being creative?

Selected responses:

Q1	Q2	Q4	Q5
0	1	Predictive capacity, Agency, Contextual analysis	Prove to me it has agency to choose for itself
6	8	Iterative thinking and creation, ability to change direction mid-production, show work	Show steps, come to different conclusions when fed similar data/asked similar questions
3	5	must be a sentient being	since the AI would likely learn through formulas/programs created by the programmer, if it could create its own programs that are beyond human comprehension then that would be creative
9	10	New Ideas, Take an old idea and adapt it to a new situation	Maybe create a recognizable graphic from lines or circles or something Or respond to questions asked in ways that were unexpected and unpredictable
7	9	Come up with a new and unseen “thing” or take something old and use in a new or different way	Do not know
6	7	problem solving	solve a problem using non-data inputs or observations
0	3	Original thought, inspiration	Come up with an idea that hadn’t been thought of before
7	10	Not merely following rules, affect and logic combined	It would have to modify its own programs
3	4	capable of thinking ”outside the box”, coming up with innovative solutions to various problems	manifest fully independent intellectual ability
10	10	free thought	adapt to change
10	10	something able to come up with new ideas	respond to complex questions and problem solve
5	6	Think of ideas and new things on your own	Write the program on its own to show its creativity
4	9	innovation, unorthodox solutions	create a new idea
3	6	Inventive, open minded, designer	Synthesize to make something unique and relative to a need, feeling, etc. May have an aesthetic component
4	6	free choice	make something creative w/o human input