Creativity veRsus Classical Computation

Eric P. Nichols, Center for Research on Concepts and Cognition
Alexandre Linhares, Getulio Vargas Foundation

Introduction

The words "creativity" and "computation" conjure up vastly different images. "Creativity" evokes artists, musicians, breakthroughs in physics, surprise, and beauty, whereas "computation" summons up ideas of calculation, machines, gears, logic, and precision. Can we imagine a mechanism by which computation can yield creative results? The idea verges on paradoxical.

Of course, if we regard humans, and in particular human brains, as electro-chemical machines obeying the laws of physics, then we can easily imagine that the creative output of brains is in principle possible to simulate via computation. This is neither satisfying nor practical for those of us who want to build creative machines. Instead of aiming for the science-fiction goal of replicating a brain *in silico,* we would like to understand the essence of creativity and compile it in to our existing computational mechanisms. We discuss how the classical AI idea of Automatic Programming (AP) can be reconsidered in terms of an architecture whose goal is creative solutions to AP problems.

Fluid Concepts Architecture

Members of the Fluid Analogies Research Group (FARG) at Hofstadter's Center for Research on Concepts and Cognition (CRCC) have developed several cognitive models in the past few decades, spanning various domains but based on a similar underlying architecture (henceforth, the Fluid Concepts Architecture). We believe that this architecture can be extended to solve problems such as those envisioned by early researchers in Automatic Programming (A).

Automatic Programming

The goal of Automatic Programming (AP) is to generate a complex program automatically from a shorter list of program specifications. This is a tremendously hard problem, similar to natural language understanding; thus actual implementations only work in tightly controlled circumstances, where the AP engine becomes little more than a special-purpose compiler. The original lofty goal of AP, however, would require more creativity on the part of the computer.

We focus discussion here on the use of creativity in AP, where creativity is required in the solution process. We touch on how this work could be extended to solve not only AP, but also automatic creativity (AC) problems in an artistic domain (such as generating music given specific constraints).

Automatic Programming *and* Creativity: Self-assembling computer code through "hedonic" feedback

We propose that humans have propensities for certain behaviors regulated by hedonic feedback. Consider the following examples:

(i) A baby is playing with a shape sorter, and the baby's behavior is extremely awkward and inefficient. Whenever a shape falls into place, however, the baby seems to be puzzled and excited about the event. This increases motivation for yet another attempt.

(ii) Rats in a lab are given three levers; the first one brings food; the second one brings a fertile female; and the third one provides a stimulus to the brains pleasure center. The rat rapidly finds itself pulling the third lever, foregoing other desirable options, until it dies of exhaustion.

(iii) As an extreme -- but similar -- case, consider a drug addict, who cannot give up the addiction, despite numerous promises to do so.

Humans tend to reproduce behavior that is deemed pleasurable, and inhibit behavior deemed painful or unpleasurable. Based on this observation, we have been experimenting with an extension to the Fluid Concepts Architecture, where hedonic feedback systems cause the model to learn how to behave in the future to recreate the positive conditions and avoid the negative. Specifically, feedback influences the probabilities for individual bits of code ("codelets") to run, and encourages specific useful sequences of codelets to occur as a larger-scale unit. This aggregation of codelets into larger chunks effectively provides a way to perform automatic programming in a Fluid Concepts system. Creativity in solutions occurs here because the system stochastically tries many different approaches in parallel, learning which ones result in positive feedback.

We have applied this model to the problem of sorting a list of numbers. After specifying the domain of the problem, the system is given a "goal" function (the score goes up when numbers are sorted correctly) but it is unaware of the nature of the overall goal. This "goal" is meant only for the purposes of having an automatic hedonic feedback system: that is, whenever two numbers are switched (and ordered), the system receives a positive hedonic feedback. This feedback will eventually reprogram the system's behavior towards having higher propensity to execute the "same" operations. Note that, rather like the baby or the rat, the system is not able to explicitly say what its goal is, or why it is doing such behavior. It is only something that the system considers "worthwhile" doing (at the expense of other potential options).

This proposal is different from previous attempts at "autoprogramming". Though we cannot provide full details and careful comparisons here, mechanisms such as genetic programming, neural networks, reinforcement learning, and classifier systems, are, each in their own way, very different in nature to this. Our idea stems from the basic behaviorist observation of stimulus-response driving and inhibiting behavior; but adapted also to subcognitive inner workings that codelets are intended to model. Hedonic feedback may regulate not only observable behavior, but also cognition. It may be one of the driving forces behind "goal-seeking and motivation" for computational models.

Initial experiments demonstrate the feasibility of the approach. It must be studied whether the approach can scale; how some the parameter settings involved affect behavior, whether it could match human data on experiments with emotionally-driven behavior, and how it should impact future Fluid Concepts Architectures. The studies, so far, bring more questions than answers. We hope, however, that this approach may be potentially able to shed light on some of the deepest questions involving human motivation, hedonic systems, and, perhaps, highly creative people's obsession with perfection.

References

Boden, M. (1995) “Creativity and Unpredictability.” *Stanford Electronic Humanities Review* 4, no. 2 (1995).

Hofstadter, D. R., and FARG. (1995) *Fluid Concepts and Creative Analogies.* New York: Basic Books.

Rich, C. and Waters, Richard C. (1988). "Automatic Programming: Myths and Prospects." *Computer*, 21(8), pp. 40-51.

Rowe, J. and Partridge, D. (1993) "Creativity: A survey of AI approaches". *AI Review.*7, pp. 43-70.