An Attempt to Generalize AI Part 18: A Brief Summary So Far

By Paul Almond

19 August 2010

Website: http://www.paul-almond.com
E-mail: info@paul-almond.com

This is the eighteenth in a series of articles attempting an overview of how minds may work and how similar systems could be implemented in computers. One of the previous articles, *An Attempt to Generalize AI – Part 15: A Complete Description*, available at http://www.paul-almond.com/AI15.pdf, provided a description of the cognitive model, but this was a long article. The purpose of this article is to provide a very short summary. This article will leave many questions unanswered: As a summary, it has to. However, it will give a general idea of the kinds of things happening in the system, and how it is supposed to do something like thinking. Readers wanting a more detailed description can obtain that from Part 15.

1 Introduction

This is the eighteenth in a series of articles attempting an overview of how minds may work and how similar systems could be implemented in computers. One of the previous articles, *An Attempt to Generalize AI – Part 15: A Complete Description*, available at http://www.paul-almond.com/AI15.pdf, provided a description of the cognitive model, but this was a long article.[1] The purpose of this article is to provide a very short summary. This article will leave many questions unanswered: As a summary, it has to. However, it will give a general idea of the kinds of things happening in the system, and how it is supposed to do something like thinking. Readers wanting a more detailed description can obtain that from Part 15. Readers wanting to know more about some of the issues remaining with this model, and how they might be approached, may wish to read *An Attempt to Generalize AI – Part 17: What I'm Thinking*, available at http://www.paul-almond.com/AI17.pdf, though I advise reading this article first.[2]

In previous articles, I have used a lot of references, particularly to other articles in the series. I want this article to be as short and simple as reasonably possible, so there will be minimal referencing here. However, all the previous articles in the series are listed in the bibliography.

[1] Almond, P., 2010. *An Attempt to Generalize AI - Part 15: A Complete Description*. [Online] paul-almond.com. http://www.paul-almond.com/AI15.pdf or http://www.paul-almond.com/AI15.doc.
[2] Almond, P., 2010. *An Attempt to Generalize AI - Part 17: What I'm Thinking*. [Online] paul-almond.com. http://www.paul-almond.com/AI15.pdf or http://www.paul-almond.com/AI15.doc.

2 Basic Functioning of the Hierarchy

2.1 Basic Structure of the Hierarchy

The system is a hierarchy of *pattern instances*. Each pattern instance is a simple, computational unit, with a conceptual value of 0/1. By "conceptual" I mean that in an idealized view it has this value, but in a working system we will rarely know this value with certainty: It will be represented by a probability.

Each time an input or output occurs, the value is stored. These values form the bottom level of pattern instances. The rest of the pattern instances are used to build up the hierarchy on top of this. The bottom level of the hierarchy contains pattern instances corresponding to inputs/outputs occurring over a period of time, and the upper levels of the hierarchy are an abstraction of this. The conceptual value of a pattern instance is unchanging, as it depends, directly or indirectly, on inputs/outputs occurring at specific times.

Each pattern instance accepts pattern inputs from other pattern instances in the hierarchy and follows some logic to generate its own value. (An exception is the bottom-level pattern instances, corresponding to external inputs/outputs: Each of these automatically has the value with which the relevant input/output occurred.)

Pattern instances are grouped together in *patterns*. The pattern instances of a pattern can be spread throughout the hierarchy. An important idea is that *all the pattern instances of a pattern share the same logic and have something in common in the way they are connected to the hierarchy*. A pattern is a set of pattern instances that share the same logic meet some criteria in the way they are wired into the hierarchy. An analogy: It is as if someone read the "instruction manual" for a pattern, and it told him to go through the hierarchy, adding pattern instances for that pattern to it and connecting them to other pattern instances. The logic of the pattern instances in a pattern, and what they have in common about they way they are wired into the hierarchy, is defined by the *pattern specification* of that pattern.

2.2 How the Hierarchy Works

There will tend to be uncertainty about the values of pattern instances, so pattern instances in a working system are represented by probability values: The hierarchy is therefore a probabilistic network.

When an input/output occurs, the relevant pattern instance in the bottom level of the hierarchy is updated with a probability value of 0/1. However, this only works for inputs/outputs that have occurred. What about inputs/outputs that have not occurred? This is dealt with by propagating probabilistic information through the hierarchy.

The most obvious method is to use pattern instances' logic, but probabilistically. When the pattern inputs for a pattern instance are known (which will come from the pattern instances which it uses as pattern inputs), we can use the logic for that pattern instance to determine its pattern output. When we only know the pattern inputs for a pattern instance partially, or probabilistically, we can still apply the pattern instance's logic probabilistically. E.g. if a pattern instance has four inputs, A, B, C and D, which come from other pattern instances, and we have probabilities of 0.24, 0.07, 0.85 and 0.5 respectively, the pattern instance's logic can be looked up and used to determine a probability for the pattern output. This can work upwards *and* downwards: If we have a probability for a pattern instance's pattern output, we can use the pattern instance's logic to obtain a probability for any pattern instances acting as its pattern inputs.

We will have information about inputs/outputs that have occurred, and the above process will allow that to be propagated through the hierarchy. We want to be able to get probabilities that tell us *a lot* about pattern instances corresponding to future inputs/outputs, and the above method will not be good enough for that.

This is where patterns come in. The pattern instances of a pattern will be spread out throughout the hierarchy. Some will be connected into parts of the hierarchy that depend a lot on past inputs/outputs – which we already know about. Some will be connected into parts of the hierarchy that depend more on future inputs/outputs – and we will know less about these. However, since the pattern instances in a pattern are all related by being part of that pattern, we can use them as a kind of swarm intelligence. For a particular pattern, we can look at the pattern instances that we know a lot about. This will give us information about the relative frequencies with which different combinations of pattern inputs are likely to occur. The point, here, is that whatever statistics we find can then be applied to other pattern instances in the same pattern, about which we know less: Being in the same pattern makes them part of the same "population" and subject to any statistical tendencies observed for that population.

As an example, suppose we encountered a pattern instance with pattern inputs A, B, C and D with probabilities of 0.75, 0.6, 0.55 and 0.52 respectively. Here, we know something about the A and B pattern inputs, and not much about the C and D pattern inputs. BUT we will already know about many more pattern instances for this pattern which are more strongly dependent on inputs/outputs that have already occurred and for which we have a lot more information on *all four inputs*. This will tell us something about the likelihood of different combinations of pattern inputs, and it will allow us to assign meaningful probabilities to *all* the pattern inputs – *including C and D*.

We can use this to propagate probabilistic information much further into the hierarchy, both upwards and downwards so that, ultimately, bottom-level pattern instances corresponding to inputs/outputs that have not yet occurred are being assigned meaningful probability values. The hierarchy is now making useful predictions of future input/output values.

3 Action Selection

3.1 The Action Selection Process

An evaluation function score value is continually computed from recent inputs and provided to the hierarchy, encoded as inputs, so that the corresponding pattern instance values appear in the bottom level.

When an output is required, the system tries both possible values – 0 and 1. For each value, the bottom-level pattern instance corresponding to the output is updated with that value as if it had just occurred, and the probabilistic information is propagated through the hierarchy. The probabilities of pattern instances corresponding to future input of an evaluation function score value are then examined, giving an indication of the expected score after the output occurs with this value.

After doing this for both output values, the output value with the highest score is selected. The output actually occurs with this value and the hierarchy is updated accordingly.

3.2 What Is Really Happening in the Action Selection Process

The above process may appear to be the main way by which actions are selected, but action selection is really happening in the hierarchy itself. When an output value is tried, the hierarchy's predictions of the system's own follow-on actions will determine the context of this output and the eventual score. As the hierarchy relates inputs and outputs, part of the model will describe the system's own behavior, making predictions about it, and this is what we refer to as a "self". The idea that the self is associated with a system's own modeling of itself has previously been suggested by Metzinger.[3]

The action selection process is needed to establish an initial history that goes "the right way", and to prevent noise from degrading the system's behavior. One issue is whether such an approach needs to be modified, giving one that involves coming in "higher up", with changes being made deeper into the hierarchy.

---

[3] Metzinger, T., 2003. *Being No One: The Self-Model Theory of Subjectivity*. Cambridge (MA): MIT Press. Metzinger, T., 2009. *The EGO Tunnel: The Science of the Mind and the Myth of the Self*. New York: Basic Books.

4 Relevance

We need to restrict the hierarchy to using those patterns that are useful and, from those patterns, using those pattern instances that are useful. This is primarily done by the *exploratory relevance process*.

All this is based on the idea that the action selection process is looking at probability values for pattern instances corresponding to specific future inputs. The most useful patterns and pattern instances are therefore those that contribute to reduction in uncertainty of the specific, bottom-level pattern instances, corresponding to future inputs, used in the action selection process. These pattern instances are assigned some degree of relevance. Relevance is then back-propagated to pattern instances that have reduced their uncertainty in propagation, then relevance is back-propagated to pattern instances that have reduced the uncertainty in these and so on. Ultimately, the relevance of any pattern instance in the hierarchy is known.

The hierarchy is pruned according to its relevance. Pattern instances are continually removed, while new ones are added. The lower a pattern instance's relevance, the more likely it is that it will be removed at any time. Higher-relevance pattern instances will tend to stay in the hierarchy for longer. Any new pattern instance will have to connect to the existing pattern instances in the hierarchy as its pattern inputs, and these will tend to be ones that have remained due to high relevance. The process is an exploratory one, with the hierarchy growing from existing high relevance regions.

The exploratory relevance process, as well as selecting appropriate pattern instances, also involves selecting appropriate patterns for use, based on their usefulness, with regard to reducing uncertainty in the pattern instances that are of interest, when tried.

A modification to this process, which involves back-propagating less relevance back from a pattern instance when it is known about with a high degree of confidence, provides forgetting.

A process like this may be somewhat short-sighted. Reflexive outputs are also used. A reflexive output is an output that is made in the same way as a conventional output, but instead of affecting the outside world, it manipulates the hierarchy itself, so that, by making particular reflexive outputs, the system can affect the addition or removal of pattern instances in its own hierarchy. The idea of this is that, if the system can learn in general to make outputs that improve its situation, it will automatically be able to learn how to make outputs that improve its modeling ability and, therefore, its *view* of the future. Self-modification, using reflexive outputs, is therefore an important part of optimization.

More detailed description here: http://www.paul-almond.com/AI15.pdf.

5 Bibliography

Almond, P., 2010. *An Attempt to Generalize AI - Part 1: The Modeling System*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI01.pdf or http://www.paul-almond.com/AI01.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 2: Planning and Actions*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI02.pdf or http://www.paul-almond.com/AI02.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 3: Forgetting*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI03.pdf or http://www.paul-almond.com/AI03.doc [Accessed 19 August 2010].

Almond, P., 2010. An Attempt to Generalize AI - Part 4: Modeling Efficiency. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI04.pdf or http://www.paul-almond.com/AI04.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI05.pdf or http://www.paul-almond.com/AI05.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 6: Measuring Relevance*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI06.pdf or http://www.paul-almond.com/AI06.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 7: A Basic, Exploratory Relevance Process*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI07.pdf or http://www.paul-almond.com/AI07.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 8: Forgetting as Part of the Exploratory Relevance Process*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI08.pdf or http://www.paul-almond.com/AI08.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 9: Improving the Exploratory Relevance Process*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI09.pdf or http://www.paul-almond.com/AI09.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 10: Alternatives for Pattern Instance Construction*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI10.pdf or http://www.paul-almond.com/AI10.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 11: Explaining Dreaming*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI11.pdf or http://www.paul-almond.com/AI11.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 12: Pattern Relevance*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI12.pdf or http://www.paul-almond.com/AI12.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 13: Reflexive Outputs*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI13.pdf or http://www.paul-almond.com/AI13.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 14: Mind Control Speculation*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI14.pdf or http://www.paul-almond.com/AI14.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 15: A Complete Description*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI15.pdf or http://www.paul-almond.com/AI15.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 16: Speculation on Autism*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI16.pdf or http://www.paul-almond.com/AI16.doc [Accessed 19 August 2010].

Almond, P., 2010. *An Attempt to Generalize AI - Part 17: What I'm Thinking*. [Online] paul-almond.com. Available at: http://www.paul-almond.com/AI17.pdf or http://www.paul-almond.com/AI17.doc [Accessed 19 August 2010].

Metzinger, T., 2003. *Being No One: The Self-Model Theory of Subjectivity*. Cambridge (MA): MIT Press.

Metzinger, T., 2009. *The EGO Tunnel: The Science of the Mind and the Myth of the Self*. New York: Basic Books.