An Attempt to Generalize AI Part 8: Forgetting as Part of the Exploratory Relevance Process

By Paul Almond

7 May 2010

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

This is the eighth in a series of articles attempting an overview of how minds may work and how similar systems could be implemented in computers. Previous articles have described a probabilistic hierarchy based on *patterns*. A pattern has a specification describing a set, or population, of *pattern instances*, distributed throughout a hierarchy containing the pattern instances of all the patterns. Each pattern's set of pattern instances is used to obtain statistical information for probabilistic predictions. Each pattern's population of pattern instances is to be specified in a very general way, to provide a very general ontology. An *exploratory relevance process* has been described, which achieves relevance of the hierarchy by removing low-relevance pattern instances while the hierarchy "grows", so that it will tend to retreat from low-relevance regions and grow into high-relevance ones. The exploratory relevance process uses a *relevance measurement process*, which back-propagates relevance through the hierarchy. The exploratory relevance process, as previously described, does not provide forgetting – the removal of obsolete pattern instances which are no longer having any useful effect on the hierarchy. A forgetting process was described earlier, but that description now needs revising after the change to a completely probabilistic hierarchy. In this article, the relevance measurement process is modified to take account of obsolescence of pattern instances, so that they will tend to be assigned low relevance and the exploratory relevance process will remove them. Forgetting is reduced to being merely a special case of the exploratory relevance process.

Table of Contents

List of Abbreviations

AI	artificial intelligence
BERP	basic, exploratory relevance process
EFS	evaluation function score
ERP	exploratory relevance process
RMP	relevance measurement process

1 Introduction

This article is the eight in a series about artificial intelligence (AI) and how our own minds might work. The first article, *An Attempt to Generalize AI - Part 1: The Modeling System*, is available at http://www.paul-almond.com/AI01.pdf.[1] The second article, *An Attempt to Generalize AI - Part 2: Planning and Actions*, is at http://www/paul-almond.com/AI02.pdf.[2] The third article, *An Attempt to Generalize AI - Part 3: Forgetting*, is at http://www.paul-almond.com/AI03.pdf.[3]

These three articles described a hierarchy based on *patterns*, which are sets of *pattern instances*, and were intended to give an idea of how humans may model the world, plan actions and discard information from the model when it is no longer useful. The fourth article, *An Attempt to Generalize AI - Part 4: Modeling Efficiency*, which is at http://www.paul-almond.com/AI04.pdf, suggested that pattern instances should be allowed to have *incompletely specified pattern inputs.*[4] The hierarchy might contain information about some of the pattern inputs to a pattern instance, while others, for practical purposes, would be non-existent. This would allow the removal of pattern instances from the hierarchy without having to remove what was "above" them, and it could simplify the connection of new pattern instances into the hierarchy. This would be done so that the hierarchy could be "pruned" by some process seeking to maximize its relevance. For this to be practical, the hierarchy needed to be *completely* probabilistic: It had previously relied on the special case of pattern instances known about with certainty (previously referred to as "fixed" pattern instances) and this reliance needed removing. This issue was dealt with in the fifth article of this series, *An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy*, which is at http://www.paul-almond.com/AI05.pdf.[5]

That made a process to provide relevance in the hierarchy feasible. A relevance measurement process (RMP) was needed first. The sixth article, *An Attempt to Generalize AI – Part 6: Measuring Relevance*, which is at http://www.paul-almond.com/AI06.pdf, described a back-propagation process for measuring relevance in

[1] Almond, P. (2010). An Attempt to Generalize AI - Part 1: The Modeling System. *paul-almond.com*. http://www.paul-almond.com/AI01.pdf. (Also available at http://www.paul-almond.com/AI01.doc.)
[2] Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. http://www.paul-almond.com/AI02.pdf. (Also available at http://www.paul-almond.com/AI02.doc.)
[3] Almond, P. (2010). An Attempt to Generalize AI - Part 3: Forgetting. *paul-almond.com*. http://www.paul-almond.com/AI03.pdf. (Also available at http://www.paul-almond.com/AI03.doc.)
[4] Almond, P. (2010). An Attempt to Generalize AI - Part 4: Modeling Efficiency. *paul-almond.com*. http://www.paul-almond.com/AI04.pdf. (Also available at http://www.paul-almond.com/AI04.doc.)
[5] Almond, P. (2010). An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy. *paul-almond.com*. http://www.paul-almond.com/AI05.pdf. (Also available at http://www.paul-almond.com/AI05.doc.)

the hierarchy.[6] The problem is made tractable by the way in which the *action selection process*, described in the second article, *An Attempt to Generalize AI – Part 2: Planning and Actions*, works.[7] The action selection process, which is used to *drive* the system's behavior in a particular direction, relies on predictions of pattern instances which will be used for future input of a continually computed evaluation function score (EFS). The requirement for such specific predictions from the hierarchy provides a natural end-point – what the hierarchy is expected to produce – and a clear goal: The hierarchy needs to minimize the uncertainty in these particular pattern instances. This tells us what is most relevant and allows us to describe a process of back-propagation of relevance from these pattern instances, through the rest of the hierarchy, allowing the relevance of every part of the hierarchy to be measured.

The seventh article, *An Attempt to Generalize AI – Part 7: A Basic, Exploratory Relevance Process*, which is at http://www.paul-almond.com/AI07.pdf, described the *basic exploratory relevance process* (BERP), which uses this measuring process to direct the growth and pruning of the hierarchy.[8] This will be relevant in this article, which is about *forgetting*.

In the third article of this series, *An Attempt to Generalize AI - Part 3: Forgetting*, I discussed the *basic forgetting procedure*; however, this is not workable after the change to a completely probabilistic hierarchy in *An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy*. The general idea of the forgetting process – removing pattern instances when they have ceased to have an effect on others in the hierarchy is still valid, but it needs transferring to the probabilistic hierarchy. Forgetting is about removing pattern instances that are *obsolete*, and the BERP, and similar processes that might be used to provide relevance, are about removing pattern instances that are *irrelevant*. Could these processes be related to each other? In *An Attempt to Generalize AI – Part 7: A Basic, Exploratory Relevance Process*, I stated that the proposed process for relevance did not, in itself, provide a process for forgetting, because forgetting was needed to deal with obsolescence of pattern instances, which was not necessarily the same as lack of relevance, with relevance defined in the way that it was. This article will be about changing that. The relevance measurement process (RMP) will be modified so that relevance values take account of obsolescence, causing the BERP, or any other exploratory relevance process (ERP), to perform forgetting. Forgetting will become merely a special case of the processing used to provide relevance in the hierarchy.

---

[6] Almond, P. (2010). An Attempt to Generalize AI - Part 6: Measuring Relevance. *paul-almond.com*. http://www.paul-almond.com/AI06.pdf. (Also available at http://www.paul-almond.com/AI06.doc.)
[7] Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. http://www.paul-almond.com/AI02.pdf. (Also available at http://www.paul-almond.com/AI02.doc.)
[8] Almond, P. (2010). An Attempt to Generalize AI - Part 7: A Basic, Exploratory Relevance Process. *paul-almond.com*. http://www.paul-almond.com/AI07.pdf. (Also available at http://www.paul-almond.com/AI07.doc.)

2 A Note on Probability Values

As I have done previously,[9] in this article, I will assume that each pattern instance in the actual hierarchy has a single probability, as would be the case for a system in which pattern instances in the conceptual hierarchy had 0/1 values. What will be discussed here would equally well apply in a system in which each pattern instance was described by multiple probability values.

[9] Almond, P. (2010). An Attempt to Generalize AI - Part 6: Measuring Relevance. *paul-almond.com*. http://www.paul-almond.com/AI06.pdf. p. 6. (Also available at http://www.paul-almond.com/AI06.doc.)

3 Relevance Propagation and Degree of Uncertainty

3.1 The Previous Forgetting Method

In *An Attempt to Generalize AI - Part 3: Forgetting*, the basic forgetting procedure was described.[10] When any pattern instance has become "fixed" (meaning that its probability value is either 0 or 1 and its actual value in the conceptual hierarchy is known with certainty), it is only retained as long as there is at least one pattern instance using it as a pattern input which itself does not yet have a known pattern output. When this condition is no longer met, the pattern instance is erased.

Basing a forgetting process on the special case of pattern instances with no uncertainty – "fixed" pattern instances – like this is no longer tenable, following the change to a completely probabilistic hierarchy in *An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy*.[11] The special case of a pattern instance with no uncertainty can still be of interest, however, when considering how forgetting should work.

3.2 The Relevance Measurement Process

In the relevance measurement process (RMP) described in *An Attempt to Generalize AI – Part 6: Measuring Relevance*, particular pattern instances corresponding to future inputs/outputs are regarded as relevant due to their use in the action selection process.[12] Relevance is back-propagated from these pattern instances. A pattern instance propagates a total amount of relevance equal to its own relevance back to those pattern instances that have directly affected its probability value, and it is shared out amongst them in proportion to the amount of effect that they have had on reducing its uncertainty.

We need to think about what all this is supposed to achieve. Whenever the RMP computes relevance, this is based on the probability values that exist in the hierarchy at the time. This means that when a relevance value is obtained for a pattern instance it is not necessarily the relevance that the pattern instance will have *in future*, but is actually the relevance that it has *already had* in the propagation of probability values that has already occurred through the hierarchy. This can be justified in two ways, as follows.

---

[10] Almond, P. (2010). An Attempt to Generalize AI - Part 3: Forgetting. *paul-almond.com*. http://www.paul-almond.com/AI03.pdf. p. 15. (Also available at http://www.paul-almond.com/AI03.doc.)
[11] Almond, P. (2010). An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy. *paul-almond.com*. http://www.paul-almond.com/AI05.pdf. (Also available at http://www.paul-almond.com/AI05.doc.)
[12] Almond, P. (2010). An Attempt to Generalize AI - Part 6: Measuring Relevance. *paul-almond.com*. http://www.paul-almond.com/AI06.pdf. (Also available at http://www.paul-almond.com/AI06.doc.)

- We should expect the relevance of pattern instances to change gradually. Therefore, a pattern instance's relevance to the current situation should be a good indication of its likely relevance in the immediate future.
- When the BERP, previously described in *An Attempt to Generalize AI – Part 7: A Basic, Exploratory Relevance Process*,[13] or some other ERP, removes pattern instances from the hierarchy, those remaining are not just being left because they may be directly useful. New pattern instances are connected to existing ones in the hierarchy, so the pattern instances selected to remain in the hierarchy are also providing a structure from which the hierarchy will grow, and the new pattern instances are more likely to be highly relevant in the future if the ones to which they are being connected are already known to have high relevance – even if these get removed shortly afterwards.

Both of these show why relevance measurements based on propagation that has already occurred can be a useful guide to which pattern instances will be useful in the future, even if the RMP does not provide a completely accurate prediction of future relevance, but this does not change the fact that we should still want the relevance values to indicate future relevance *as accurately as possible*.

We will now consider the RMP in relation to the special case of a pattern instance for which the value is known with certainty – what was previously known as a "fixed" pattern instance.

3.3 The Special Case of a Pattern Instance with No Uncertainty

Suppose in a computer's actual hierarchy there is a single pattern instance, X, in the hierarchy for which, due to the probabilistic information being propagated to it directly by logic application or statistics application from other pattern instances, the probability value is 0 or 1. This pattern instance is known about with certainty. It is now known that the corresponding pattern instance in the conceptual hierarchy has an actual pattern output value (not a probability) of 1. X's probability can never change again: If it did change again, this would imply that the hierarchy's assignment of a probability value indicating certainty to it had been incorrect.

X would have been assigned this probability due to propagation of probabilistic information directly from other pattern instances. In the RMP, X will back-propagate a total amount of relevance equal to its own relevance to these pattern instances, and it will be shared out among them according to the contribution each has made to reducing its uncertainty. A contribution to reducing X's uncertainty has clearly been made,

[13] Almond, P. (2010). An Attempt to Generalize AI - Part 7: A Basic, Exploratory Relevance Process. *paul-almond.com*. http://www.paul-almond.com/AI07.pdf. pp. 13-14. (Also available at http://www.paul-almond.com/AI07.doc.)

because X's probability has gone to 0 or 1. X will therefore propagate relevance to the pattern instances which directly affected its probability and reduced the uncertainty in it.

The reasoning behind this should be that the pattern instances receiving the relevance, because they are already known to have an important role in reducing uncertainty in X, should be expected to reduce it in future. We might also take the view that pattern instances that we might connect to them in future could reasonably be expected to have similar relevance. However we view this, we should still want to propagate an amount of relevance to each pattern instance, from X, that reflects the future expectations of relevance as accurately as possible.

So, what are our future expectations of relevance here? Relevance is being propagated from X to other pattern instances, but X's probability of 0 or 1 means that X is already known about with certainty. These other pattern instances can never tell us anything useful about X in future because we already know everything! The pattern instances which have received relevance from X will have no role to play in reducing X's uncertainty in future: X has no uncertainty left to reduce. In reality, we could just set X's probability to 0 or 1, and *back-propagate no relevance from X*.

The approach just described is the same as the basic forgetting procedure from the third article, except that it is now just part of the processing to provide relevance in the hierarchy: It is now part of the BERP or some other ERP. The basic forgetting procedure cannot be used in its current form, with the completely probabilistic hierarchy, because hardly any pattern instances will be known about with certainty. This does not change the fact that the approach just described, although impractical due to being usable only in uncommon special cases, would still be valid. This suggests that a generalization of this could be used, and to see how this would work we will now consider back-propagation of relevance from pattern instances which are known about with *almost* complete certainty.

3.4 A Pattern Instance with *Almost* Complete Certainty

Suppose there is a situation *almost* the same as the one just discussed, in 3.3. In the actual hierarchy there is a single pattern instance, X, in the hierarchy for which the probability value is now *almost* 0 or 1; for example, it may be 0.001 or 0.999. This pattern instance is known about with almost complete certainty. In the previous situation, in 3.3, the BERP or other ERP was back-propagating relevance to the pattern instances that had influenced X, but really no relevance should have been back-propagated, because the certainty in X made the other pattern instances redundant. Here, the situation is *almost* the same. If the pattern instances in the previous situation should have been receiving no relevance at all from X, then it makes no sense to say that the pattern instances in this almost identical situation should receive the normal amount of relevance: They will be telling us about a pattern instance about which we already know *almost* everything anyway, so they should be receiving *hardly any* relevance.

For the previous situation, in 3.3, I mentioned that when we knew about a pattern instance with certainty we could just permanently set its probability to 0 or 1 and not rely on any further propagation. Here, things need a bit more consideration, but I will return to this issue shortly.

Now that we have considered the case of pattern instances which are known about with almost complete certainty, we can go to the general case of a pattern instance known about with *any* degree of certainty.

3.5 A Pattern Instance with *Any* Degree of Certainty

As just discussed, in 3.4, when a pattern instance is known about with *almost* complete certainty, it should propagate hardly any relevance back because the value of any more probabilistic information that it receives in the future is reduced. There is no dividing line, however, at which the degree of certainty becomes "almost complete". If we can make a case for reducing relevance back-propagation from a pattern instance known about with almost complete certainty, as just discussed in 3.4, it follows that the total amount of reference propagated back from some pattern instance, X, should be affected by the degree of certainty for any probability that X could have, the full amount of X's relevance being available for propagation only when nothing is known about X (when its probability value is 0.5), and the total amount of relevance being propagated being reduced as the degree of certainty about X increases: As X's probability gets closer to 0 or 1, the total amount of relevance that it propagates back should be reduced.

This now gives us the basis for an approach to forgetting that can replace the previous forgetting process in *An Attempt to Generalize AI - Part 3: Forgetting*,[14] and which is just part of the processing that the BERP, or some other ERP, performs to maintain relevance in the hierarchy.

[14] Almond, P. (2010). An Attempt to Generalize AI - Part 3: Forgetting. *paul-almond.com*. http://www.paul-almond.com/AI03.pdf. p. 15. (Also available at http://www.paul-almond.com/AI03.doc.)

4 Modifying the Relevance Measurement Process to Take Account of Degree of Certainty

4.1 The Basic Modification

To provide forgetting, the RMP, previously described in *An Attempt to Generalize AI – Part 6: Measuring Relevance*,[15] is modified as follows.

4.1.1 Total Amount of Relevance

When relevance is propagated from some pattern instance, X, to those pattern instances which have directly affected its probability value,[16] the *maximum* total amount of relevance that will be propagated is the same as X's own relevance; however this is adjusted to take account of the degree of certainty in X. The closer X is to being known about with complete certainty – the closer its probability value is to 0 or 1 – then the lower is the total amount of relevance that X back-propagates.

- For a pattern instance with a **probability of 0.5**, the total amount of relevance that is back-propagated is equal to X's relevance.
- For a pattern instance with a **probability of 0 or 1**, no relevance is back-propagated.
- For a pattern instance with **some other probability value between these values**, the amount of relevance back-propagated is some fraction of X's relevance, the amount decreasing as the probability gets closer to 0 or 1.

4.1.2 Allocation of Relevance

The relevance that is being back propagated is shared out among the individual pattern instances which are considered relevant to X, each receiving a proportion of the relevance in accordance with the extent to which it reduces uncertainty in X. This is done as in the previous description of the RMP: The only difference is that the *total amount* of relevance available for propagation is modified as just described, in 4.1.1.

---

[15] Almond, P. (2010). An Attempt to Generalize AI - Part 6: Measuring Relevance. *paul-almond.com*. http://www.paul-almond.com/AI06.pdf. pp. 16-19. (Also available at http://www.paul-almond.com/AI06.doc.)

[16] or degree of uncertainty in the case of more than one such value

4.2 Preventing Loss of Information

4.2.1 The Issue of Loss of Information

The issue could be raised that the approach to forgetting just described, in 4.1, results in the loss of information from the hierarchy.

The purpose of the modification to the RMP just described is to reduce the relevance values of pattern instances that are obsolete, so that the BERP, or whatever ERP is in use, will tend to remove them from the hierarchy. If a pattern instance is known about with a high degree of certainty, little relevance will be back-propagated from it, and this is likely to cause the BERP, or some other ERP, to remove the pattern instances that are causing the high degree of certainty in the first place. With these pattern instances gone, if we propagate all the probabilistic information through from the bottom-level again, would this not mean that the previous degree of certainty will not be achieved? By removing obsolete pattern instances, are we not removing the very thing that is providing the degree of certainty that justified their removal, and do we not lose that degree of certainty without them?

4.2.2 The Solution

When discussing the special case of a pattern instance about which there is complete certainty, in Section 3.3: The Special Case of a Pattern Instance with No Uncertainty, on page 8, I stated that we could just set the pattern instance permanently with the 0 or 1 probability. We might try this kind of approach here, with the general situation; however, there is a better solution, as follows.

When a pattern instance is removed, we do not necessarily remove it completely, immediately. Instead, a pattern instance, on “removal” can become a *ghost pattern instance*. A ghost pattern instance has its probability value retained. Probabilistic information can no longer be propagated into a ghost pattern instance: A ghost pattern instance’s probability value can never be changed. It can no longer propagate any relevance. However, it can still affect other pattern instances through propagation of probabilistic information in logic application or statistics application.

If the BERP, or some other ERP, makes the decision to remove a pattern instance, and the pattern instance is not directly affecting any other pattern instance through propagation of probabilistic information (and other ghost pattern instances do not count – they cannot be affected, anyway), the pattern instance is removed conventionally, and all information about it is removed from the hierarchy.

If the BERP, or some other ERP, makes the decision to remove a pattern instance, and the pattern instance *is* directly affecting at least one other pattern instance (and it needs to be one that is not a ghost pattern instance), then a full removal does not occur immediately. Instead, the pattern instance becomes a ghost pattern instance, with its probability value remaining in the hierarchy, still able to influence other pattern instances through propagation.

If this situation changes, later, so that the ghost pattern instance is no longer directly affecting another pattern instance through propagation, the removal is completed and the probability value is removed as well: Nothing now remains of the pattern instance.

It may seem that the removal of pattern instances will be made much less effective, as ghost pattern instances would still need dealing with as probability values, meaning that computing resources have not been completely freed up by the removal. This would not cause severe problems. The BERP, or ERP, would be removing huge numbers of pattern instances. A pattern instance may persist as a ghost for a while, because it is directly affecting some pattern instance that is still active in the hierarchy, but it would not be propagating any relevance back to *other* pattern instances that are not directly affecting active pattern instances, making it more likely that they *will* be removed. Also, because a ghost pattern instance cannot change, another pattern instance cannot linger as a ghost by affecting it. A pattern instance can only linger as a ghost, after being removed, if it is logically "near" the actual hierarchy, so a relatively small number of pattern instances may be lingering after removal near the edges of the actual hierarchy at any time, but the ones further out will tend to have been removed *properly* due to lack of relevance.

4.2.3 Ghost pattern instances affect relevance processing in general.

There is no special forgetting process, so ghost pattern instances are not just involved with forgetting: The removal of pattern instances as part of the general processing to provide relevance in the BERP or ERP is affected by them. *All* pattern instance removal as part of the processing to provide relevance in the hierarchy now involves pattern instances remaining as ghost pattern instances when they are still affecting active pattern instances by probabilistic propagation. When pattern instances have incompletely specified pattern inputs, the "missing" pattern inputs would now connect to ghost pattern inputs. A ghost pattern instance is really a representation of a pattern instance in the conceptual hierarchy.

5 Issues

5.1 What if we just remember the probability?

Some readers may wonder why a pattern instance should be back-propagating *any* relevance to the pattern instances that have determined its probability, unless they are *continually* reducing its uncertainty. Could we not just as well store the previous probability that the pattern instances propagated, and does this not mean that any further propagation that failed to reduce uncertainty was redundant, and therefore irrelevant?

For example, suppose after probabilistic propagation through the hierarchy, a pattern instance, X, has a probability of 0.2. The next time any propagation occurs through the hierarchy, X's probability is still 0.2. We already knew that X's probability was 0.2: In the absence of any information from propagation, we would have just assigned it the previous probability value anyway. Given this, should we not say that nothing, beyond its own probability is relevant to X – unless some pattern instances can provide a reduction in uncertainty?

I would not agree with this. The first thing to note is that the hierarchy, although describing a temporal situation, is not a temporal object. In the conceptual hierarchy, which the actual hierarchy is modeling, each pattern instance has a pattern output value and there is no change: The concept of time does not even mean anything. In the actual, hierarchy, in a computer, the values of some of the pattern instances are unknown, because they depend on future inputs/outputs that have not yet occurred, and so have to be described probabilistically. As inputs/outputs occur, more information about the hierarchy becomes available, and probabilities in the actual hierarchy change to reflect this.

This means that if some pattern instances recently helped to reduce the uncertainty in X, but the last time that propagation occurred there was no change, it does not follow that those pattern instances are irrelevant to X: The probability that X has was still determined by those pattern instances. If we took such an approach it would be hard for any pattern instance to have any relevance for any appreciable time: As soon as some pattern instances generated some information in another pattern instance, we could simply refer to that information and rule that, unless these pattern instances had anything else to offer *immediately*, they were irrelevant. This would be ignoring the fact that those pattern instances, by being responsible for X's current probability, might be expected to be particularly relevant to any change in this probability in future – and also that they indicate regions of the hierarchy where we should think it is particularly likely that new, high relevance pattern instances can be added.

It should be noted that the rationale for limiting the amount of relevance back-propagated from a pattern instance based on the degree of certainty, in Section 3.5: A

Pattern Instance with *Any* Degree of Certainty, on page 10, is entirely different to the kind of reasoning being discussed here. When relevance is limited like that, it is because the scope for pattern instances to do anything useful is limited in future by the degree of certainty that already exists. A situation in which pattern instances have made some useful contribution and in which there is still scope for them to offer more is not the same.

5.2 Isn't this inconsistent?

5.2.1 Two Different Standards

There may *seem* to be inconsistency in the way relevance is back-propagated from a pattern instance, as described in Section 4.1: The Basic Modification, on page 11.

1. The total relevance back-propagated from a pattern instance, X, *decreases* as the certainty in the pattern instance *increases*, implying that the pattern instances receiving the relevance are *penalized* for achieving a high degree of certainty in X, and this may seem to imply that we *do not want* a high degree of certainty in X.
2. The proportion of the total relevance back-propagated from X to an individual pattern instance increases as the contribution of that pattern instance to achieving a high degree of certainty in X increases, implying that the pattern instances receiving the relevance are *rewarded* for achieving a high degree of certainty in X, and this seems to imply that we *want* a high degree of certainty in X.

5.2.2 Why there is No Inconsistency

There is no inconsistency here, because each of the cases just described, in 5.2.1, is actually referring to a different thing. One refers to a group of individual pattern instances and the other refers to the group.

- In (1) it is the *group* of pattern instances directly affecting X that has its reward reduced for achieving a high degree of certainty in X, by having the total amount of relevance reduced as this degree of certainty increases.
- In (2) it is *individual* pattern instances that are rewarded for helping to achieve a high degree of certainty in X, by giving each a proportion of the total relevance given to the group that increases as the degree of certainty that it achieves increases.

There is nothing inconsistent about this. As the certainty in X increases, the benefit of any propagation into X is reduced, so less total relevance is back-propagated; however, we still want to share out what relevance is back-propagated according to merit, with each pattern instance receiving a share according to the contribution that it makes.

5.2.3 An Analogy

As an analogy, suppose we are manufacturing some product that we want to improve, so that it will be more commercially successful. Maybe we want more people to buy it. Maybe we want it to be more reliable, so we have fewer returns. Maybe we want it cheaper to manufacture. Whatever the specific objective, the general idea is that we want the product to earn us more money.

We offer reward money to our employees for thinking of ways to improve the product. Each month, a prize fund is made available, and the money is given to employees who made useful suggestions for improving the product that month. The proportion of that month's prize money that any single employee receives depends on how good his/her idea is: Employees whose improvements to the product earn us a lot of money receive a lot of that month's prize money as a reward.

As time passes, the product becomes harder to improve. The easier improvements have been made earlier on, and it is hard for anyone to think up really significant, new ones. The product design is becoming *optimized*. The returns from making further improvements start to diminish: The total value of the ideas we receive each month tends to be less than it was when we started. We reduce the prize fund for each month, so less money will be paid out each month. What we are prepared to give to the group is reduced to reflect the reduced scope for improving the product. However, this does not change the fact that we are still going to share that money among individual employees according to what each has contributed.

5.2.4 Relevance is about allocation of computing resources.

The analogy just given, in 5.2.3, should have shown that, just as there is no inconsistency in what we are *doing* here, there is no inconsistency in what we *want* either. What we want is *always* a high degree of certainty in pattern instances. When we reduce the total amount of relevance back-propagated from some pattern instance, dependent on its degree of certainty, this does not mean that we do not want a high degree of certainty. All it means is that the degree of certainty already there limits the amount of extra certainty that we can get in future, so we are limiting the computing resources that will be spent on trying to get that certainty so that more are available elsewhere. The amount of relevance being provided is what ultimately determines resource allocation, by controlling the density of the hierarchy.

6 Bottom-Level Pattern Instances and Forgetting

The issue of bottom-level pattern instances still needs consideration with regard to any ERP that could cause their removal, as was stated in the previous article, *An Attempt to Generalize AI – Part 7: A Basic, Exploratory Relevance Process*, as follows.

"The issue of bottom-level pattern instances needs some consideration. These will tend to be removed by any forgetting process, but some will tend to be removed in the existing pruning by the BERP. Normally, when pattern instances are removed, they can be added again if needed: The hierarchy can be reduced in some region where it is insufficiently relevant, and later regrown if that region becomes relevant again. The information in the hierarchy is based on bottom-level pattern instances corresponding to previous inputs/outputs. If these are removed, the information in them is lost from the hierarchy. It may be advisable to make the conditions for removal of such pattern instances more demanding than for others." [17]

[17] Almond, P. (2010). An Attempt to Generalize AI - Part 7: A Basic, Exploratory Relevance Process. *paul-almond.com*. http://www.paul-almond.com/AI07.pdf. p. 17.
(Also available at http://www.paul-almond.com/AI07.doc.)

7 Conclusion

Previously, I stated that the basic, exploratory relevance process (BERP), or any other exploratory relevance process (ERP), would not prune the hierarchy to provide forgetting, given the way that relevance was then defined.[18] This issue has now been dealt with. Instead of the separate forgetting process previously described,[19] forgetting is now dealt with as part of the pruning of the hierarchy by part of the general process to provide relevance: the BERP or whatever ERP is in use.

Forgetting has been included in the processing for relevance by modifying the way in which relevance is computed for pattern instances. The previous forgetting process relied on pattern instances which were known about with certainty: Pattern instances which only affected these were considered obsolete and removed from the hierarchy. This kind of idea is still used, but it is generalized to deal with the completely probabilistic nature of the hierarchy as it is currently described, and the expected rareness of pattern instances about which there is complete knowledge. The processing to provide relevance – the BERP [20] or some other ERP – takes account of relevance values assigned to pattern instances by the relevance measurement process (RMP)[21]. The RMP works by back-propagating relevance from pattern instances to those on which their probability values are strongly dependent. The modification of this to provide forgetting is as follows.

When relevance is propagated from some pattern instance, X, to those pattern instances which have directly affected its probability value, the *maximum* total amount of relevance that will be propagated is the same as X's own relevance; however this is adjusted to take account of the degree of certainty in X. The closer X is to being known about with complete certainty – the closer its probability value is to 0 or 1 – then the lower is the total amount of relevance that X back-propagates.

The idea behind this is that the more that is already known about a pattern instance, the less the scope is for finding out more about it in future, and so the lower the value is of any computation performed for this purpose.

---

[18] Almond, P. (2010). An Attempt to Generalize AI - Part 7: A Basic, Exploratory Relevance Process. *paul-almond.com*. http://www.paul-almond.com/AI07.pdf. p. 17. (Also available at http://www.paul-almond.com/AI07.doc.)
[19] Almond, P. (2010). An Attempt to Generalize AI - Part 3: Forgetting. *paul-almond.com*. http://www.paul-almond.com/AI03.pdf. p. 15. (Also available at http://www.paul-almond.com/AI03.doc.)
[20] Almond, P. (2010). An Attempt to Generalize AI - Part 7: A Basic, Exploratory Relevance Process. *paul-almond.com*. http://www.paul-almond.com/AI07.pdf. pp. 13-14. (Also available at http://www.paul-almond.com/AI07.doc.)
[21] Almond, P. (2010). An Attempt to Generalize AI - Part 6: Measuring Relevance. *paul-almond.com*. http://www.paul-almond.com/AI06.pdf. pp. 16-19. (Also available at http://www.paul-almond.com/AI06.doc.)

As has been stated previously, we need to be careful about removing bottom-level pattern instances from the hierarchy.[22] This would mean loss of information. Bottom-level pattern instances will need removing, but the standard to be met for this to be allowed may need to be higher to ensure that they really are obsolete first.

The removal of obsolete pattern instances in "forgetting" would cause loss of information. A solution to this is provided by ghost pattern instances. Ghost pattern instances are left behind when pattern instances are "removed" while they are still affecting other pattern instances in the hierarchy by probabilistic propagation. A ghost pattern instance can affect other pattern instances by probabilistic propagation, but does not back-propagate relevance, and cannot cause another pattern instance that used to affect it to remain as a ghost pattern instance. This means that ghost pattern instances would only tend to linger around the "edges" of the active part of the hierarchy.

The idea that *forgetting* is mainly part of the processing to maintain *relevance* has implications for a view of how forgetting works in humans. Some people might regard the brain's thoughts as distinct from its memory, but this series of articles takes a different view in which there is just the hierarchy. It suggests that forgetting is much closer to the brain simply "withdrawing" its hierarchy from low-relevance regions than might often be thought.

---

[22] Almond, P. (2010). An Attempt to Generalize AI - Part 7: A Basic, Exploratory Relevance Process. *paul-almond.com*. http://www.paul-almond.com/AI07.pdf. p. 17.
(Also available at http://www.paul-almond.com/AI07.doc.)

8 Bibliography

Almond, P. (2010). An Attempt to Generalize AI - Part 1: The Modeling System. *paul-almond.com*. Retrieved 7 May 2010 from http://www.paul-almond.com/AI01.pdf.
(Also available at http://www.paul-almond.com/AI01.doc.)

Almond, P. (2010). An Attempt to Generalize AI - Part 2: Planning and Actions. *paul-almond.com*. Retrieved 7 May 2010 from http://www.paul-almond.com/AI02.pdf.
(Also available at http://www.paul-almond.com/AI02.doc.)

Almond, P. (2010). An Attempt to Generalize AI - Part 3: Forgetting. *paul-almond.com*. Retrieved 7 May 2010 from http://www.paul-almond.com/AI03.pdf.
(Also available at http://www.paul-almond.com/AI03.doc.)

Ibid. p. 15.

Almond, P. (2010). An Attempt to Generalize AI - Part 4: Modeling Efficiency. *paul-almond.com*. Retrieved 7 May 2010 from http://www.paul-almond.com/AI04.pdf.
(Also available at http://www.paul-almond.com/AI04.doc.)

Almond, P. (2010). An Attempt to Generalize AI - Part 5: A Completely Probabilistic Hierarchy. *paul-almond.com*. Retrieved 7 May 2010 from http://www.paul-almond.com/AI05.pdf.
(Also available at http://www.paul-almond.com/AI05.doc.)

Almond, P. (2010). An Attempt to Generalize AI - Part 6: Measuring Relevance. *paul-almond.com*. Retrieved 7 May 2010 from http://www.paul-almond.com/AI06.pdf.
(Also available at http://www.paul-almond.com/AI06.doc.)

Ibid. p. 6.

Ibid. pp. 16-19.

Almond, P. (2010). An Attempt to Generalize AI - Part 7: A Basic, Exploratory Relevance Process. *paul-almond.com*. Retrieved 7 May 2010 from http://www.paul-almond.com/AI07.pdf.
(Also available at http://www.paul-almond.com/AI07.doc.)

Ibid pp. 13-14.

Ibid. p. 16.

Ibid. p. 17.