A NON-DETERMINISTIC APPROACH TO ANALOGY, INVOLVING THE ISING MODEL OF FERROMAGNETISM

Douglas R. Hofstadter

Fluid Analogies Research Group, Perry Building
330 Packard Road, Ann Arbor, Michigan 48104

ABSTRACT

A close analysis of several abstract analogies reveals the critical role played by directed links created in the act of perceiving the structures in the analogy. These directed links, having bi-stable orientation properties, are similar to the bi-stable spins of the Ising model of ferromagnetism. The similarity is enhanced by the fact that the links' orientations are not deterministic but stochastic, and the degree of order and disorder in the system is regulated by a formal parameter playing a role analogous to that of temperature in the Ising model. The paper is concerned principally with showing how temperature-controlled flipping of the directed links allows "perceptual Bloch domains" to emerge, thus facilitating discovery of subtle analogies.

In the making of an analogy between two situations, a critical ingredient is how those situations are framed in terms of known concepts. Framing a situation in terms of concepts is much like visual perception of a scene, in which the goal is to attach numerous labels ("chair", "elephant", "Vesuvius") to regions of the visual field. The difference is that a situation is generally abstract rather than visual, and consequently the labels to be attached to parts of it are usually at a higher level of abstraction than those in visual perception.

Our work on analogies[1,2,3] involves highly idealized <u>situations</u> represented by strings of letters of the alphabet. An <u>event</u> in such a situation is a change in the original string. Thus, a typical event would be the changing of the string **abc** into the string **abd**. In fact, we take this event as our <u>prototype event.</u> Our goal is for our computer program "Copycat" to be able to make numerous interesting analogies with that event. For instance, if **abc -> abd**, what analogous event should happen in the <u>target situation</u> **ijkl**? There are several conceivable answers, but the most satisfying one for the vast majority of people is: **ijkl -> ijkm**. There seems to have been extracted a <u>rule</u> from the prototype event: *"Replace the rightmost letter by its alphabetic successor".* It seems that this rule is then applied to the target situation, yielding the answer.

On closer analysis, one sees that things are not quite that simple. For instance, consider a different target situation, **iijjkkll**. If the rule cited above were simply applied, as is, to this new target, one would get the following result: **iijjkkll -> iijjkklm**, which is contrary to most people's preference on esthetic grounds, which are of major importance in analogy-making. Most people strongly prefer **iijjkkmm** as the outcome. The intuitive explanation for this answer is that generally, the rule should be interpreted a little loosely, and that here in particular, the phrase *"rightmost letter"* should be interpreted according to its new context. In the target situation **iijjkkll**, it seems that doubled letters play the role that single letters played in the prototype, so that the adjusted rule would say something like this: *"Replace the rightmost <u>doubled-letter</u> by its alphabetic successor".* Of course, that rule yields the appropriate answer.

We call the process whereby a rule is modified according to its new context the <u>translation</u> of the rule. Translation is a key process in the operation of the Copycat program. It is important to see translation occur in a number of different ways, so consider the following new target situation: **srqp**. Application of the "raw" rule (i.e., the untranslated rule) to this target would yield the answer **srqq**, an answer that few people find appealing. Once again, therefore, it seems that translation is called for. However, here we find an interesting split among people. Some prefer the answer **trqp**, while others prefer **srqo**. These answers reveal what translated rules are being created and utilized. In the case of answer **trqp**, it is clear that the operation *replacement by alphabetic successor* was carried out on the *leftmost* letter, rather than the rightmost. On the other hand, in the case of answer **srqo**, the site of the operation remained fixed, but now the operation itself was adjusted into *replacement by alphabet <u>predecessor</u>.*

We have now seen three distinct translations of the raw rule *"Replace the rightmost letter by its alphabetic successor".* They are summarized below:

Target situation: **iijjkkll**.

Translated rule:
Replace the rightmost doubled-letter by its alphabetic successor.

Resultant answer: **iijjkkmm**.

Target situation: **srqp**.

Translated rule:
Replace the leftmost letter by its alphabetic successor.

Resultant answer: **trqp**.

Target situation: **srqp**.

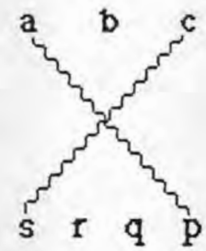
Translated rule:
Replace the rightmost letter by its alphabetic predecessor.

Resultant answer: **srqo**.

It seems that subtle contextual pressures must be applied to a rule so that it can "flex" or adapt itself to a new situation. In particular, some parts of the rule will stay constant while other parts of it will be modified. The crux of our research project is to determine how such modifications are brought about by the constellation of pressures that arise when the target situation is compared to the prototype situation. But for such a comparison to be made, there must already exist perceptions of the two situations. It is the process of situation perception that we are concerned with in this paper, since that process chronologically precedes and logically underlies the critical process of rule translation.

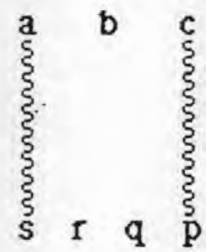
Because the target situation **srqp** has two distinct answers that appeal to people, it follows that there must be two distinct perceptions of that situation that cause distinct constellations of mental pressures, ultimately leading to the bifurcation in opinions about rule translation. Let us try to characterize the problem. We are comparing prototype situation **abc** with target situation **srqp**. In one view, the letter **s** plays the role of the **c** (i.e., it is the site of change), while in the other view, the letter **p** plays the role of the **c**. Thus, there are really two different mappings going on that give rise to the two different answers:

Mapping 1:



(diagonal mapping)

Mapping 2:



(vertical mapping)

The wiggly lines connecting counterpart entities in a given mapping are called <u>bridges</u>. Bridges play an extremely central role in the Copycat program, for it is they (or more precisely, their <u>pylons</u>) that carry all information about rule translation. However, precisely how bridges and pylons accomplish this need not concern us here. Intuitively, it is clear that the establishment of credible counterparts between situations tells a great deal about how to adapt a statement about the first situation to the second situation.

Clearly the diagonal mapping is the one that gives rise to the answer **trqp**, for in it the **s** is the counterpart of the **c** and the roles of concepts *left* and *right* have been reversed. It is relatively easy to see how these bridges would suggest substituting "*leftmost*" for "*rightmost*" in the rule, thus accounting for the rule translation that leads to the answer **trqp**.

In the vertical mapping, the **p** is the counterpart of the **c** and left and right have not been reversed. Now, these bridges are supposed to give rise to the answer **srqo**. To do so, they would have to suggest substituting "*predecessor*" for "*successor*", for if that translation of the rule were not carried out, the raw rule would have to be applied, and we would get the distinctly less satisfying answer **srqq**. But nothing in the diagram suggests such a substitution of concepts. There must be more to the vertical mapping than what is shown in the diagram if it is to give us the desired answer. How can we augment our perception of **srqp** to give it the requisite richness?

The answer, curiously enough, is suggested by a more careful examination of the appeal of the diagonal mapping. What is it about **srqp** that suggests a diagonal mapping? After all, it is very unlikely that someone faced with target situation **ijkl** would think of replacing the **i** by another letter. It goes without question that the proper site for change is the **l**. What makes **srqp** different? The answer is that our minds are taking into account the fact that **abc** and **ijkl** are <u>forwards alphabetic sequences</u>, whereas **srqp** is a <u>backwards alphabetic sequence.</u> But our diagrams do not, so far, provide for indications concerning the "internal fabric" of a string of letters. Let us therefore try a simple representation of the internal fabric of strings **abc**, **ijkl**, and **srqp**:

a --> b --> c

i --> j --> k --> l

s <-- r <-- q <-- p

Each arrow represents a <u>successorship link</u> between adjacent letters, and we see that in the upper two strings, the arrows all flow to the right, whereas in the lower string, the arrows all flow to the left. If we were to take these arrows as our guidelines for suggesting bridges, they would unequivocally push for the <u>vertical</u> mapping of **abc** onto **ijkl**, and the <u>diagonal</u> mapping of **abc** onto **srqp**. So far, we seem to have only decreased the justification for the vertical mapping of **abc** onto **srqp**! Adding arrows has certainly enriched our representation of what is going on, but hasn't yet solved our puzzle about how to justify the mapping that gives rise to the rule translation that in turn gives rise to the answer **srqo**.

Let us now recall what the desired rule translation was. It was the substitution of "*predecessor*" for "*successor*". If we wish to get the concept of <u>predecessorship</u> into the picture, it would seem that we would have to have a representation for that concept in our diagrams. And indeed, our diagrams have indubitably manifested a bias towards successorship and against predecessorship. So let us try again with our three strings, now giving equal time to predecessorship.

a <== b <== c

i <== j <== k <== l

s ==> r ==> q ==> p

In these diagrams, an arrow with double thickness represents a predecessorship link. We now realize that each of our three strings has a "bivalence": it can be seen as being composed of either successorship links or predecessorship links. And for each string, switching the type of link defining its internal fabric switches the direction of flow of the arrows.

We now have adequate notation to reexamine the mappings of **abc** onto **srqp**, so let us present enriched diagrams for those situations.

a --> b --> c (internal fabric: right successorship)

s <-- r <-- q <-- p (internal fabric: left successorship)

---

a --> b --> c (internal fabric: right successorship)

s ==> r ==> q ==> p (internal fabric: right predecessorship)

Here we have presented **abc** the same way both times, but accompanied by two different visions of **srqp**.

The upper vision frames both **abc** and **srqp** in terms of successorship links, and therefore the flows of arrows in the two strings are opposed: *right* versus *left*. The two starting-points of the flow of arrows (**a** and **p**) are each other's counterparts, as are the two finishing-points (**c** and **s**). This clearly suggests diagonal bridges:

a --> b --> c

Mapping 1: (diagonal mapping)

s<-r<-q<-p

The pylons of these two bridges, taking into account the identical link types but antiparallel link directions, say that when translation is called for here, the concepts *successor* and *predecessor* should be kept constant, while *left* and *right* should be swapped.

The lower vision, on the other hand, frames both strings in terms of right-moving arrows, and therefore the link types are different: *successorship* versus *predecessorship*. The two starting-points of the flow of arrows (now **a** and **s**) are each other's counterparts, as are the two finishing-points (**c** and **p**). This clearly suggests vertical bridges:

Mapping 2: (vertical mapping)

a --> **b** --> **c**

s => **r** => **q** => **p**

The pylons of these two bridges, taking into account the parallel link directions but opposite link types, say that when translation is called for here, the concepts *left* and *right* should be kept constant, while *predecessor* and *successor* should be swapped. So now we understand not only how the vertical mapping is established, but also how, once it has been set up, the pylons supporting its bridges mediate the proper translation of the rule, so as to produce the answer **srqo**. The insight that allowed us to get to the core of these two answers to the given problem was the notion that arrows, or links, have an intrinsic "bivalence".

This brings us to the central question of this article, namely: "How is it decided, for a given link inside a string of letters, which of its two facets will prevail?" This, after all, is what determines which of the two alternate visions of **srqp** will be chosen, thus determining the mapping, and thereby, the answer to the analogy problem. What makes the **srqp** problem especially interesting is that we wish to have both visions be possible, in principle, on different runs of the program. This would imply that the program need not produce identical answers on distinct runs, and therefore that there is some non-determinism in the program. That is certainly the case, and we now proceed to describe it.

The idea in a nutshell is that each inter-letter link is a bi-stable creature, choosing which way to point according to probabilistic laws rather than deterministically. When any link is first inserted between two letters, it makes such a choice, but that choice is not necessarily final. As the program runs, the link is from time to time given the chance to "change its mind". This makes links sound like capricious creatures, and perhaps gives the impression that utter chaos reigns, with links flipping direction all the time, and no order ever emerging. Such would indeed be the case if all the decisions were unbiased -- in other words, if each link simply flipped a fair coin each time to decide which way to point. But there are time-dependent pressures, both local and global, that bias the coins and tend to ensure that in the long run, order emerges. By local pressure, we mean that each link is somewhat biased to agree with what its closest neighbors are doing. This is roughly analogous to peer pressure. By global pressure, we mean that each link is somewhat biased to agree with the statistically predominant trends in the entire "world". This is roughly analogous to a national mood.

The final factor involved in the emergence of order from this probabilistic chaos is a notion called temperature[4,5,6,7], which serves to regulate the fairness of all the coins at once. A high temperature ("boiling") means that all coins are very nearly unbiased, so that every link can quite capriciously "change its mind", and thus the situation is highly volatile. A low temperature ("freezing") means that every link slavishly follows the latest trend (local or global), and thus the tiniest initial bias is rapidly magnified into an avalanche of conformism from which there is no escape.

The problem with high temperature is easy to see: it is that the system never settles down. The problem with low temperature is subtler; it is that the system is on a hair-trigger and will jump to a conclusion based on only one or two initial coin-flips. An intermediate temperature has some of the good and some of the bad qualities of both these extremes, and is thus not a good compromise. The best solution is to let the system regulate its own temperature, starting it out very high and then lowering it slightly whenever hints of order starts to emerge, and raising it slightly whenever order has not increased recently. The effect of such self-regulation is to "coax" the system gradually into a highly ordered state and to "freeze" it in that state. Of course, to do this, the system has to have a way of measuring its own order. We will describe that shortly.

In order to see how these factors work together to let an orderly vision emerge from an initially inchoate situation, let us follow an example. For the purposes of this discussion, the fact that there are two situations rather than one makes absolutely no difference. In fact, we can combine our two situations **abc** and **srqp** into one longer string, **abcsrqp**. This will serve just as well as the two separate strings to illustrate the ideas of link insertion and link flipping.

Let us look at what happens when the very first link is inserted inside our long string. The choice of site is one of the many non-deterministic aspects of our program; it could take place anywhere inside the string, depending on the output of a random-number generator. Suppose that the locus between the **r** and the **q** is chosen. Then one of the following two diagrams must result:

(1) **a b c s r** <- **q p**

(2) **a b c s r** => **q p**

In (1), we have a left-successor link, and in (2), a right-predecessor link. (The scenarios **r** -> **q** and **r** <= **q** represent false statements, and are impossible according to the rules of the game.) Now at the outset, our world is totally unbiased, favoring neither left-pointing nor right-pointing arrows, and favoring neither successorship nor predecessorship. This very first coin flip, therefore, is a 50-50 affair. Let us arbitrarily suppose, then, that fate (in the form of a random-number generator) favors the link in diagram (2). How does that solitary right-predecessor link, once inserted, begin to establish "peer pressure" and a "national mood"?

Locally, this link will tend to bias coin flips affecting its neighbor links (those at the **s-r** and **q-p** loci, although such links don't yet exist). In particular, their coins will be biased to favor rightwards motion and predecessorship. Globally, this link will also tend to bias all coin flips, no matter where they are, towards rightwards motion and predecessorship. Thus this link will tend to say to all links "Be like me!", but it will say it more forcefully to its two neighbors.

Suppose now that the next randomly-selected locus for link insertion is that between **a** and **b**. This link will be under no local pressures (it is too far from the only existent link), but under two distinct global pressures: (1) to be a predecessor-type link, and (2) to point rightwards. The first could be called link-type pressure, and the second could be called directional pressure. Note that in this particular case, these pressures happen to favor opposite outcomes, because the only links consistent with reality are **a** -> **b** (appeasing directional pressure but violating link-type pressure), and **a** <= **b** (appeasing link-type pressure but violating directional pressure). So in this case, the pressures cancel each other out, and consequently we have another 50-50 coin flip. (Strictly speaking, the pressures need not cancel each other precisely, because one of them may be given more weight than the other. But for simplicity's sake, let us right now assume that both pressures are considered equally important.)

Suppose our coin winds up selecting the left-predecessor link (**a <= b**). Now we have two predecessor links, and no successor links. That, in anyone's book, should be read as distinct pressure toward predecessorship. On the other hand, we have one right-moving link and one left-moving link. This is a clear case of cancellation, meaning that there is no pressure towards left or right as of yet. If this early-formed bias favoring predecessorship survives, we will be very likely to settle, as the temperature falls toward freezing, into the following final state:

```
a <= b <= c    s => r => q => p
 <======        ==========>
```

This state can be decomposed into two obvious separate parts (**abc** and **srqp**), each with its own uniform internal fabric, shown underneath. Such a region is called a Bloch domain, by analogy with the phenomenon of that name in ferromagnetism, described below.

Obviously, if the second flip had come out the opposite way, namely selecting **a -> b**, there would have been pressure favoring right-moving arrows, while pressures toward successorship and precedecessorship would have canceled each other out, and so the system would have been more likely to settle into the following final state, at low temperature:

```
a -> b -> c    s => r => q => p
 ------->       ==========>
```

Here again, the same two Bloch domains have emerged, but the one on the left has a different uniform internal fabric.

Just to show that **abc** and **srqp** are not the only possible Bloch domains, here is one other possible final state (although it is certainly less likely to crop up):

```
a <= b -> c    s => r => q <- p
 <==  -->       ======>   <--
```

In this state there are three tiny Bloch domains and one medium-sized one.

One might well ask why this particular state would be unlikely as a final state. The answer has to do with how order is measured and how temperature is regulated. We would ideally like final states to have long Bloch domains, because such groupings are similar to what humans tend to perceive. In the above situation, for instance, people would like to see the link between the **q** and the **p** flip, as well as either of the two links inside the **abc** region. Our strategy is thus for the system to lower its temperature slightly whenever a domain having a uniform internal fabric grows longer, and to raise the temperature whenever such a domain becomes shorter. Statistically speaking, the effect of this strategy will be a tendency for long Bloch domains to lock in stably -- but it does not totally prevent short ones from cropping up. And this is important, because there is no guarantee that perceiving the longest possible Bloch domains always provides the best answers to analogy problems.

It is critical that one begin at a high temperature, thus allowing the system to insert its first links relatively unbiasedly and to explore many possible "perceptions" by letting links flip freely, but it is of course important that one finish at a low temperature, thus ensuring that the system has committed itself stably to one unvarying set of links (i.e., to one stable "perception" of the situation).

This is perhaps the appropriate time to bring in the analogy of the Ising model of ferromagnetism[8]. In a ferromagnetic substance, the atoms are arranged in a regular lattice, and each atom is the locus of a spinning electron. For the purposes of this discussion, a spinning electron can be thought of as a tiny magnet capable of pointing in only two directions, usually called up and down. In a real substance, the interactions among spins can be very complex. The Ising model is an idealized model of a ferromagnetic substance, but a very accurate one. Each spin is assumed to be susceptible to "peer pressures" only from its nearest neighbors. (In one dimension, each spin has two nearest neighbors; in two dimensions, four; and in three, eight.) More precisely, this means that each spin "wants" to align itself with the local magnetic field created by its nearest neighbors. Temperature, however, spoils any certainty that it will so conform.

Moreover, if there is a global external magnetic field acting on the substance, every single spin is subject to that "national mood" as well. This external magnetic field H_{global} is added to the sum H_{local} of the magnetic fields of the nearest neighbors, to create a total magnetic field H_{total}, whose value varies from spot to spot and from moment to moment.

At high temperatures, the Ising model has spins flipping wildly, and forming no global order. At low temperatures, the Ising model settles into disjoint macroscopic regions of atoms whose spins are all aligned. These regions are known (not surprisingly) as Bloch domains, and they are the source of the exceedingly strong intrinsic magnetic fields that substances such as iron characteristically exhibit.

Like our links, Ising-model spins flip probabilistically, biased by the sum of the local magnetic field set up by their neighbors and the global magnetic field (if any exists). When the temperature is high, however, any bias is essentially ignored, so that flips tend to be truly 50-50. As the temperature is lowered, biases receive more and more attention -- and at very low temperatures, a bias becomes effectively an iron-clad rule, so that all randomness is removed.

The process can be expressed quite simply in mathematical terms. To each of a given spin's two possible orientations, there corresponds an energy. Barring coincidences, one of these energies will be lower than the other, and the larger this gap between energies is, the more the spin will want to assume the lower-energy state. The only thing holding it back from doing so is, of course, the temperature. In particular, the probability that a spin will be found in a state of energy E when the temperature is T is proportional to the following expression:

$$e^{-E/kT}$$

where k is the Boltzmann constant. For our purposes, k is irrelevant, so let us simply set it to 1. Let us now suppose that the energies corresponding to the up-state and the down-state are, respectively, E_u and E_d. Then the respective probabilities, p_u and p_d, of the spin being found in those states are (to within a common factor of proportionality):

$$p_u = e^{-E_u/T} \qquad \text{and} \qquad p_d = e^{-E_d/T}$$

Since the spin must point either up or down, the probability that it will point up is given by the following ratio:

$$\frac{p_u}{p_u + p_d} = \frac{e^{-E_u/T}}{e^{-E_u/T} + e^{-E_d/T}} = \frac{1}{1 + e^{(E_u - E_d)/T}}$$

(In this ratio, the common factor of proportionality cancels out, fortunately.) We can interpret this formula in two ways: if we have a real ferromagnetic substance, then it represents the probability that a given spin will be found pointing up rather than down; if, however, we are computationally simulating such a substance, then it tells us how to bias our coin flip determining the direction of a given spin.

What has not yet been explained is how to calculate the energies attached to spin-up and spin-down states. This is, fortunately, very simple. Suppose a particular spin has value s, and is immersed in a magnetic field of value H. Actually, s can assume only two possible values: +1 (spin up) and -1 (spin down), whereas H can assume any real value, positive or negative. If s and H have the same sign, then s is aligned with H, otherwise s opposes H. Electromagnetic theory tells us that the energy associated with our spin is:

$$E = -sH$$

In particular, for up-spins and down-spins respectively, the two energies are:

$$E_u = -H \qquad \text{and} \qquad E_d = H$$

What this means in words is that a spin parallel to H has a lower energy than a spin antiparallel to H -- and the larger H is, the bigger that energy discrepancy is. When these two expressions are substituted into the formula for the probability of a coin-flip choosing "up", we get the following expression:

$$\frac{1}{1 + e^{-2H/T}}$$

This has a very simple interpretation. When T is very large, the exponential is close to 1, so that the whole expression is close to 1/2, meaning that the coin flip is essentially unbiased -- just about equally likely to pick "spin down" and "spin up". When H/T is <u>positive</u> and enormous (so that the bias towards "spin-up" should be great), the exponential is very nearly zero, so that the whole expression is very nearly 1, meaning that the coin flip is almost certain to choose "spin up". Finally, when H/T is <u>negative</u> and enormous (so that the bias towards "spin-down" should be great), the exponential is very nearly infinity, so that the whole expression is very nearly zero, meaning that the coin flip will almost never choose "spin up". This is just what we said earlier: high temperature means that coin flips are unbiased so that no global order emerges, whereas low temperature tends to enforce biases very strongly, meaning that spins will line up with their surrounding magnetic field and will form large uniform Bloch domains.

The Ising-model approach lends itself readily to our work, by analogy. The temperature T has already been introduced and, since it is an arbitrary parameter that can be raised or lowered at will, needs no further explanation. Flippable spins are, of course, bivalent links. What, though, corresponds to the magnetic field H? Well, we have seen that H tends to coerce spins to align with it; therefore, H ought to be equated with <u>pressure to conform</u>.

We saw above that in the Copycat world, there are actually two distinct "flavors" of pressure to conform: link-type pressure and directional pressure. To emphasize the analogy to ferromagnetism, let us rename them <u>link-type field</u> H_t and <u>directional field</u> H_d. H_t is a real number telling us how much we should favor successorship links over predecessorship links, and similarly, H_d is a real number telling us how much we should favor right-pointing arrows over left-pointing arrows. (A negative value of H_t means that predecessorship should be favored over successorship, and similarly, a negative value of H_d means we should favor left-pointing arrows.) The following formulas for these quantities suggest themselves:

$$H_t = \#(\text{succ-links}) - \#(\text{pred-links})$$

$$H_d = \#(\text{right-links}) - \#(\text{left-links})$$

Here, a notation such as "#(x)-#(y)" simply means "Count up the number of current instances of x, and subtract from it the number of current instances of y". These definitions capture the idea that fields are up-to-the-minute fad-measurers or "polls" tracking the popularities of link types and link directions.

Every time a link is about to be inserted or flipped, the bias of the associated coin flip is determined by a calculation using the values of these two fields at that locus. But there is an interesting feedback effect as well: right after each coin flip, the new spin state affects H_t and H_d in turn, because that spin state itself has to be counted if the two polls are to remain up to date.

The fact that there are two types of "magnetic field" reminds us that a link, as well, is actually two spins in one: a "link-type" spin where +1 and -1 mean "successor" and "predecessor" respectively, and a "directional" spin, where +1 and -1 mean "right" and "left" respectively. Thus instead of just one value s associated with a link, we have a pair of such values per link: s_t and s_d. To calculate the energy associated with a given link's state, we multiply each spin by its corresponding field and sum the results:

$$E_t = -s_t H_t \qquad\qquad E_d = -s_d H_d$$

$$E_{total} = E_t + E_d$$

We have overlooked one detail -- there are actually two components to each type of pressure to conform: the global component, which doesn't vary from spot to spot, and the local component, which does, since it depends on a locus's immediate neighbors. Therefore we should write:

$$H_{tl} = \#_{local}(\text{succ-links}) - \#_{local}(\text{pred-links})$$

$$H_{tg} = \#_{global}(\text{succ-links}) - \#_{global}(\text{pred-links})$$

$$H_{dl} = \#_{local}(\text{right-links}) - \#_{local}(\text{left-links})$$

$$H_{dg} = \#_{global}(\text{right-links}) - \#_{global}(\text{left-links})$$

The subscript "local" means that one should poll only the immediate neighbors on either side of the given locus, while "global" means that one should poll all links, no matter where they are located.

This means that to make E_{total}, we really should be summing four quantities rather than two:

$$E_{tl} = -s_t H_{tl} \qquad E_{dl} = -s_d H_{dl}$$

$$E_{tg} = -s_t H_{tg} \qquad E_{dg} = -s_d H_{dg}$$

$$E_{total} = c_{tl}E_{tl} + c_{tg}E_{tg} + c_{dl}E_{dl} + c_{dg}E_{dg}$$

The only thing non-obvious thing here is the presence of coefficients, $\{c_{ij}\}$. They are included because it is quite conceivable that one might wish to weight global fields more or less heavily than local fields, and, as was briefly mentioned earlier, link-type fields more or less heavily than directional fields.

For the sake of concreteness, let us take the following situation:

a -> b -> c   s => r <- q => p

and concentrate on the link in the locus between **r** and **q**. Cursory examination of the situation shows that its two neighbors would very much like it to flip into a right-predecessor link, thus: **r => q**. Globally, the situation is a little more ambiguous, since there is considerable pressure toward right-pointing links, but at the same time, successor links are slightly favored over predecessor links. If all the coefficients $\{c_{ij}\}$ are equal, we would expect that this link would be more likely to flip than to remain as it is.

To check these intuitive conclusions, let us first calculate the values of all the field components at the given locus:

$$H_{tl} = \#_{local}(\text{succ-links}) - \#_{local}(\text{pred-links}) = 0 - 2 = -2$$

$$H_{tg} = \#_{global}(\text{succ-links}) - \#_{global}(\text{pred-links}) = 3 - 2 = +1$$

$$H_{dl} = \#_{local}(\text{right-links}) - \#_{local}(\text{left-links}) = 2 - 0 = +2$$

$$H_{dg} = \#_{global}(\text{right-links}) - \#_{global}(\text{left-links}) = 4 - 1 = +3$$

As things stand right now, $s_t = +1$ (it is a successor link), and $s_d = -1$ (it points to the left). Were it to flip, then we would have $s_t = -1$ and $s_d = +1$. So we need to calculate the energies of these two rival states. We will assume all the $\{c_{ij}\}$ are equal to 1, for simplicity. For the actual state, we have:

$$E_{tl} = -s_t H_{tl} = -(+1)(-2) = 2 \qquad E_{dl} = -s_d H_{dl} = -(-1)(+2) = 2$$

$$E_{tg} = -s_t H_{tg} = -(+1)(+1) = -1 \qquad E_{dg} = -s_d H_{dg} = -(-1)(+3) = 3$$

The sum of these four energies is +6. Now what about the energy if the link were flipped around? Well, this is rather trivial; since both s_t and s_d would change in sign, all four contributions would likewise change in sign. (Note: the four fields stay unchanged until the link actually flips; only at that point are they updated.) Consequently the energy of the hypothetical flipped state would be -6, which is lower by 12 units. Already we can see that our link will be biased towards flipping. The only question is by how much -- and that, of course, depends on the ratio of the energy gap to the temperature. Specifically, the probability of the link's remaining in its current orientation is:

$$\frac{1}{1 + e^{12/T}}$$

If the temperature is, say, 12, then this quantity is $1/(1+e)$, or about 0.27, so that the chance of reversal is almost 75 percent. At double that temperature, this quantity is about 0.38, so the chance of reversal is about 62 percent, which makes sense, since high temperatures tend to reduce biases. At a temperature of 6, this quantity is about 0.12, so that there is almost a 90 percent chance of the link's flipping around! As was mentioned earlier, links tend to "go with the flow" when the temperature is low.

Let us now summarize. Given any situation, a program can easily determine the magnitudes of all four types of pressure ("magnetic fields") at any link-locus. From those values, it calculates the energy gap between the link's two possible orientations, and then, using the current temperature, it determines the bias on the coin it is about to flip. Using a random-number generator, it flips a coin to decide which way to point the current link (and incidentally, whether that link already exists or is about to be created makes no difference to the calculation). Having done this for a given link, it can now go on to another link, and then another, and so on. As it carries out these link-insertion and link-flipping processes, one or more Bloch domains may begin to emerge, and as they grow, the temperature will fall. As the temperature falls, the probability of flipping a link in violation of either the "peer pressure" caused by the Bloch domain it belongs to or the "national mood" caused by the totality of Bloch domains also falls. There is thus a strong tendency toward "locking in" to stable, large Bloch domains. As a result, the Copycat program tends to zero in on highly plausible perceptions of these abstract situations.

It is at this point of Copycat's work that the mapping, or bridge-building, stage takes over from the perceptual, or link-making, stage; once the mapping is done, Copycat goes on to the rule-translation stage, which leads immediately to the construction of its answer to the given analogy problem. And thus ends Copycat's work.

References

Hofstadter, Douglas R. "The Copycat Project: An Experiment in Nondeterminism and Creative Analogies." Cambridge, Massachusetts: M.I.T. Artificial Intelligence Laboratory AI Memo #755, April 1984.

Hofstadter, Douglas R. "Analogies and Roles in Human and Machine Thinking". Chapter 24 of Metamagical Themas. New York, Basic Books (1985): 547-603.

Hofstadter, Douglas R. "Simple and Not-So-Simple Analogies in the Copycat Domain." Unpublished FARG Document (January, 1984), available through Fluid Analogies Research Group, Perry Building, 330 Packard Road, Ann Arbor, Michigan 48104, U.S.A.

Hofstadter, Douglas R. "The Architecture of Jumbo". In Proceedings of the Second Machine Learning Workshop. Monticello, Illinois (1983). Also available through Fluid Analogies Research Group, Perry Building, 330 Packard Road, Ann Arbor, Michigan 48104, U.S.A.

Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi. "Optimization by Simulated Annealing". Science **220**, no. 4598 (May 13, 1983): 671-80.

Smolensky, Paul. "Probabilistic Analysis of Inference and Learning in Massively Distributed Parallel Cognitive Systems: The Framework of Harmony Theory". University of California at San Diego, Institute for Cognitive Science Technical Report (1984).

Fahlman, S. E., G. E. Hinton, and T. J. Sejnowski. "Massively Parallel Architectures for Artificial Intelligence: NETL, Thistle, and Boltzmann Machines". In Proceedings of the National Conference on Artificial Intelligence (1983), available through the American Association for Artificial Intelligence, 545 Burgess Drive, Menlo Park, California 94025.

Ziman, J. M. Principles of the Theory of Solids (2nd ed.). Cambridge, U.K., Cambridge University Press (1972): 353-66; 372.