Motivation for a New Semantics for Vagueness

Mark A. Changizi
Department of Computer Science, University College Cork
Cork, Ireland

Abstract

Vagueness is the phenomenon that natural language predicates have borderline regions of applicability and that the boundaries of the borderline region are not determinable. A theory is presented which argues that vagueness is due to the fact that we are computationally bound by Church's Thesis. Syntactic and semantic models motivated by the theory are introduced. Each disallows the use of classical negation, capturing the fact that it is generally only possible to semidecide but not decide our interpretations of natural language predicates. The role of negation is filled, for each predicate R, by the existence of a dual predicate $nonR$ that acts as if it is the negation, although its interpretation is generally, at best, only an approximation to the complement of R. Multiple "levels" of vagueness are modeled using concepts from recursion theory.

Introduction

In this paper I present a new explanation of vagueness. Section 1 specifies some of the aspects of vagueness in need of explanation; Section 2 motivates the theory, concluding with a characterization of vagueness; Section 3 discusses how vagueness follows from the characterization; and Section 4 presents simple idealized logics of vagueness motivated by my explanatory theory.

1 The phenomenon of vagueness

What is the phenomenon of vagueness? What is it that is in need of explanation? Some names of phenomena comprising the phenomenon of vagueness are 'borderline region', 'higher-order vagueness', 'sorites paradox' and 'ineliminability'; the first two are central. All things equal, a theory that satisfies more of the phenomena is more favorable.

What are these phenomena? It is dangerous to attempt to precisely and formally define them since we have no clear pre-theoretic agreement on what exactly are the data, and any such definition is likely to be theory-laden to some extent. Accordingly I want to remain open-minded. I will give the rough idea for each phenomenon with the understanding that I am in no way defining what it is exactly. The best I hope for is an independently motivated theory that results in certain plausible phenomena that seem to match closely with the rough definitions of those named above. On to the phenomena.

The *borderline region* phenomenon is roughly the phenomenon that for a vague predicate R we find ourselves with objects for which R neither clearly applies nor clearly does not apply; these objects are in the borderline region. Or, an object is borderline if it does not fit neatly into just one category. Alternatively, an object is borderline R if when we are given the choice "Which is it, R or not, and not both?" we do not know quite how to respond, and our non-response is seemingly not because we simply do not know, but because it seems fantastic to suppose there is exactly one correct response; this is partially what distinguishes vagueness from other sorts of unknowabilities. I say "seemingly" above because otherwise I exclude the possibility of an epistemic two-valued theory being correct. The borderline region is also connected with the phenomenon that we are incapable of drawing a single sharp line distinguishing between things R and things not R, and more than this, it is that any line drawn would seem ad hoc, arbitrary and wrong. Sometimes the borderline region is defined more epistemically as that region for which knowledge concerning membership in R is unattainable. The phenomenon is probably best communicated by example: wolves are borderline dog, violet is borderline blue, and so on.

Higher-order vagueness, second-order vagueness in particular, is the phenomenon that we find ourselves incapable of determining boundaries of the borderline region. Alternatively, imagine pulling hairs out of the head of a man who is definitely not bald. Second-order vagueness is exemplified by the fact that we do not find ourselves pulling out a single hair for which we are able to determine that the man suddenly becomes borderline bald. We find objects, or states of this man's head, which are borderline borderline bald. More explicitly epistemically, knowledge of the boundaries—if there are any—of the borderline region is unattainable. Higher-order vagueness, more generally, is the phenomenon that we find ourselves incapable of determining any semantically distinct boundaries at all between definitely bald and definitely not bald.

There is a third phenomenon linked to vagueness: the *sorites paradox*. Its standard form is exemplified by the following two-premise argument and conclusion: (i) 1 grain of sand cannot make a heap, (ii) for all n, if n grains of sand cannot make a heap, then $n+1$ grains of sand cannot make a heap, (iii) there are no heaps of sand. (i) is obviously true.[1] (ii) is very compelling, since to deny it means that there is some n such that n grains of sand cannot make a heap but $n+1$ grains can make a heap—that there is a to-the-grain distinction between being a heap and not—and this seems fantastic. (i) and (ii), though, imply (iii), which is obviously false. The sorites paradox is part of the phenomenon of vagueness in that it may be built using any vague predicate. There are two related constraints on a theory of vagueness. The first is that it locate the fault in the sorites argument, and do so without having to make fantastic claims. I do not mean to imply that the classical negation of the induction step cannot possibly be the solution to the sorites paradox. Epistemic, two-valued theories do exactly this, but it is incumbent upon the theory to say why it is not so fantastic; for example, that we cannot ever determine or know the boundary, and this is why the suggestion that there is a boundary seems incredible. The second constraint is that a theory's post-theoretic notion of the phenomenon of vagueness should be such that a sorites argument built around a predicate displaying the phenomenon is paradoxical; i.e., denying the induction step must seem paradoxical. If the argument loses its paradoxical aspect, then the phenomenon claimed to be vagueness has a lesser claim to vagueness. For example, if we cannot ever determine or know the boundary but still believe quite reasonably that there is one, then there is nothing paradoxical in the sorites argument since the induction step can be (classically) denied readily without intuitive difficulties.

The final phenomenon, *ineliminability*, is less central to vagueness: it is often felt that vagueness is not something we can simply eliminate from natural language. For example, it is sometimes said that any attempt to eliminate vagueness through precisification would, at best, radically undermine the meanings of natural language concepts. Also, restricting oneself to some delimited context is also thought to be unhelpful in eliminating vagueness—vagueness occurs within contexts. My theory of vagueness explains (Subsection 3.4) and accommodates (Subsection 4.2) a variety of ways in which vagueness is ineliminable. I am less confident that this phenomenon is a necessary constraint on a theory of vagueness, although I am inclined to think that some degree of ineliminability must be addressed, lest we be left to wonder why we have not cured ourselves of vagueness.

The phenomena just presented need to be explained, not just modeled. That is, a logic devised just to accommodate these phenomena is not sufficient to have the status of an explanatory theory of vagueness. I want to know why there is vagueness, not just how to describe it.

2 Theory

Suppose that Church's Thesis is true, you are finite and thus bound by it, and you are capable of computing any computable function (I refer to this latter conjunct as "sufficiently powerful"). You have, then, at your disposal programs "in the head" that may be used for your computations; and that is all you have at your disposal. Programs in the head must be employed in order to determine whether some predicate of natural language applies to some given object. For example, your program for 'bald' takes as input the natural number coding some particular human scalp (say, Yul Brynner's) and outputs YES. The set of inputs on which your program for 'bald' outputs YES is your *interpretation* of the predicate, and it is thus true that your program *semidecides*[2] your interpretation. This is the model of human cognition I presume.

[1] Although some have sought to save us from paradox by denying the base case. Unger [8] and Wheeler [9] deny that there are non-heaps by denying that there is a concept heapness at all.

[2] A program semidecides a set if it halts and says YES on all and only those inputs in the set; but when input with something not in the set, the program might not halt at all.

These programs are acquired as a result of a learning process, but however they are obtained there is one thing of which we can be certain: the programs are not in general algorithms. One of the basic results in the theory of computation is that there is no algorithm for algorithmhood. In choosing programs from the set of all possible programs it is impossible for you to generally choose algorithms only, and therefore it is true that although your program for 'bald' semidecides the associated interpretation, it does not generally decide it. Your interpretations are "usually"[3] semidecidable but not decidable using the programs that determine them; you can tell when something is in the interpretation of 'bald', but not generally when something is not (and this is true even if the interpretation happens to be recursive, or even finite).

But you want to be able to identify things that are not bald (i.e., identify things in your interpretation of the natural language expression 'not bald'); you cannot just refuse to identify half the scalps brought before you! Not only can you not easily refuse to identify scalps that are not bald, you are, as a matter of fact, generally able to identify things that are in your interpretation of 'not bald'. Since the program for 'bald' is not helpful in this regard it must be a different program that is responsible for this (this other program also is not in general an algorithm). Whatever this other program may do, though, it is not generally capable of semideciding the complement of your interpretation of 'bald' (i.e., the interpretation of '¬bald') because if it were possible to always acquire such a program, then the pair of them could be used as a single algorithm, and this contradicts the impossibility of generally acquiring algorithms; your interpretation of 'not bald' cannot generally be that of '¬bald' (although the idea is that it be a near enough approximation). 'not bald', having its own program in the head and interpretation, should be represented in logic not as '¬bald' but as a single predicate itself: 'non-bald', the *dual* to 'bald'. Therefore you generally have two programs used when you discuss baldness, neither program is an algorithm, and your interpretation of 'bald' and 'not bald' are not complements. 'Yul is bald' and 'Yul is not bald' are translated into logic as, respectively, 'bald(Yul)' and 'non-bald(Yul)' (*not* '¬bald(Yul)'). 'non-bald' is a distinct predicate, pragmatically related but logically unrelated to 'bald'. We will see that this leads to vagueness.

That ends the informal presentation of my theory, which I call the Undecidability Theory of Vagueness.[4] In a nutshell it is: (a) you are finite and sufficiently powerful, (b) so you cannot generally acquire algorithms for your concepts, say the one denoted by the predicate R, (c) thus you can generally only semidecide your interpretation of R, (d) and to cope with the "black hole" complement of your interpretation of R you must resort to another program which is (e) also not generally an algorithm and (f) is at best able to provide an interpretation for 'not R' that is a rough approximation to the complement of your interpretation of R; (g) thus 'not R' is a predicate distinct from R, and I have labeled it '$nonR$' and called it the dual to R. The three key elements of this development are (i) your interpretation of R is determined by a program in your head that is capable of semideciding but not deciding it, (ii) your interpretation of 'not R' is determined by a program in your head that is capable of semideciding but not deciding it, and (iii) your interpretation of R is not the complement of your interpretation of 'not R'. The development above has lead to the conclusion that "most" natural language predicates are subject to (i), (ii) and (iii). My characterization of vagueness is that *a predicate is vague if and only if it satisfies (i), (ii) and (iii)*. Allow me to emphasize that my theory does *not* identify vagueness with undecidability.

3 Vagueness

In this section I demonstrate how the Undecidability Theory of Vagueness explains vagueness.

[3] More formally, this "usually" is captured by the fact that the set of algorithms is Π_2, and the set of non-algorithms Σ_2; the relative difficulty of acquiring algorithms versus non-algorithms is analogous to the relative difficulty of determining cases where a program does not halt versus when it does. I use scare quotes around 'usually' or 'most' when meaning this.

[4] The development from this section is the subject of its own full paper (Changizi [2]). That paper argues that the characterization holds for "most" natural language predicates, and this claim is shown to follow from three hypotheses, each which is defended in detail: The first hypothesis is that natural language users are bound by Church's Thesis and are sufficiently powerful, as discussed here. The second is that natural language user's interpretations of natural language predicates are determined using programs in the head (rather than non-computable definitions, say). And the third is that natural language users do not constrain the set of possible programs to a recursive subset of the set of algorithms. That paper shows the extent to which each hypothesis may be weakened without losing the claim.

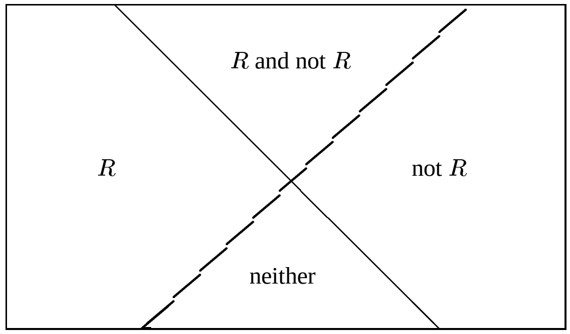

Figure 1

Figure 1: The definitely R and definitely not R regions are on the left and right sides, the borderline regions in between, with overlap region on top and gap region on the bottom.

3.1 Borderline region

Your interpretation of 'not R' is not equal to that of '$\neg R$' and there are three possibilities. The first is that your interpretation of R and that of 'not R' overlap and every object is in at least one of the two interpretations (overlap region only); there are objects c such that the natural language sentences 'c is R' and 'c is not R' are both true. The second is that there is no overlap and there are objects which are in neither interpretation (gap region only); there are objects c such that the natural language sentences 'c is R' and 'c is not R' are both false. The third is that there is an overlap region and a gap region. The borderline region is the union of the overlap and gap region. The definitely R region is the part of your interpretation of R not part of the borderline region, and the definitely not R region is the part of your interpretation of 'not R' not part of the borderline region. See Figure 1.

For any object c in the borderline region, either it is in the overlap, in which case the natural language sentence 'c is R and not R' is true, or it is in the gap region, in which case 'c is neither R nor not R' is true. This is plausible; each of these cases fits well with the datum of a borderline region: that there are objects which do not seem to fit neatly into just one category. For any object c in the definitely R region the natural language sentence 'c is R' is true but 'c is not R' is false. Finally, for any object c in the definitely not R region the natural language sentence 'c is not R' is true but 'c is R' is false.

The borderline region phenomenon my theory displays does more than simply have objects "which do not seem to fit neatly into just one category," the objects are sometimes in both categories, and sometimes in neither. The former is rarely considered a viable option since it is not obvious how to devise a coherent semantics respecting it, although Thorpe [7] is an exception. Not knowing how to build such a semantics and there not being an adequate such semantics are, of course, two different things. I am able to get away with building an adequate semantics including overlap via denying that the natural language negation of a predicate is logically related to the predicate. While this would certainly be an unmotivated supposition without my theoretical development from Section 2, with my theory the supposition is natural, compelling and unavoidable. Furthermore, there is a benefit in allowing that borderline R objects are both R and not: natural language users do make exactly such utterances. We will see later that the overlap borderline region is in principle eliminable, but I do not claim that individuals actually do eliminate it.

On the gap region it may be criticized that since neither predicate applies, perhaps the gap region consists of inapplicable objects. For example, 'red' applies to some apples, 'not red' applies to bananas, but neither applies to adverbs. Adverbs are not, however, genuine borderline cases of redness, and if the gap region consists only of inapplicables, then it is not part of the phenomenon of vagueness and does not help to explain the borderline region. The answer to this critique is that I have implicitly been assuming that the relevant domain under consideration with respect to any predicate contains only applicables, and even under such an assumption there will be a gap region. Expanding the domain to include inapplicables as well only serves to possibly enlarge the already existing gap region.

My theory is consistent with a two-valued "true" semantics (as opposed to your interpretations resulting from the programs in your head); for example, perhaps BALD is the true extension of 'bald' and its complement the true extension of 'not bald'. Under such a supposition of a two-valued true semantics, my theory may be interpreted as an epistemic theory of vagueness. In epistemic theories of vagueness the borderline region is characterized as those objects for which knowledge of membership is unattainable, where "membership" refers to membership in the true extension or true complement. How does the Undecidability Theory of Vagueness explain this unattainability of knowledge?

Although BALD is the true extension of 'bald', you are not generally capable of acquiring a program in the head that decides BALD, even if BALD is decidable, because you are not generally capable of acquiring algorithms. Your interpretation of 'bald' is semidecidable but not generally decidable by the program responsible for it, and even if you are so lucky to correctly interpret it (i.e., your interpretation is equal to the extension BALD), if you want to be able to respond to queries about 'not bald' you must acquire a second program in the head, and this program will not generally correctly interpret 'not bald'. Your interpretations of 'bald' and 'not bald' are, as before, not complements, and the programs for each only semidecide them. There are therefore objects for which you are incapable of determining or even knowing, using your programs in your head, whether or not it is a member of BALD.

3.2 Higher-order vagueness

Although you cannot draw a sharp line between 'bald' and 'not bald', can you draw a sharp line between 'definitely bald' and 'borderline bald'? There is, in fact, a sharp line here posited by my theory, but are you generally capable of drawing it? No. The two programs in the head for baldness (one for 'bald' and one for 'not bald') are not powerful enough to determine the lines. To see this intuitively, imagine starting in the 'overlap bald' region and moving toward the 'definitely not bald' region. While in the borderline region both programs eventually halt, just as you move over the line the program for 'bald' does not generally halt, and of course you are not generally able to know that it will never halt—you cannot generally know when you have crossed the line. Similar observations hold for the other sharp lines of the borderline region. This seems to be consistent with our observations of higher-order vagueness, and it solves the problem without having to postulate semantically distinct higher-order borderline regions. This latter aspect is good since it puts a stop to the regress of higher and higher order semantically distinct borderline regions, all which amount to nothing if when one is finished there is still a knowable sharp line between the definite region and the non-definite region.

Can this phenomenon be the phenomenon of higher-order vagueness? In my theory what does it "feel like" to not be capable of determining the boundaries of the borderline region? Well it feels like whatever it feels like to attempt to decide a set using a program that only semidecides it. One might try to make the following criticism: Let us take the set of even numbers and supply you with a program that only says YES exactly when a number is even, and is otherwise silent. Do the evens now seem vague through the lens of this program? There are a number of problems with such a criticism as stated. First, it is not enough that the program simply says YES when an input is even and is silent otherwise. When we say that the program semidecides but does not decide the set of evens we mean that if the program is silent we are not sure whether it will at any moment converge and say YES. The program's silence is not translatable to NO. The critic can now say that, of course, that is what he meant in the first place. Second, it is difficult to judge our intuitions with a predicate like 'even' for which we already have a program in the head for deciding it. The critic can now say that imagine instead that it is some new predicate R for which we have no intuitions. The third problem is that even with these fixes the question the critic needs to ask is not whether R-ness seems vague, but whether R-ness seems to have whatever feel higher-order vagueness has. This is because R is not vague according to my theory since it does not satisfy the characterization ((i), (ii) and (iii)) of vagueness. On this modified question it is unclear that we have any compelling intuitions that the answer is NO. When using the given program to attempt to decide the extension of R, you will be incapable of seeing where exactly is the boundary, and therefore many objects you will be unable to classify. These objects plausibly are just like the borderline borderline objects.

To be sure, there is no real semantically distinct second-order borderline region as there is for the first-order borderline region, but a sort of second-order borderline region still is accounted for. What about the boundaries of the second-order borderline region, third-order vagueness? Can we determine the (not semantically distinct) boundaries of the second-order borderline region? My official response is to deny that I am responsible for even worrying about explaining this. Our intuitions are pushed beyond the limit here; there is no clear data that higher-order vagueness goes this high. Are not not-generally-possible-to-determine boundaries of the borderline region good enough for higher-order vagueness?

A critic may ask the following: Let us suppose that you have two programs that only semidecide their respective interpretations, and let us also suppose that the interpretations are not complements. If these programs are for some predicate R and 'not R' then is R-ness necessarily vague? For example, let us take the predicate 'theorem of Peano Arithmetic', whose extension is recursively enumerable but not recursive, and presume that you have the program semideciding it. As a mathematician (pretend you are one if you are not) you are surely capable of determining some non-theorems, and you must therefore utilize some sort of program in the head for 'not theorem'. But surely 'theorem' is not now vague.

The difference between this case and vague natural language predicates is two-fold. (a) You as a mathematician are conscious that you are not actually semideciding the complement of the set of theorems of Peano Arithmetic with your second program. You understand that it is only a heuristic, and it is possible that it might even occasionally be incorrect, saying that a theorem is not a theorem. That is, the set determined by your program for 'not theorem' is not your interpretation of 'not theorem'. (b) You know that your program for 'theorem' is correct. Now suppose I ask you about a sentence σ of arithmetic that actually is a theorem, "Is σ a theorem or not?" You run both programs in your head and find, say, that both programs say YES. However, here you know that your first program, the one for 'theorem' is correct (via a proof you have seen), whereas you know that your program for 'not theorem' is only a heuristic (via a proof that there is no correct such program). (ii) from the characterization of vagueness is not satisfied here since your interpretation of 'not theorem' is not determined by any heuristic program you may employ. Your interpretation of it is, instead, the complement of the interpretation of 'theorem', however inaccessible that may be to you.

The critic may continue, however. What if you do not know that the first program is correct, and do not know that the second program is incorrect? What if these programs are just what you are given, and you do not know that, as it happens, the first happens to correctly semidecide a certain mathematically precise set? Is the predicate 'theorem' now vague to you (where we imagine that the word 'theorem' is just some word to you, with no connotation)? My response is: Yes. Suppose that the theorems of Peano Arithmetic were part of the everyday furniture of the universe, just like bald things. Suppose also that you have no understanding of metaproof, so you do not know anything about whether your programs actually do correctly determine the extension of 'theorem'. You would, I claim, acquire a program in the head for 'theorem', and supposing you were lucky enough to get the correct one, you would nevertheless need a second program to respond to queries, "σ is not a theorem, right?" If your second program happened to be the heuristic one mentioned above and you were presented with the same sentence as mentioned above that is actually a theorem, you would run both programs and find that it seems to classify as both a theorem and not. Furthermore, you would not seem to be able to determine any boundary between where such borderline cases end, and so on. That is, even 'theorem of Peano Arithmetic' would then seem vague. This is, however, no longer fantastic, since you have no idea of the mathematical nature of theoremhood. Such a scenario with 'theorem' cannot violate any of our pre-theoretic intuitions, since we do not have intuitions about what it would be like if theoremhood were part of the furniture of the universe just like tablehood and chairhood.

3.3 The sorites paradox

We arrive at the sorites paradox, which I give here in the following form: (i) 0 hairs is bald, (ii) for all n, if n hairs is bald, so is $n+1$, (iii) therefore you are bald no matter how many hairs you have. Notice that I have stated the argument in natural language; many researchers on vagueness state the paradox in some logical language, which is strange since the paradox is one in natural language. Presenting it in a logical language inevitably makes certain biased presumptions; for example that 'not' is to be translated to the classical negation '$\neg$'.

What is usually dangerous about rejecting premise (ii) is that it implies there is an n_0 such that n_0 hairs is bald but n_0+1 hairs is not; i.e., it usually leads to there being no borderline region. In my theory's case, though, what happens? A sorites series moves along a "path" that is most gradual from definitely R to definitely 'not R'; it must therefore cross the borderline region lest it not be "most gradual." Therefore, either the series crosses the gap region or the borderline region. Imagine starting in the definitely bald region and moving toward the gap borderline region. Eventually there will be a number n_0 such that n_0 hairs is definitely bald but it is not (this is a classical negation) the case that n_0+1 is bald, and you cannot in general determine where this occurs. However, this in no way prevents n_0+1 from being borderline bald, i.e., being neither bald nor not bald. Or if there is an overlap region, imagine starting in the 'definitely bald' region and moving toward the overlap borderline region. Eventually there will be a number n_0 such that n_0 hairs is definitely bald but $n_0 + 1$ hairs is not bald, and you cannot in general determine where this occurs. Again, this in no way prevents $n_0 + 1$ from being borderline bald, i.e., this time being both bald and not bald. Thus, eventually the denial of (ii) will occur—and you will not know when—but it does not imply the lack of a borderline region. The sorites paradox is thus prevented without losing vagueness. This was the first constraint concerning the sorites paradox (see Section 1): to locate the fault in the sorites argument, and to do so without being forced into making fantastic claims.

One technical difficulty with this explanation is that I have not specified how to take the natural language negation of the induction step. I have presumed in the discussion above that I can move from "it is not the case that for all n, if n hairs is bald, so is $n + 1$" to "there is an n such that n hairs is bald but $n + 1$ hairs is not bald." 'not' here is not necessarily '$\neg$', and I presume that natural language 'not' operates on formulas as does '$\neg$'; after all, '$\neg$' is originally modeled on 'not', not the other way around. I make this explicit in Subsection 4.1 with the negation function α.

The other constraint from Section 1 concerning the sorites paradox is that the post-theoretic phenomenon claimed to be vagueness should be such that the suggestion that there might be a to-the-hair division between bald and not bald seems outlandish. If there is nothing fantastic, there is nothing paradoxical about the sorites argument, and there is no vagueness. Is such a sharp division between the interpretations of R and 'not R' incredible for predicates displaying my post-theoretic phenomena of vagueness? Yes. The reason is that we not only have no experience or capability of drawing a single line between 'bald' and 'not bald', we find the idea that there could be such a line to be absurd. This is because we feel we have cases which are genuinely borderline, being either both bald and not, or neither. This is also linked with *tolerance*, the feeling that natural language predicates must be insensitive to small changes.[5] We feel there is tolerance because we have never experienced sharp lines and we have cases we believe are borderline.

3.4 Ineliminability

Vagueness is not to be easily circumvented, or so it is usually thought, and my theory of vagueness leads to several ways in which it may be said that vagueness is ineliminable. We will see some deeper notions of ineliminability consistent with my theory in Subsection 4.2; this section addresses only the ineliminability emanating from the characterization of vagueness (i), (ii) and (iii). One major notion of ineliminability emanates from the fact that there is nothing particular to us humans assumed in the theory; ideal computing devices such as HAL from *2001 Space Odyssey* and Data from *Star Trek* are subject to vagueness as well. Greater powers of discrimination and computation cannot overcome the dilemma of a borderline region and higher-order vagueness. Why, though, is vagueness ineliminable for them?

There actually is one aspect of vagueness that is in principle eliminable: the overlap borderline region. Given two programs P_1 and P_2 with overlapping interpretations[6] A_1 and A_2, we can build two new programs P_1' and P_2' with interpretations A_1' and A_2' such that A_1' and A_2' do not overlap. $P_1'(x)$ and $P_2'(x)$ each work by simultaneously running $P_1(x)$ and $P_2(x)$. If $P_1(x)$ says YES first, then $P_1'(x)$ says YES and $P_2'(x)$ is made to not halt at all. If $P_2(x)$ says YES before or at the same time as $P_1(x)$, then $P_2'(x)$ says YES and $P_1'(x)$ is made to not halt. A_1' is a subset of A_1, A_2' is a subset of A_2, the union of the primed As is equal to that of the unprimed As, and, finally, A_1' and A_2' are disjoint, i.e., they have no overlap borderline region. When choosing programs in the head, one need just pick two arbitrary ones P_1 and P_2 and then one can obtain from these two derivative programs P_1' and P_2' such that there is no overlap in their interpretations. Alternatively, one may instead work with just one program in the head which outputs YES exactly on the interpretation of R and NO exactly on the interpretation of 'not R'; an overlap region does not have the possible of arising in this framework. Therefore, it is generally possible to interpret natural predicates R and 'not R' in such a way that there is no overlap borderline region. Overlap borderline regions are, then, not inevitable.

This said, let us consider whether the borderline region may be completely eliminated. Once the two programs exist for, say, baldness, perhaps it is possible to find a single new algorithm for baldness that divies up the universe into YES and NO in such a way that anything that is definitely bald (with respect to the two programs) falls into YES, and anything that is definitely not bald falls into NO (i.e., it respects the definite cases). This algorithm would act by classifying each member of the borderline region as either bald or not, and would serve to redefine baldness so as to be non-vague all the while preserving the definite cases (see Figure 2). The algorithm would amount to a precisification of baldness. While we have just observed that such a feat is possible for the overlap borderline region, the gap region

[5] See Dummett [3] and Wright [10]; and see Burns [1] for discussion.
[6] By the "interpretation of a program" I refer to the set of objects on which the program outputs YES.

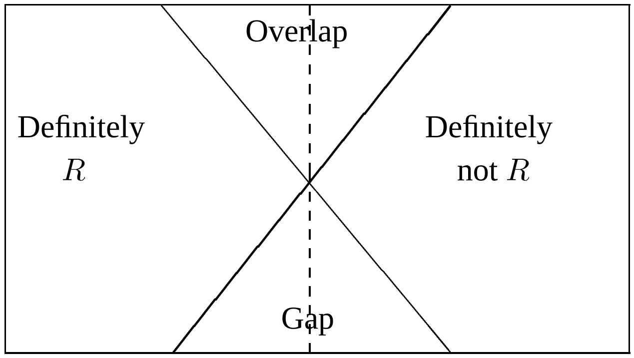

Figure 2: A successful precisification would recursively cut through the borderline region as shown in the dotted line, leaving the definite regions untouched.

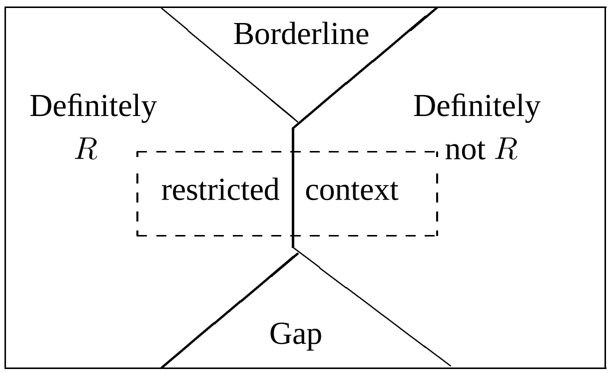

Figure 3: A successful restriction would consist of a recursive subset (a context) consisting of no borderline region as shown by the dotted box.

is not similarly eliminable. If it were generally possible to precisify the gap region and obtain an algorithm for a precisification of baldness, then it would have been generally possible to find such an algorithm in the first place (i.e., pick two programs and then precisify them), contradicting the non-recursiveness of the set of algorithms. Therefore it is not generally possible to eliminate the borderline region, and I call this sort of ineliminability *non-precisifiability*. Also, if the borderline region consists of only an overlap region, then the borderline region is completely eliminable. Since the borderline region is not generally completely eliminable, the borderline region generally has a gap region.

May you carefully restrict yourself to certain well-defined contexts, and within these contexts might vagueness be eliminated? We usually do not think so. For example, we do not seem to find ourselves able to identify a group of people (say, infants) such that baldness is no longer vague amongst that group. My theory explains this sort of ineliminability. What you would like to find is a subset of the universe of objects such that there is no longer a borderline region for baldness; you would like to eliminate vagueness by restricting the context to one where there is no borderline region (see Figure 3). Not just any subset (or context) will do—you need to be able to recognize (via a program in the head) when something is or is not in that subset, and this implies that you need an algorithm. But now you are back to the same old problem yet again: you cannot generally acquire algorithms. Your contexts are not generally decidable by the programs for them. You may then acquire a context which does not include any of the borderline region but be presented with objects for which you are incapable of knowing whether it is in the context. The objects may in actuality not be in the context, but you may then judge them to be borderline cases and thereby see vagueness. Perhaps you only judge an object to be part of the context if the program for the context actually says YES that it is part of the context; if this were so then a single program only semideciding the context is sufficient. The difficulty with this response is that you may well be asked about some object whether it is part of the context and what its categorization is with respect to the vague predicate, and you cannot just refuse to answer. A further difficulty is that contexts very often are natural language concepts; e.g., attractiveness among bald people, or quickness among cats. Therefore, the context itself is vague. Even if you manage to secure a context that is decided by the program for it, because you cannot generally determine where the borderlines are it is not generally possible for you to be assured that the context does not include some of the borderline region. It is not, then, generally possible for you to restrict yourself to a context wherein the borderline region is eliminated, and I call this sort of ineliminability *non-restrictability*.

I have discussed two sorts of ineliminabilities concerning the borderline phenomenon; more varieties of each of these types will be addressed in Subsection 4.2 where I develop a semantics capable of displaying deeper notions of vagueness. Higher-order vagueness is also ineliminable as well. To begin with, it is not generally possible to correct the two programs so that they decide their respect interpretations. If it were possible, the program determining the interpretation of R could be "corrected" to an algorithm in the first place, and there would be no need for a second program in the head at all. But it is not possible to generally acquire an algorithm, and thus it is not possible to so correct the programs.

Although the two programs for baldness cannot determine the boundaries of the borderline region, it is possible for the borderline region to be recursive and thus it is in principle possible for you to have an algorithm deciding it. If you had such an algorithm there would be no more higher-order vagueness since you could determine the boundaries of the borderline region. However, it is not generally possible to find the algorithm. For one, it is not generally possible to pick an algorithm rather than a non-algorithm for the job of attempting to decided the borderline region. And two, you cannot be sure, even given that you have an oracle handing you any desired algorithm, whether what it is deciding is the borderline region since you cannot generally know what things are in the borderline region.

4 Models

I have now concluded my explanation of vagueness, and all that is left is to see what sort of models of vagueness are motivated by the theory. What is the correct logic of our concepts? I have admitted since Section 2 that standard classical two-valued logic (with classical negation) may well be an appropriate model of the true extensions of natural language predicates, since I have been acting as if the true semantics is two-valued. What we want, though, is to know how to model the concepts humans actually have, and this is the topic of this section. The first subsection shows how to capture vagueness in classical logic without classical negation; I provide an axiomatization of vagueness. The second (and final) subsection concentrates on natural language *semantics* only, and I show how to model various depths of vagueness, again using classical semantics.

4.1 Axiomatization of vagueness

As we have observed in Section 2, natural language 'not' is not "usually" translated to the classical negation $\neg$. Instead, 'not R' is translated to its *dual* '$nonR$', a distinct predicate. An adequate logic of "most" of our natural language predicates is, then, a logic without classical negation. I now give a first-order axiomatization of this "most." It is capable of capturing the borderline and higher-order vagueness phenomena emanating from the Undecidability Theory of Vagueness. It is not sufficient to guarantee some of the "deeper" notions of vagueness introduced below in Section 4.2.

Recall my characterization of vagueness: (i) your interpretation of R is determined by a program in your head that is capable of semideciding but not deciding it, (ii) your interpretation of 'not R' is determined by a program in your head that is capable of semideciding but not deciding it, and (iii) your interpretation of R is not the complement of your interpretation of 'not R'. Since it is "usually" the case that your programs in the head are not algorithms, I simplify things and presume that it is always the case that your programs in the head are not algorithms, and furthermore that every predicate of our language is vague. The axiomatization and resulting first order theory (i.e., the set of all theorems deducible from the axioms) are meant only for vague predicates. Because for vague predicates the complements of their interpretations are inaccessible and the natural language negation of R is a distinct predicate $nonR$, the language I choose for my first order theory is a novel one in that the logical symbols are the usual ones ($\wedge$, $\vee$, $\rightarrow$, $=$, $\forall$ and $\exists$) except that the classical negation symbol $\neg$ is excluded. $\wedge$, $\vee$ and $\rightarrow$ are no longer, then, interdefinable as usual (although $A \vee B$ and $(A \rightarrow B) \rightarrow B$ are equivalent); neither are $\forall$ and $\exists$. In addition to these logical symbols, the language L consists of the predicates symbols from the set $\{R_1, nonR_1, R_2, nonR_2, \ldots\}$, as well as many constant symbols as we please. These predicate symbols are to represent the natural language predicates; each natural language predicate R is to be translated to some R_i and each natural language negation 'not R' is to be translated to $nonR_i$. The logical axioms are the usual ones from the predicate calculus minus all those with an occurrence of $\neg$. In addition to the standard rules of inference of Modus Ponens (from ϕ and $\phi \rightarrow \psi$ infer ψ) and Generalization (from ϕ infer $(\forall x)\phi$) are to be added all those rules of inference normally (i.e., when $\neg$ is part of the language) deducible so long as the symbol $\neg$ does not appear in the inference rule itself. One such inference rule, for example, is 'from $\phi \wedge \psi$ infer ϕ.' For every pair $R_i, nonR_i$ the following sentence is a non-logical axiom: $(\exists x)(R_i(x) \wedge nonR_i(x))$. Let V denote the set of theorems deducible from these axioms; I call this theory "V". On the semantics side, all of the familiar notions of truth and validity still apply.

What does theory V have to do with vagueness? Most obviously, the axioms force there to be a borderline region for each predicate pair R_i and $nonR_i$. That is, for any model of V and any i, the interpretations of R_i and $nonR_i$ are not complements; (iii) from the characterization of vagueness is therefore captured. In fact, the interpretations of R and $nonR$ are guaranteed to overlap; the borderline region is guaranteed to have an overlap region but not guaranteed to have a gap region. If you recall from Subsection 3.4, though, the overlap region is eliminable and it is the gap region that is ineliminable. Why have I chosen to force there to be an overlap region, then? The reason is that the gap region is not determinable using the programs since in that region neither program converges. Thus it is not possible to make statements about the gap region, which is the interpretation of the sentence '$\neg R \wedge \neg nonR$'. And this is the sort of reason motivating the banishment of $\neg$ in the first place, since the complements of natural language predicates are not accessible. Given this, I cannot state in my language L that the gap overlap region is non-empty, for in order to do so I would need $\neg$ at my disposal. I still would like a way of satisfying part (iii) of the characterization of vagueness, though, which only requires that the interpretations of R and $nonR$ not be complements, and it suffices for this to use the axiom $(\exists x)(R_i(x) \wedge nonR_i(x))$.

How are (i) and (ii) captured? As just mentioned, it is not possible in V to make statements about the complement of the interpretation of R_i. This is because $\neg$ is not in the language L and no logical combination of the other logical symbols suffices to mimic it. Thus, one can say in V that something is R_i, but not that something is in its complement, and this is exactly the dilemma your program in the head finds itself concerning the interpretation for which it is responsible: the program can say that something is in the interpretation of the predicate, but not generally when something is not. (Note that this axiomatization is a simplification in that it treats the "not generally" as "not".) V captures (i) from the characterization of vagueness, and for the analogous reasons it captures (ii) as well. V is, then, an axiomatization of vagueness. The axioms serve to get (iii), and disallowing $\neg$ serves to capture (i) and (ii).

Let me make clear what I mean through an example. Let R_1 be the predicate symbol for 'bald'. Suppose that I start in the definitely bald region and move toward the borderline region (under any fixed model of V); let the constant symbols $c_0, c_1, \ldots$ be this sorites series, c_0 definitely bald, moving toward borderline as we move down the list of constant symbols. The fact that c_0 is definitely bald is not something that may be stated in V, since to state that something is definitely bald one must say not only that it is in the interpretation of R_1 but also that it is in the complement of the interpretation of $nonR_1$, and this latter is impossible without $\neg$. $R_1(c_0)$ is true, and one is able to say this in the language L by simply asserting that $R_1(c_0)$. $nonR_1(c_0)$ is false, but it is not possible to state this fact in L. For some i, eventually $nonR_1(c_i)$ becomes true, and suddenly it becomes assertable. If it is a borderline case, then now one may assert both $R_1(c_i)$ and $nonR_1(c_i)$. The language L is not capable, though, of stating where the line is at which the borderline region begins.

There is one remaining problem: how can a logic be a logic of natural language without having some role for 'not'? Natural language negation 'not' is, in my account, a function taking a predicate to its dual (from a program in the head to its dual program in the head). 'not', though, sometimes applies to whole sentences in natural language, not just to individual predicates. I let α, defined as follows, play the role of natural language negation.

1. $\alpha(R) = nonR$ and $\alpha(nonR) = R$. (That is, double negations of 'not' cancel.)

2. If $\phi = R(t_1, \ldots, t_n)$, then $\alpha(\phi) = \alpha(R)(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ are terms of L. The same is true for $nonR$.

3. $\alpha(\phi \wedge \psi) = \alpha(\phi) \vee \alpha(\psi)$, $\alpha(\phi \vee \psi) = \alpha(\phi) \wedge \alpha(\psi)$, $\alpha(\phi \rightarrow \psi) = \phi \wedge \alpha\psi$, $\alpha(\forall x\phi) = \exists x\alpha(\phi)$.

The natural language 'not' is to be translated to α, and the resulting sentence will then be without α. For example, "It is not the case that baldness implies shortness" is translated at first to $\alpha\forall x(B(x) \rightarrow S(x))$. α then moves through the sentence just as would $\neg$, leaving us with $\exists x(B(x) \wedge nonS(x))$, which is a sentence of L above (supposing B and S are R_i and R_j for some i and j). One may freely manipulate sentences using α, so that for example the natural language sentence above and "there are people that are bald and not short" are equivalent; however, the latter version is the literal translation from L.

There are some interesting meta-consequences of this logic without $\neg$.

First, there are no contradictions, for the statement of contradictions requires classical negation. Our natural language negation 'not' is not generally correctly interpreted as $\neg$, and confining ourselves only to those parts of natural language where 'not' is not $\neg$, we too are incapable of stating contradictions. Since there are no contradictions the set of every sentence of L must be consistent (otherwise, by the Compactness Theorem, which also holds here, there would be a contradictory sentence). For any set of sentences we can utter there is a possible world making them all true. This gives new meaning to "Never say never" and "Nothing is impossible," and is one explanation for why philosophy is such a slippery fish. That is, just when you think you have backed your colleague into a contradictory corner and you say, "You must, then, hold that p, and p is a clear contradiction," he can always reply, "I am not so sure. Here is a situation in which p obtains..." As philosophers (especially those in ethics, I would estimate) we regularly find ourselves on both sides of this scenario. Note that I have not committed myself to a claim that natural language 'not' *never* is properly translated as classical negation; sometimes when you say "it is not the case that p" you may well mean "$\neg p$." For example, you may say, "Whatever my interpretation of 'bald' is, I am referring to its complement, even if that complement is not my (usual) interpretation of 'not bald'." This logic here developed does not apply in such cases.

Second, not every "regular" valid sentence is a sentence any longer, since many "regular" valid sentences have an occurrence of $\neg$. For example, $(\forall x)(R(x) \vee \neg R(x))$ is not a sentence, and the natural language sentence 'R or not R' is translated instead to $(\forall x)(R(x) \vee nonR(x))$, which is not a logical validity. And although there are no logical contradictions, there are some "regular" logical validities left over. For example, $(\forall x)(R_1(x) \rightarrow R_1(x))$, $(\forall x)((R_1(x) \wedge R_2(x)) \rightarrow R_1(x))$ and $(\forall x)(R_1(x) \rightarrow (R_2(x) \rightarrow R_1(x)))$. The natural language versions of these logical validities are more intuitively valid, I believe, than those of ones with an occurrence of $\neg$, since the latter suffer from borderline cases in ways the former do not.

4.2 Modeling deeper vagueness

The Undecidability Theory of Vagueness's characterization of vagueness was capable of first order axiomatization in V; for any model of V the concepts R_i-ness for each i satisfies the characterization with respect to the language L and so is vague. There are deeper notions of vagueness for which it is possible to provide a semantics, and this is my aim in this section. I do not necessarily claim that vagueness is (or is not) of these deeper kinds; rather my aim here is only to show that a rich semantics of various depths of vagueness is possible. Thus, for this section I am not primarily interested in explanation as I have been previously.

The semantics for vagueness I provide relies on certain computational complexity notions concerning sets of natural numbers (e.g., the non-recursiveness of an interpretation), and the semantics is not first order axiomatizable. This is because in order for a set of axioms to be such that (all and) the only structures modeling it are those of the sort of semantics I will describe, the axioms would have to syntactically encode the information that the interpretation of a certain predicate such as 'bald' must be so and so computationally complex; e.g., recursively enumerable. However, 'recursively enumerable' only is meaningful from inside the model that is arithmetic; nonstandard interpretations of the axioms would mean that the axiom saying " 'bald' is recursively enumerable" no longer means what we thought it means.

The semantics I give is a classical semantics in that each predicate receives a set as its interpretation. The only uniqueness is that rather than baldness being represented by just one set B, there are two sets, B and $nonB$, which together comprise the concept of baldness.

We have seen how the characterization of vagueness acquires a borderline region: it is because your interpretations of R and 'not R' are not complements, which is part (iii) of the characterization. The borderline region is ineliminable as discussed in Subsection 3.4 to the extent that it is not generally possible to avoid (iii) "most" of the time. However, the interpretations of R and 'not R' may well be recursive; in fact, it will "usually" be the case that they are recursive.[7] If they are recursive, then it is in principle possible for the interpretations to each be decided by an algorithm, even though it is not generally possible for you to find the algorithm. If you were to have the algorithm for R, say, you would no longer have need for the interpretation of 'not R'. The characterization of vagueness would fail and there would be no vagueness. One may say that a deeper notion of a borderline region would be where there simply exists no algorithms for deciding the interpretation of R and 'not R'. It is not then possible to be content with just one program for one of the predicates since neither program can possibly be an algorithm. It is possible to give interpretations to R and 'not R' satisfying this: *make each of them recursively enumerable but not recursive.* This semantics might even be the case for some of your concepts—i.e., the interpretations of R and 'not R' might actually be recursively enumerable but not recursive—and there are at least two reasons for thinking so. (a) There are natural recursively enumerable but not recursive mathematical sets and one might begin to wonder why there would not be some natural language ones as well. (b) Although "usually" your interpretations are recursive (as mentioned above), you are not generally capable of avoiding non-recursive interpretations; occasionally, then, they may well occur. Vagueness of this depth I refer to as *depth-1-vagueness*. *depth-0-vagueness* refers to the vagueness accounted for prior to this section, that given by the characterization.

Recall that higher-order vagueness came from the fact that the programs for R and 'not R' semidecide but do not decide their respective interpretations; this emanates from parts (i) and (ii) of the characterization. This sort of higher-order vagueness is of the "depth-0-vagueness" sort. I noted in Section 3.4 that the borderline region could well be recursive,[8] in which case it is in principle possible for you to have an algorithm deciding it. Higher-order vagueness would be eliminated were you to have such an algorithm and use it in this way. I noted also in Subsection 3.4 that even when the borderline region is recursive it is not generally possible for you to acquire an algorithm for it, but one might be inclined to say that there is a sense in which it is eliminable in that it is possible, in principle, to possess an algorithm deciding it. It is possible to give interpretations to R and 'not R' so that the higher-order vagueness of the borderline region is ineliminable: *choose their interpretations so that their borderline region (the overlap region unioned with the gap region) is not recursive*. Vagueness of this depth I call *depth-2-vagueness*.

In Subsection 3.4 I discussed precisifiability, which was the ability to modify the interpretations of R and 'not R' so that the definite regions are unaffected but the new interpretations are complements; intuitively, a precisification draws a line down through the borderline region, and does so in such a way that an algorithm can decide the new concept. It was noted in Subsection 3.4 that it is not generally possible to so eliminate the borderline region since if it were, one could generally acquire algorithms, which is impossible; this is the depth-0-vagueness notion of non-precisifiability. One might say, though, that it is still in principle possible for there to be, in some cases, a precisification; a deeper notion of non-precisifiability would disallow even this. It is possible to interpret R and 'not R' in such a way that there exists no precisification: *choose their interpretations so that the definite regions are recursively inseparable.*[9] In such cases it is truly in principle impossible for a finite agent to precisify the concept. Vagueness of this depth I call *depth-3-vagueness*.

Depth-i-vagueness is a strict hierarchy. That is, for $i < j$, depth-j-vagueness implies depth-i-vagueness but not vice versa. The proof of this is given in Appendix A as Theorem 1. Also, there are predicates having each of these sorts of vagueness (see Theorem 2 in Appendix A for the existence of depth-3-vagueness; Theorem 1 possesses the information implying the existence of the others).

Finally I move to the semantics of deeper non-restrictability. Restrictability was the idea of focusing just on some recursive subset of the universe such that within the context there is no longer any borderline region. Recall from Subsection 3.4 we saw that the characterization of vagueness leads to its own depth-0-vagueness notion of non-restrictability; it is not generally possible for you to acquire algorithms for such contexts, and even with this problem ignored you are incapable of being assured that the context does not include some of the borderline. Still, there are cases where it is in principle possible to have such a restriction, and if one could do this, one would have ridded oneself of vagueness

[7] This is captured by the fact that the set of recursive sets is Σ_3, and the set of recursively enumerable but not recursive sets is Π_3. The relative difficulty of acquiring recursively enumerable but not recursive sets versus recursive sets is analogous to the relative difficulty of determining cases where a program halts versus when it does not.

[8] In fact, since the interpretations are "usually" recursive (see earlier footnote in this section) and the set of recursive sets is closed under intersection, union and complementation, the borderline region is "usually" recursive.

[9] Two disjoint sets A and B are *recursively inseparable* if and only if there is no recursive set C such that $A \subseteq C$ and $B \cap C = \emptyset$.

within that context. I cannot possibly devise a semantics that makes this impossible for all possible restrictions; restrictions to finite contexts cannot be stopped. These are not interesting, though. Any interesting context will be infinite, and it is possible to interpret R and 'not R' in such a way that there exists no such restriction to an infinite, recursive context: *choose their interpretations to be simple sets.*[10] In such cases it is truly in principle impossible for a finite agent to "restrict away" the vagueness with an infinite context. Vagueness of this depth I call *unrestrictable vagueness*. (Theorem 3 in Appendix A shows that there are predicates that are unrestrictably vague.) When R and 'not R' are interpreted as simple sets there must be a non-empty overlap region; otherwise the interpretation of R, which is recursively enumerable, is a subset of the complement of the interpretation of 'not R', contradicting the simple-ness of the latter interpretation.

The relationships between unrestrictable vagueness and depth-i-vagueness for each i is not completely known. It is only known that there are cases of unrestrictable vagueness that are not cases of depth-3-vagueness: let the interpretations of R and 'not R' be simple sets with union equal to the naturals (that there is one see Rogers [5]). Since the borderline region is all overlap it is eliminable as discussed in Subsection 3.4.

The following summarizes the semantics of vagueness introduced in this section. Depth-0-vagueness is not actually an entirely semantic notion; part (iii) of the characterization is, requiring that the interpretation of R and 'not R' not be complements, but parts (i) and (ii) explicitly refer to the programs you happened to have chosen. The other sorts of vagueness are user-independent; so long as the user is Church-bound the vagueness of that depth will be experienced.

Vagueness Type	R and 'not R' interpreted so that they. . .
Depth-0	. . . satisfy the characterization ((i), (ii) and (iii)).
Depth-1	. . . are recursively enumerable but not recursive.
Depth-2	. . . have a non-recursive borderline region.
Depth-3	. . . have recursively inseparable definite regions.
Unrestrictable	. . . are simple sets.

5 Conclusion

I have demonstrated how a particularly weak model of human cognition—that humans can compute all and only the Turing-computable—leads to a characterization of vagueness ((i), (ii) and (iii)) (Section 2). The characterization leads to many of the central aspects of vagueness (Section 3). Finally, the characterization is shown to be axiomatizable and deeper notions of vagueness can be modeled in classical two-valued logic. Current weaknesses include the degree to which the models of vagueness are idealized. Future work might increase their complexity to accommodate natural language 'not' sometimes being translated to '$\neg$' and other times to 'non'.

A Proofs

Theorem 1 *1. Depth-1-vagueness implies depth-0-vagueness but not vice versa.*

2. Depth-2-vagueness implies depth-1-vagueness but not vice versa.

3. Depth-3-vagueness implies depth-2-vagueness but not vice versa.

Proof. To prove 1, realize that if the interpretation of R and 'not R' are recursively enumerable but not recursive, then (i) the program for R can semidecide but not possibly decide the interpretation of R, (ii) the program for 'not R' can semidecide but not possibly decide the interpretation of 'not R', and (iii) their interpretations cannot possibly be complements since if two recursively enumerable sets are complements then they are recursive; but they are by supposition not recursive. Thus, depth-1-vagueness implies depth-0-vagueness. The other implication does not hold since (i), (ii) and (iii) may well hold and yet the interpretations be recursive.

[10] A set is *simple* if it is recursively enumerable and its complement has no infinite recursively enumerable subsets. Post [4] first used simple sets in his search for, intuitively, easy recursively enumerable but not recursive sets. See Soare [6] for a modern account of them.

To prove 2, the implication easily holds since if it is not the case that depth-1-vagueness, then this means the interpretations of R and 'not R' are (or might well be) recursive, in which case the borderline region is recursive. To see that the counter-implication does not hold consider a set A and its dual set $nonA$ as described below; these are the interpretations of R and 'not R', respectively. A and $nonA$ will be constructed so that they are recursively enumerable but not recursive—i.e., they display depth-1-vagueness—yet the borderline region is still recursive. Let C be any recursively enumerable but not recursive subset of $\{n|n \text{ is divisible by } 3\}$. Let $A = C \cup \{3n+1|n \in \omega\}$, $nonA = C \cup \{3n+2|n \in \omega\}$. A and $nonA$ are recursively but not recursive. Note that $A - nonA = \{3n+1|n \in \omega\}$ and $nonA - A = \{3n+2|n \in \omega\}$, and each of these is recursive. But $(A - nonA) \cup (nonA - A)$ is the complement of the borderline region of A and $nonA$, $(A \cap nonA) \cup (\neg A \cap \neg nonA)$, so the borderline region of A and $nonA$ is recursive.

Proving 3, let us begin with the counter-implication and show that it does not hold. Take the proof of statement 2's counter-implication and everywhere modify $\{3n+1|n \in \omega\}$ to any recursively enumerable but not recursive subset $D_1 \subset \{3n+1|n \in \omega\}$, and $\{3n+2|n \in \omega\}$ to any r.e. but not recursive subset $D_2 \subset \{3n+2|n \in \omega\}$ ($D_1 \cup D_2$ must be r.e. but not recursive). Now, (a) the borderline region is not recursive, since if it were, its complement $D_1 \cup D_2$ would be recursive, which is a contradiction; thus depth-2-vagueness is displayed. And (b), the definite regions are recursively separable by $E = \{3n+1|n \in \omega\}$, so depth-3-vagueness is not displayed.

Proving the implication in 3, I show its contrapositive. Fix any depth-1-vague but not depth-2-vague borderline region (if the borderline region is depth-0-vague only, then the interpretation of R is recursive and it can be used as the precisification). Let B be the recursive borderline region. The following program decides the 'definitely R' region, $A - nonA$, where A and $nonA$ are the interpretations, respectively, of R and 'not R':

Begin Program P.

1. Input(x).
2. If $x \in B$, then output('NO') and halt. Otherwise continue.
3. Dovetail enumerate through A and $nonA$ (this is possible since you have programs for them) until one of them finds x (which must happen since $x \notin B$).
4. If $x \in nonA$, then output('NO') and halt. Otherwise output('YES') and halt.

End Program P.

If $x \in A - nonA$, then $x \notin B$, so program P will eventually halt and the program for $nonA$ never will. In this case $P(x) =$'YES.' Alternatively, if $x \notin A - nonA$, then either $x \in B$ or $x \in nonA - A$. If the former, $P(x) =$'NO.' If the latter, then $x \notin B$ and so $P(x) =$'NO.' Therefore, $x \in A - nonA$ if and only if $P(x) =$'YES,' so P decides $A - nonA$. We may now simply let our recursive separator be $A - nonA$, and R-ness is not depth-3-vague. □

Theorem 2 *There exists a depth-3-vague predicate.*

Proof. Let disjoint D and E be recursively inseparable recursively enumerable but not recursive subsets of $\{3n|n \in \omega\}$, and let $A = D \cup \{3n+1|n \in \omega\}$ and $nonA = E \cup \{3n+1|n \in \omega\}$. If R and 'not R' are interpreted, respectively, as A and $nonA$, then R-ness is depth-3-vague. □

Theorem 3 *There exists an unrestrictably vague predicate.*

Proof. Fix A and $nonA$ simple sets as the interpretations, respectively, of R and 'not R'. Pick any infinite recursive X such that $A \cap X$ and $nonA \cap X$ are infinite; X is the context. The borderline region of R in the restricted domain is infinite because if not, $(A - nonA) \cap X$ (the restricted definitely R region) is recursively enumerable and is an infinite subset of the complement of $nonA$, contradicting the simplicity of $nonA$. □

References

[1] L. C. Burns. *Vagueness: An Investigation into Natural Languages and the Sorites Paradox*. Kluwer, 1991.

[2] M. A. Changizi. Vagueness and rationality. Article under review, September 1997.

[3] M. Dummett. Wang's paradox. *Synthese*, 30:301–324, 1975.

[4] E. Post. Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society*, 50:284–316, 1944.

[5] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, New York, 1967.

[6] R. I. Soare. *Recursively Enumerable Sets and Degrees*. Springer Verlag, New York, 1987.

[7] D. A. Thorpe. The sorites paradox. *Synthese*, 61:391–421, 1984.

[8] P. Unger. There are no ordinary things. *Synthese*, 41:117–54, 1979.

[9] S. C. Wheeler. On that which is not. *Synthese*, 41:155–194, 1979.

[10] C. Wright. On the coherence of vague predicates. *Synthese*, 30:325–365, 1975.