

MARK A. CHANGIZI

## VAGUENESS, RATIONALITY AND UNDECIDABILITY: A THEORY OF WHY THERE IS VAGUENESS

### 1. INTRODUCTION

Vagueness is not undecidability, but undecidability does enter into an explanation of why there is vagueness. My theory, called the Undecidability Theory of Vagueness, explains vagueness largely as a result of the fact that we are computationally bound.<sup>1</sup> Vagueness is not due to any particularly human weakness, but due to a weakness that any computationally bound agent possesses; even HAL from *2001: A Space Odyssey* will probably experience vagueness. Furthermore, I will argue that vagueness is good for you. I will do so by showing that if you were a computationally bound, rational alien agent given the task of figuring out what our natural language predicates mean, you would very probably end up with vagueness. That is, unless two highly plausible hypotheses are false, it would be rational for you – i.e., in your best interest – to choose concepts in such a way that they are vague. Given that you are computationally bound, avoiding vagueness brings in greater costs than accepting it.

It is useful here in the introduction to present a brief, preliminary run-through of my theory and how it explains vagueness. The “vagueness is good for you” arguments will not appear in this introduction. I will take you to be my example natural language user.

(1) The first hypothesis is the *Church-Bound Hypothesis*, and it states that you can compute no more and no less than what a computer can compute. (2) The second hypothesis is the *Programs-in-Head Hypothesis*, and it states that what natural language predicates extensionally mean to you is determined by programs in your head. For example, an object is a dog to you if and only if your program in the head for ‘dog’ outputs YES when the (name of the) object is input into the program. It is much less plausible that many scientific, mathematical and technical predicates get their meaning to you via programs in the head, and this difference is what prevents my theory from concluding that such predicates, many which are not vague,



are vague. (3) The third and last hypothesis is the *Any-Algorithm Hypothesis*, and it states that you allow yourself the choice of any algorithm when choosing programs in the head for determining your natural language predicate meanings. (An *algorithm* is a program that halts on every input; programs sometimes do not halt on some inputs.) Informally and crudely, the three hypotheses are that (1) you are a computer, (2) you have programs in the head determining what natural language predicates mean to you, and (3) you allow yourself the fullest range of possible meanings for natural language predicates.

If these three hypotheses are true, what follows? The Programs-in-Head Hypothesis says you choose programs to determine your meanings of natural language predicates. The Any-Algorithm Hypothesis says that the set of programs from which you are choosing is a superset of the set of all algorithms. But here is the catch: one of the basic undecidability results implies that any such set of programs is undecidable. (A set is *decidable* if and only if there is program that outputs YES whenever input with an object from the set and NO whenever input with an object not in the set.) Because of the Church-Bound Hypothesis, this undecidability is a difficulty for you: in choosing from the set of programs you cannot always obtain algorithms. In fact, because picking algorithms is computationally more difficult than picking non-algorithms, you will “usually” pick non-algorithms; “most” of your programs determining the meanings of natural language predicates will not be algorithms. So, in an attempt to acquire a meaning for ‘dog’ via a program in the head that outputs YES when something is a dog to you and NO when something is not a dog to you, *there will be objects on which your program does not halt at all*. This does not mean that you will actually run into an infinite loop; it just means that you will eventually give up when running the program on such inputs.

What does this have to do with vagueness? Consider the set of objects for which the program for ‘dog’ does not halt. For any object in this set the program will neither say YES nor NO; the object will neither be a dog to you nor not a dog to you. My first theoretical claim is that this is the set of borderline cases for the predicate. What about higher-order vagueness, the phenomenon that the boundaries of the borderline region are vague? Consider trying to determine exactly which objects are part of the borderline region. To determine that some object is in the borderline region of ‘dog’ requires that you determine that your program for ‘dog’ does not halt on that object. But now we have another catch: possibly the most well-known undecidability result is the “halting problem”, which says that whether or not a program will halt on a given input is undecidable. This undecidability is a difficulty for you because of the Church-Bound