Katie Prescott

Automata and Computation

Professor Katz

9 May 2016

Rice’s Theorem

Software testing problems usually start by asking you to “decide whether the function is…” Rice’s Theorem proves that all decision problems on a Turing Machine are undecidable (i.e. halting, emptiness, equality, accepting, etc.); you cannot use an algorithm to decide anything about the properties of a given program. The only exceptions to this are the trivial properties, which are either always true or always false.

The theorem states that there is some sort of functional property that you are testing among a set of languages. This property describes how the input and output relate to each other. It also must be present in some programs and not present in others. Rice’s Theorem proves that, no matter what it is, any and all functional properties are undecidable. A property is trivial if that property contains *every* Turing Machine, or if it contains *no* Turing Machines, making the property either always true or never true, respectively.

In order to prove Rice’s Theorem, the property is reduced to the acceptance problem. Property P describes some language recognized by a Turing Machine if L(M) = L(N), meaning P contains <M> (the encoding of M) if and only if P also contains <N>. To prove, you must first assume some property P is decidable, which means there is some Turing Machine B that halts and recognizes the Turing Machine descriptions that satisfy P. You use this assumption to construct Turing Machine A that accepts {(M, w) | M is a Turing Machine that accepts string w} as its language. To do that, you let MP be a Turing Machine that satisfies P. The algorithm for A is as follows:

1. On (M, w) as input, create C(M, w):
   1. On input x, let M run string w until it accepts or runs forever
   2. Run MP on x
      1. Accept if and only if MP accepts.
      2. Otherwise, reject
2. Feed the description of C(M, w) to B
   1. If B accepts, then accept
   2. Otherwise, reject.

NOTE: if M accepts w, then L(C) = L(MP), and if M accepts w, then C(M, w) has property P

Since the accepting problem, as demonstrated in this proof by C(M, w), is undecidable, there does not exist a Turing Machine B that can decide property P. Therefore, P is undecidable.

Although Rice’s Theorem is a powerful rule for Turing Machines, it only states that the structure of a language that a Turing Machine accepts is undecidable, it does not state that everything about Turning Machines is undecidable. Rice’s Theorem cannot be used when testing machine-dependent properties, such as the amount of time it takes for a Turing Machine to run or how many times a particular cell on the tape is scanned. There is still no general rule for these questions.