The connection between program-size complexity and algorithmic probability:

\[ H(x) = -\log_2 P(x) + O(1). \]

Occam’s razor: there are few minimum-size programs

Introduction

The first half of the main theorem of this chapter is trivial:

\[ P(x) \geq 2^{-H(x)}, \]

therefore

\[ -\log_2 P(x) \leq H(x). \]

So now we only need to show that

\[ H(x) \leq -\log_2 P(x) + c. \]

In this chapter we’ll actually prove that

\[ H(x) \leq -\log_2 P_C(x) + c \]

where \( c \) depends only on the computer \( C \), not on the S-expression \( x \). Then the special case \( C = U \) will complete our proof that

\[ H(x) = -\log_2 P(x) + O(1). \]
Why is this result important? Because:

1. We need it for a crucial lemma in the next chapter, which will enable us to understand relative complexity. So we need the main theorem of this chapter in order to be able to lay bare what relative complexity is all about. In turn, we need next chapter’s result on relative complexity in order to use it at the beginning of Part III to understand what a random bit string is.

2. This chapter’s result shows that AIT is what you get when you take logarithms of probabilities. In other words,

\[ \text{AIT} \approx -\log_2 \text{Probability Theory !!!} \]

So statements in probability theory dealing with \textbf{products} of probabilities correspond to results in AIT dealing with \textbf{sums} of complexities. And divisions turn into subtractions... This gives you a feeling for what AIT is like, in a purely formal way, if you’re familiar with the formulas in probability theory. Of course the \textbf{meaning} of these formulas is completely different! Normal probability theory says nothing about computer programs.

3. Occam’s razor, a principle in medieval philosophy, states that “entities should not be multiplied”, that the best theory is a simple theory. We’ll see that minimum-size programs are essentially unique, that there can’t be too many of them, and that you can easily get one from another. Let’s do that right now!

\textbf{An application: Occam’s razor—There are few minimum-size programs}

Here is a very interesting immediate corollary of the main theorem of this chapter. Later in this chapter we’ll show that

\[ H(x) \leq -\log_2 P(x) + c. \]

This will imply that the number of minimum-size programs \( p \) for the S-expression \( x \) is always \( \leq 2^c \):

\[ \#\{ p : U(p) = x \ \& \ |p| = H(x) \} \leq 2^c. \]