Lecture Notes 4

Qiwu Zhou

Suppose we want to send the first 10^9 digits of π to a friend. How to do it? Of course we can send it directly but this method takes a long time, for the string we want to send is extremely long. Since $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} \pm \cdots$, we can write a computer program and use this formula to output the first 10^9 digits of π. Then we just send the code of this program to our friend and ask him or her to run it to get the digits. It is clear that the new method only spends a little time on sending the message. This is exactly the basic idea of Kolmogorov complexity which uses the length of a computer program to describe the complexity a string.

1 Universal Turing Machine

To formalize the notion of the length of computer programs, we need the concept of *Turing machine*. Roughly speaking, a Turing machine $\mathcal{M}$ gets a input string x and either runs eternally or outputs a string y after some time. If $\mathcal{M}$ outputs a string eventually, we say $\mathcal{M}$ halts on input x and use $\mathcal{M}(x)$ to denote the output string. The internal description of Turing machine is not important here. What we need to know is that

1. Every real world program can be described by a Turing machine and different programs correspond to different Turing machines.

2. Under some coding rule, every Turing machine $\mathcal{M}$ is just a string in $\{0,1\}^*$ and different Turing machines correspond to different strings. We abuse the notation and use $\mathcal{M}$ to denote both the Turing machine itself and the string it corresponds to. For a string $s \in \{0,1\}^*$ we use $|s|$ to denote its length.

Hennie and Stearns [1] proved the existence of a special Turing machine.

Theorem 1. *There exists a Turing machine $\mathcal{U}$ such that for every Turing machine $\mathcal{M} \in \{0,1\}^*$ and $x \in \{0,1\}^*, \mathcal{U}(\mathcal{M}, x) = \mathcal{M}(x)$, where $\mathcal{U}(\mathcal{M}, x)$ is the output string of $\mathcal{U}$ while it gets $\mathcal{M}$ and x as input string. In other words, we can use $\mathcal{U}$ to simulate the running procedure of $\mathcal{M}$ so we call $\mathcal{U}$ a* universal Turing machine. *When x is the empty string we use $\mathcal{U}(\mathcal{M})$ to denote $\mathcal{U}(\mathcal{M}, x)$.*

2 Kolmogorov complexity

Definition 2. *The* Kolmogorov compelexity $K_{\mathcal{U}}(x)$ *of a string $x \in \{0,1\}^*$ with respect to a universal Turing machine $\mathcal{U}$ is defined as*

$$K_{\mathcal{U}}(x) = \min_{\mathcal{M}:\mathcal{U}(\mathcal{M})=x} |\mathcal{M}|.$$

It's easy to see the string mentioned at the beginning has a small Kolmogorov complexity with respect to some universal Turing machine $\mathcal{U}$. An important property of Kolmogorov complexity is that it's independent of the universal Turing machine we choose up to an additive constant.

Theorem 3. *If $\mathcal{U}$ is a universal Turing machine, for any other universal Turing machine $\mathcal{U}'$ there exists a constant $c_{\mathcal{U}'}$ such that*

$$K_{\mathcal{U}}(x) \leq K_{\mathcal{U}'}(x) + c_{\mathcal{U}'}$$

for all strings $x \in \{0,1\}^*$ and the constant does not depend on x.

Proof. For every Turing machine $\mathcal{M}$ such that $\mathcal{U}'(\mathcal{M}) = x$ we can construct a new Turing machine by just concatenating the string $\mathcal{U}'$ and $\mathcal{M}$ to get a new string $\mathcal{U}'\mathcal{M}$. This string is of length $|\mathcal{U}'| + |\mathcal{M}|$. Since $\mathcal{U}$ is a universal Turing machine we can use $\mathcal{U}$ to simulate the Turing machine $\mathcal{U}'\mathcal{M}$, i.e., we have $\mathcal{U}(\mathcal{U}'\mathcal{M}) = x$. Let $c_{\mathcal{U}'} = |\mathcal{U}'|$ and $K_{\mathcal{U}}(x) \leq K_{\mathcal{U}'}(x) + c_{\mathcal{U}'}$. □

Because of this property, from now on we fix a universal Turing machine $\mathcal{U}$ and use $K(\cdot)$ to denote $K_{\mathcal{U}}(\cdot)$. Another important property of Kolmogorov complexity is unboundedness. That is to say we can always find strings with arbitrarily large Kolmogorov complexity.

Theorem 4. *For every natural number n, there exists a string s such that $K(s) \geq n$.*

Proof. The number of all programs with length less than n is

$$2^0 + 2^1 + \cdots + 2^{n-1} = 2^n - 1.$$

Since each program can produce only one possible output string, the number of strings with Kolmogorov complexity less than n is finite. There are infinitely many strings so there must be a string with Kolmogorov complexity greater than $n - 1$. □

Although Kolmogorov complexity offers a good description of the complexity of a string s, $K(s)$ is not computable.

Theorem 5. *There is no Turing machine $\mathcal{M}$ which computes the Kolmogorov complexity $K(\cdot)$ for all strings.*

Proof. Suppose there is a Turing machine $\mathcal{M}$ which computes $K(\cdot)$. We construct a new Turing machine $\mathcal{M}'$ as follows. $\mathcal{M}'$ first enumerate all natural numbers n increasingly and for every n enumerate all the strings s of length n. Then $\mathcal{M}'$ call $\mathcal{M}$ to compute $\mathcal{M}(s)$. If $\mathcal{M}(s) > |\mathcal{M}'|$ then $\mathcal{M}'$ output s and halts.

Since $\mathcal{M}'$ is also a Turing machine, $|\mathcal{M}'|$ is of finite length. By theorem 4, there is a string s with $K(s) > |\mathcal{M}'|$. Hence, $\mathcal{M}'$ must halt and output a string s. Because $\mathcal{M}'$ outputs s, we know that $K(s) < |\mathcal{M}'|$. However, $\mathcal{M}'$ outputs s if $K(s) = \mathcal{M}(s) > |\mathcal{M}'|$. This gives a contradiction. □

References

[1] F. C. Hennie and Richard Edwin Stearns. Two-tape simulation of multitape turing machines. *J. ACM*, 13(4):533–546, 1966.