

39.3 Space/Time-Bounded Compression

As described in the previous chapter, it is impossible to write the "perfect" compression algorithm that requires the fewest bits to output some bitstream B .

Space-Bounded Compression

However, what about the problem of space-bounded compression? In this problem, we take in two inputs: a bitstream B and a target size S . The goal, then, is to find a program of length $\leq S$ that outputs B .

It turns out that such a problem is also uncomputable. If it were, then we could simply binary search on different values of S to find the optimal compression program size, which is impossible as shown in the previous section.

Space-Time-Bounded Compression

What if we take our problem from above, and add a constraint that we can run at most T lines of bytecode?

It might seem unintuitive, but this kind of problem is actually solvable. We will use the following algorithm:

```
for length L = 1....S:
    for each possible program P of length L:
        while (P is running && !(B is outputted) && lines_executed < T):
            run the next line of P
```

The runtime of this algorithm is $O(T * 2^S)$, and in the end, it will either output some program `P` that has the correct output and is bounded by T and S , or return that no such program is possible.

Efficient Bounded Compression

The runtime above is exponential in S . Thus, we might ask if it's possible to solve the space-time-bounded compression problem *efficiently*. As we'll see in the next chapter, this depends on our definition of efficiency.

[Previous](#)
39.2 Optimal Compression, Kolmogorov Complexity

[Next](#)
39.4 P = NP

Last updated 1 year ago

