

1 Time is at most exponential in space

Theorem 1. $\text{SPACE}[S] \subseteq \text{TIME}[n \cdot 2^{O(S)}]$.

Proof. Let $L \in \text{SPACE}[S]$, decided by a space- S machine M . We prove that M runs in time at most $n \cdot 2^{O(S)}$.

Fact 1.1. For any $x \in \{0,1\}^n$, the number of possible different configurations in the computation of $M(x)$ is at most $n \cdot 2^{O(S)}$.

Proof. We claim that any configuration can be described by a binary string of length $O(S) + \log(n)$. Recall that a configuration consists of the worktape content, the locations of the input head and the worktape head, and the state of the machine. To describe the worktape content we need to describe at most S symbols from the constant-sized alphabet, which can be done using $O(S)$ bits; and to describe the two head locations we can use $\lceil \log(n) \rceil + O(\log(S))$ bits; and to describe the internal state we use $O(1)$ bits. Overall, the configuration's binary description is of length $O(S) + \log(n) + \log(S) + O(1) = O(S) + \log(n)$.

Since the number of binary strings of length $O(S) + \log(n)$ is at most $n \cdot 2^{O(S)}$, the number of possible different configurations is at most $n \cdot 2^{O(S)}$. \square

Fix $x \in \{0,1\}^n$, and let $C_1, C_2, C_3, \dots, C_T$ be the sequence of configurations in the computation of $M(x)$. Assume towards a contradiction that $C_i = C_j$ for some $i < j$. Then, when starting from $C \stackrel{\text{def}}{=} C_i$ and applying the transition function of M we reach the same configuration $C_j = C$. But this means that when applying the transition function from $C_j = C$ again we'll reach C_j again; repeating this argument, M never halts! That's a contradiction to the assumption that M decides L .

Hence, in the sequence C_1, \dots, C_T each configuration appears at most once. Since there are only $n \cdot 2^{O(S(n))}$ possible different configurations, $T \leq n \cdot 2^{O(S(n))}$. \blacksquare