Levin's Lower Bound Theorem

**Theorem:** Let $f$ be a total recursive function. Then there exists a total recursive function $g$ such that, for any space function $s$ such that $s(x) = \Omega(\log |x|)$

$$f \in \text{DSPACE}(s) \iff s = \Omega(g)$$

**Proof:** Let $f$ be given. Let $C = \{s : s$ is a space function and $f \in \text{DSPACE}(s)\}$. Our goal is to build a total recursive $g$ so that for any space function $s$ such that $s(x) = \Omega(\log |x|)$, $s \in C \iff s = \Omega(g)$.

Our first step will be to characterize $C$. We will build a sequence of functions $p_1, p_2, \ldots$ such that:

1. $\forall i, p_i \in C$.
2. $s$ a space function in $\Omega(\log) \implies [s \in C \iff \exists i \ s = \Omega(p_i)]$ (and in fact, for all large $i$, $[M_i$ runs in space $s$ and computes $f] \implies s = \Omega(p_i)$).
3. The function $(i, x) \rightarrow p_i(x)$ is recursive.
4. $\forall i \ p_i \geq p_{i+1}$.

The best way to describe these functions $p_1, p_2, \ldots$ is to give a program for the machine $N$ which, on inputs $i$ and $x$, computes $p_i(x)$, (thus demonstrating that property (3) above holds).

Let $f$ be computed by some machine $M_k$ in space $s_k$. (In general, the space bound of the $i$-th TM will be denoted by $s_i$.) Here is the program for $N$:

```
begin
    On input $(i, x)$,
    Mark off $\log |x|$ tape.
    Using at most $\log |x|$ tape, try for each $j \leq i$ to
determine if there is a $y$ such that $M_j(y) \neq f(y)$. Mark
all such $j$ as “cancelled”.

    Let $A = \{j \leq i : j$ has not been marked as “cancelled”\}.
    For $c := 1$ to $\infty$
        For all $j \in A \cup \{k\}$
            Try to simulate $M_j$ on $x$ using $c$ of $M_j$’s tape cells.
            If any of these simulations halts, halt and output max($c, (\log |x|)$).
end
```

Let us show that condition (1) holds: i.e., for all $i, p_i \in C$. Note that for all $j < i$ such that $M_j$ does not compute $f$, we eventually cancel $j$, and thus for all large $x$, $j$ gets cancelled during the first part of the computation on input $(i, x)$. Thus, for all large $x$, during the second part of the computation on input $(i, x)$, we are simulating only machines that compute $f$. Notice that the simulation never uses more than $ip_i(x)$ tape cells (assuming without loss of generality that, for all $j$, $M_j$ can be simulated using at most $js_j$ space). Notice also that the
simulation always us to successfully run some machine computing $f$. By the linear speed-up theorem, $f$ can be computed in space $p_i$. (The reader may verify that, in addition, $p_i$ is a space function.)

Conditions (2) and (4) are easily verified. That is left to the reader.

Now it will suffice to construct our function $g$ so that the following two conditions are satisfied:

(a) $\forall i \ p_i \geq g$ a.e.

(b) For all space functions $s_i$ such that $s_i(x) = \Omega(\log |x|)$,

$$\exists^* x \ s_i(x) < p_i(x) \implies \exists y > i \ s_i(y) < g(y).$$

Before we show that properties (a) and (b) hold, let us prove that they are sufficient to prove the claim.

It is clear (by condition (2)) that if $f$ is computable in space $s$, then $\exists i \ s = \Omega(p_i)$. Since by (a), $p_i \geq g$ a.e., it follows that $s = \Omega(g)$.

Conversely, let $s$ be a space function, $s(x) = \Omega(\log x)$, such that $f$ is not computable in space $s$. We need to show that, for all $k$, $k s(x) < g(x)$ for infinitely many $x$. By the linear speed-up theorem, we know that $f$ is not computable in space $ks$. It follows from (2) that $\forall i \exists^* x \ ks(x) < p_i(x)$. There are infinitely many machines $M_{j_1}, M_{j_2}, \ldots$ that have space complexity $ks$. (This is a property of most reasonable indexing systems; assume without loss of generality that we are using some such “reasonable” indexing.) That is, for all $x$, $ks(x) = s_{j_1}(x) = s_{j_2}(x) = \ldots$ Thus $\forall r \forall i \exists^* x \ s_{j_r}(x) < p_i(x)$. By property (b), $\forall r \exists y > j_r s_{j_r}(y) < g(y)$. Since $ks = s_{j_r}$, it follows that $ks(y) < g(y)$ for infinitely many $y$.

Now that we know that (a) and (b) are sufficient, let us show how to compute $g$.

**Begin**

On input $x$

Mark off $\log |x|$ space.

For $i := 1$ to $\log |x|$

Attempt to determine (using at most $\log |x|$ space) if there is any $y$ such that $i < y < x$

such that $s_i(y) < g(y)$. (Call any such $i$

“cancelled”.)

Choose the least uncalled $i$ such that $s_i(x) < p_i(x)$,

and cancel $i$ and set $g(x) = p_i(x)$. (I.e., output $p_i(x)$.)

(If there is no such $i$, set $g(x) = p_{\log |x|}(x)$.)

**End**

To verify that (b) holds, let us assume that $\exists^* x \ s_i(x) < p_i(x)$. We need to show that there is some $y > i$ such that $s_i(y) < g(y)$. Let $R$ be the set of all $j < i$ that ever get cancelled. It can be verified that all $j \in R$ get cancelled, using at most $k$ tape cells, for some $k \in \mathbb{N}$. Choose $x$ so that $k < \log |x|$ and
$x > i$ and $s_i(x) < p_i(x)$. (By assumption, such an $x$ exists.) On input $x$, either $i$ is already cancelled (in which case there is some $y > i$ such that $s_i(y) < g(y)$), or $i$ will be the least uncanceled index such that $s_i(x) < p_i(x)$ and we will set $g(x) = p_i(x) > s_i(x)$, and we cancel $i$.

To verify that (a) holds, we need to show that for all $i$ and for all large $x$, $p_i(x) \geq g(x)$. Again, let $R$ be the set of all $j < i$ that ever get cancelled. For all large $x$, it can be verified on input $x$ that everything in $R$ gets cancelled, and thus, for all large $x$, there is some $j > i$ such that $g(x) = p_j(x)$. Since, by property (4), $j > i \implies p_j(x) \leq p_i(x)$, we have that for all large $x$, $g(x) \leq p_i(x)$.

That completes the proof.

The only English-language reference I have found for this result is: Leonid A. Levin, *Computational Complexity of Functions*, in Boston University Technical Report BUCS-TR-85-005, 1985. This is a one-page partial translation of the original paper: Leonid A. Levin, *Complexity of Algorithms and Computations*, Ed. Kosmidiadi, Maslov, Petri, “Mir”, Moscow, 1974, 174–185. This (Russian) paper seems to be the only place that the proof has been published. Similar (although somewhat weaker and much messier) results may be found in A. R. Meyer and K. Winklmann, *The Fundamental Theorem of Complexity Theory, (preliminary version)*, in Ed. J. W. DeBakker and J. van Leeuwen, Mathematical Center Tracts 198 (1979), 97–112. Much of the proof in this handout is based on proofs in Meyer and Winklmann.