**Provable Intractability**

Our aim now is to show that there are languages (or, equivalently, decision problems) that we can prove are not in \( P \).

This is done by showing that, for every reasonable function \( f \), there is a language that is not in \( \text{TIME}(f(n)) \).

The proof is based on the diagonal method, as in the proof of the undecidability of the halting problem.

**Constructible Functions**

A complexity class such as \( \text{TIME}(f(n)) \) can be very unnatural, if \( f(n) \) is.

From now on, we restrict our bounding functions \( f(n) \) to be proper functions:

**Definition**

A function \( f : \mathbb{N} \to \mathbb{N} \) is constructible if:

- \( f \) is non-decreasing, i.e. \( f(n+1) \geq f(n) \) for all \( n \); and
- there is a deterministic machine \( M \) which, on any input of length \( n \), replaces the input with the string \( 0^{f(n)} \), and \( M \) runs in time \( O(n + f(n)) \) and uses \( O(f(n)) \) work space.

**Examples**

All of the following functions are constructible:

- \( \lfloor \log n \rfloor \);
- \( n^2 \);
- \( n \);
- \( 2^n \).

If \( f \) and \( g \) are constructible functions, then so are \( f + g, f \cdot g, 2^f \) and \( f(g) \) (this last, provided that \( f(n) > n \)).

**Using Constructible Functions**

Recall \( \text{NTIME}(f(n)) \) is defined as the class of those languages \( L \) accepted by a nondeterministic Turing machine \( M \), such that for every \( x \in L \), there is an accepting computation of \( M \) on \( x \) of length at most \( O(f(n)) \).

If \( f \) is a constructible function then any language in \( \text{NTIME}(f(n)) \) is accepted by a machine for which all computations are of length at most \( O(f(n)) \).

Also, given a Turing machine \( M \) and a constructible function \( f \), we can define a machine that simulates \( M \) for \( f(n) \) steps.
**Time Hierarchy Theorem**

For any constructible function $f$, with $f(n) \geq n$, define the $f$-bounded *halting language* to be:

$$H_f = \{ [M], x \mid M \text{ accepts } x \text{ in } f(|x|) \text{ steps} \}$$

where $[M]$ is a description of $M$ in some fixed encoding scheme.

Then, we can show

$H_f \in \text{TIME}(f(n)^3)$ and $H_f \not\in \text{TIME}(f(|n|/2))$

**Time Hierarchy Theorem**

For any constructible function $f(n) \geq n$, $\text{TIME}(f(n))$ is properly contained in $\text{TIME}(f(2n+1)^3)$.

**Strong Hierarchy Theorems**

For any constructible function $f(n) \geq n$, $\text{TIME}(f(n))$ is properly contained in $\text{TIME}(f(n)(\log f(n)))$.

**Space Hierarchy Theorem**

For any pair of constructible functions $f$ and $g$, with $f = O(g)$ and $g \neq O(f)$, there is a language in $\text{SPACE}(g(n))$ that is not in $\text{SPACE}(f(n))$.

Similar results can be established for nondeterministic time and space classes.

**Consequences**

- For each $k$, $\text{TIME}(n^k) \neq \text{TIME}(n^{k+1})$.
- $P \neq \text{EXP}$.
- $L \neq \text{PSPACE}$.
- Any language that is $\text{EXP}$-complete is not in $P$.
- There are no problems in $P$ that are complete under linear time reductions.

**P-complete Problems**

It makes little sense to talk of complete problems for the class $P$ with respect to polynomial time reducibility $\leq_P$.

There are problems that are complete for $P$ with respect to logarithmic space reductions $\leq_L$.

One example is $\text{CVP}$—the circuit value problem.

- If $\text{CVP} \in L$ then $L = P$.
- If $\text{CVP} \in NL$ then $\text{NL} = P$. 