Proof of Fact

Proof of Fact $\Rightarrow$ Let $L$ be decidable and let $M$ be a Turing machine that decides $L$. By swapping $q_{\text{accept}}$ and $q_{\text{reject}}$ of $M$ we get a Turing machine $M'$ that decides $L$. So both $L$ and $\overline{L}$ are Turing-decidable, and thus, Turing-recognizable.

Proof of Fact (cont'd)

$\Leftarrow$ Let $L$ and $\overline{L}$ be recognized by TMs $M_1$ and $M_2$, respectively. Define a two-tape machine $M$ that, on input $x$, does the following:

1. $M$ copies $x$ onto Tape 2.
2. $M$ repeats the following until either $M_1$ or $M_2$ accepts:
   (a) $M$ simulates one step of $M_1$ on Tape 1 then one step of $M_2$ on Tape 2.
3. $M$ accepts $x$ if $M_1$ has accepted and rejects $M_2$ has accepted.

Then $M$ decides $L$ because for every $x$, at least one of the two machines halts on input $x$.

Fact

The Halting Problem

Define $A_{\text{Turing}} = \{(M, w) \mid M \text{ is a Turing machine and accepts } w\}$.

Theorem. $A_{\text{Turing}}$ is not decidable.

Then we have:

Corollary. $A_{\text{Turing}}$ is not Turing-recognizable, and thus, not decidable.

For this corollary we need the following fact.

Fact. A language $L$ is decidable if and only if both $L$ and $\overline{L}$ are Turing-recognizable.

Proof of Corollary. $A_{\text{Turing}}$ is Turing-recognizable and is not decidable. So, $A_{\text{Turing}}$ is Turing-recognizable.

Corollary
**Diagonalization**

A set is **countable** if either it is finite or it has the same size as $\mathcal{N}$; i.e., there is a one-to-one, onto correspondence between $\mathcal{N}$ (or there is a bijection from the set to $\mathcal{N}$).

**Fact.** Let $\mathcal{Q}$ be the set of all positive rational numbers and $\mathcal{R}$ the set of all positive real numbers. Then $\mathcal{Q}$ is countable while $\mathcal{R}$ is not.

**Proof** For the former, each member of $\mathcal{Q}$ is expressed as a fraction $\frac{m}{n}$ such that $m, n \in \mathcal{N}$ and $\gcd(m, n) = 1$.

So we have only to come up with a bijection from $\mathcal{N}$ to the set $\{\frac{m}{n} \mid m, n \geq 1 \& \gcd(m, n) = 1\}$. 

**An Immediate Application of Diagonalization**

**Corollary.** There is a language that is not Turing-recognizable.

**Proof** The set of Turing machines is countable:

1. Fix an encoding scheme of Turing machines on an alphabet $\Sigma$.
2. Go through all the strings in $\Sigma$, e.g., in lexicographic order, and assign numbers to all legal encodings by counting how many legal encodings have been seen so far.

A language over $\Sigma$ can be viewed as an infinite binary number $0, b_1 b_2 b_3 \ldots$, called the characteristic sequence, where for each $i \geq 1$, $b_i$ corresponds to the membership of the $i$th string in the language. So the languages have the same cardinality as the set of binary reals between 0 and 1, which is uncountable.

---

**$\mathcal{Q}$ is countable**

For $p = 1, 2, \ldots$, visit the integral points on the line $m + n = p$ in the first quadrant of the $xy$-plane and count how many pairs $(m, n)$ such that $\gcd(m, n) = 1$ have been seen.
**Proof of Theorem ($A_{TM}$ is not decidable)**

Assume that $A_{TM}$ is decidable. Let $T$ be a Turing machine that decides $A_{TM}$. Define $D$ to be a machine that, on input $w$,

1. Check whether $w$ is a legal encoding of some Turing machine, say $M$. If not, immediately reject $w$.
2. Simulate $T$ on $\langle M, \langle M \rangle \rangle$.
3. If $T$ accepts, then reject; otherwise, accept.

Since $T$ decides $A_{TM}$ by assumption, $M$ always halts; so does $D$. For every Turing machine $M$,

\[ D \text{ accepts } \langle M \rangle \Leftrightarrow M \text{ does not accept } \langle M \rangle \]

With $M = D$, we have

\[ D \text{ accepts } \langle D \rangle \Leftrightarrow D \text{ does not accept } \langle D \rangle. \]

This is a contradiction. $\blacksquare$