Lecture #16: Proofs of Undecidability — Examples I
A Useful Subroutine

Recall that the lecture notes included a proof that the language All_{TM}, consisting of encodings of Turing machines that accepted every input string, was an ***undecidable*** language.

This was established by showing that $\text{A}_{\text{TM}} \preceq_{\text{M}} \text{All}_{\text{TM}}$. In particular, the ***many-one reduction*** from A_{TM} to All_{TM}, used, was a total function $f : \Sigma_{\text{TM}}^{\star} \to \Sigma_{\text{TM}}^{\star}$ satisfying the following properties.

- If $\mu \in \Sigma_{\text{TM}}^{\star}$ and $\mu \notin$ TM+I, so that μ does not encode a Turing machine and an input for that Turing machine, then $f(\mu) = \lambda$.
- If $\mu \in$ TM+I, so that μ encodes a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ (whose start state, q_0, is not in $\{q_{\text{accept}}, q_{\text{reject}}\}$) and an input $\omega \in \Sigma^{\star}$ for M, then $f(\mu)$ is ***an encoding of another Turing machine***, $\mathcal{M}_{\langle M, \omega \rangle}$, that implements the algorithm shown in Figure 1 on page 2.

This can certainly be confusing! This document is intended to try to make it less confusing than it otherwise would be. As later examples may suggest, the Turing machine $\mathcal{M}_{\langle M, \omega \rangle}$ is a generally useful component, so that it is helpful to understand that its encoding really can be computed, when it is needed.

More about the Turing Machine $\mathcal{M}_{\langle M, \omega \rangle}$

The Turing machine $\mathcal{M}_{\langle M, \omega \rangle}$ will have the same input alphabet, Σ, and the same tape alphabet, Γ, as the Turing machine M that is encoded by the string $\mu \in \Sigma_{\text{TM}}^{\star}$. As described in the lecture notes the machine will have two components:

- The first component replaces the input (some string $\nu \in \Sigma^{\star}$) with the string ω — by replacing the input on the machine's tape with a copy of ω.
- The second component is a copy of the encoded Turing machine M, with states renamed (because this is now a component of a larger Turing machine).

```
On input $\nu \in \Sigma^\star$ {
1.  Replace $\nu$ with $\omega$ on the tape, and enter $M$'s start state (so that $M$ is in its initial
    configuration for input $\omega$).
2.  Run $M$ (now, with input $\omega$) — accepting if $M$ eventually accepts $\omega$, rejecting if
    $M$ eventually rejects $\omega$, and looping otherwise.
}
```

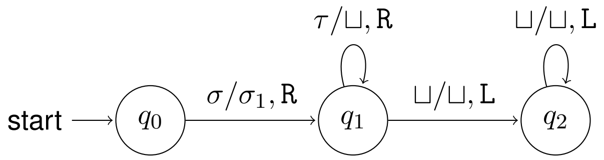
Figure 1: Algorithm Implemented by $\mathcal{M}_{\langle M,\,\omega\rangle}$

Implementation-Level Details: The First Component When ω is the Empty String

Suppose, first, that $\omega = \lambda$. Then it is sufficient to sweep to the right until a copy of "$\sqcup$" is found — erasing most non-blank symbols with copies of "$\sqcup$" in the process — and then sweep back to the left, leaving copies of "$\sqcup$" unchanged, until the leftmost cell on the tape on the tape is reached. The symbol at this cell can then be replaced with "$\sqcup$" as well, with the tape head moving ***left*** once again (leaving the tape head at the leftmost cell).

In order to make it possible to *find* the leftmost cell of the tape, during the sweep back to the right, let us overwrite the *first* symbol seen with the non-blank symbol, σ_1, instead of "$\sqcup$". The leftmost cell can then be detected during the sweep back to the right, because it will be the first (and only) cell where a non-blank symbol is found.

A Turing machine that carries out this process is as follows.


$\tau/\sqcup, \mathtt{R}$
$\sqcup/\sqcup, \mathtt{L}$
start
q_0
$\sigma/\sigma_1, \mathtt{R}$
q_1
$\sqcup/\sqcup, \mathtt{L}$
q_2

Suppose now that $|\Gamma| = m + 1$, so that

$$\Gamma = \{\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_m\}.$$

- In the above picture, "σ" represents every symbol in Γ — so that the above picture shows a transition
$$\delta(q_0, \sigma_i) = (q_1, \sigma_1, \mathtt{R})$$
for every integer i such that $0 \le i \le m$. These are the *first* transitions whose encodings should be included in the encoding of the transition function for $\mathcal{M}_{\langle M,\,\omega\rangle}$.
- In the above picture, "τ" represents every *non-blank* symbol in Γ — so that the above picture shows a transition
$$\delta(q_1, \sqcup) = (q_2, \sqcup, \mathtt{L})$$

as well as a transition

$$\delta(q_1, \sigma_i) = (q_1, \sqcup, \mathtt{R})$$

for every integer i such that $1 \leq i \leq m$. These are the *next* transitions whose encodings should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega \rangle}$.

- While the picture does not show it, there should be a transition from q_2 to the first state in the second component — which will be called "q_3" in this case — for every non-blank symbol $\tau \in \Gamma$.[1] In particular, the next sequence of transitions whose encodings are to be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega \rangle}$ are the transition

$$\delta(q_2, \sqcup) = (q_2, \sqcup, \mathtt{L})$$

and the transitions

$$\delta(q_2, \sigma_i) = (q_3, \sqcup, \mathtt{L})$$

for every integer i such that $1 \leq i \leq m$.

Implementation-Level Details: The First Component When $|\omega| = 1$

Suppose, next, that $|\omega| = 1$, so that $\omega = \alpha \in \Sigma$ — so that $\alpha = \sigma_j$ for some integer j such that $1 \leq j \leq |\Sigma| = h$. In this case, to begin a sweep right, while marking the leftmost cell of the tape, suppose we replace the symbol at the leftmost cell with α, moving right — and then replace the symbol now visible, writing "$\sqcup$" and moving right again. Suppose that the machine continues right, replacing non-blank symbols with "$\sqcup$" until "$\sqcup$" is seen on the tape. The machine can now sweep left until a *non-blank* symbol is seen; this must be the copy of α on the leftmost cell of the tape. It suffices to move left (so that the tape head remains at the leftmost cell), leaving the copy of α unchanged, and proceed to the next stage of the computation.

A Turing machine that carries out this process is as follows.

$$\begin{tikzcd}
\text{start} \arrow[r] & q_0 \arrow[r, "\sigma/\alpha, \mathtt{R}"] & q_1 \arrow[r, "\sigma/\sqcup, \mathtt{R}"] & q_2 \arrow[loop above, "\tau/\sqcup, \mathtt{R}"] \arrow[r, "\sqcup/\sqcup, \mathtt{L}"] & q_3 \arrow[loop above, "\sqcup/\sqcup, \mathtt{L}"]
\end{tikzcd}$$

- In the above picture, "σ" represents every symbol in Γ — so that the above picture shows a transition

$$\delta(q_0, \sigma_i) = (q_1, \alpha, \mathtt{R})$$

[1] In fact, the only one of these transitions that will be used is the transition for $\tau = \sigma_1$. The remaining transitions are simply being chosen to make the transition function as predictable — and easy to generate — as possible.

for every integer i such that $0 \leq i \leq m$. These are the *first* transitions whose encodings should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega \rangle}$.

- The above picture also shows a transition

  $$\delta(q_1, \sigma_i) = (q_2, \sqcup, \mathtt{R})$$

  for every integer i such that $0 \leq i \leq m$. This is the *second* sequence of transitions whose encodings should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega \rangle}$.

- In the above picture, "τ" represents every *non-blank* symbol in Γ — so that the above picture shows a transition

  $$\delta(q_2, \sqcup) = (q_3, \sqcup, \mathtt{L})$$

  as well as a transition

  $$\delta(q_2, \sigma_i) = (q_1, \sqcup, \mathtt{R})$$

  for every integer i such that $1 \leq i \leq m$. These are the *next* transitions whose encodings should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega \rangle}$.

- While the picture does not show it, there should be a transition from q_3 to the first state in the second component — which will be called "q_4" in this case — for every non-blank symbol $\tau \in \Gamma$.[2] In particular, the next sequence of transitions whose encodings are to be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega \rangle}$ are the transition

  $$\delta(q_3, \sqcup) = (q_3, \sqcup, \mathtt{L})$$

  and the transitions

  $$\delta(q_3, \sigma_i) = (q_4, \alpha, \mathtt{L})$$

  for every integer i such that $1 \leq i \leq m$.

Implementation-Level Details: The First Component When $|\omega| \geq 2$

Finally, suppose that $|\omega| \geq 2$, so that

$$\omega = \alpha_1 \alpha_2 \ldots, \alpha_n$$

for an integer n such that $n \geq 2$, and for $\alpha_1, \alpha_2, \ldots, \alpha_n \in \Sigma$.

Now, one way to mark leftmost cell of the tape, while beginning of the copying of ω onto the tape, is to write "$\sqcup$" onto the leftmost cell moving right – moving right. The Turing machine implementing this part of the algorithm, for this case, is as follows.

[2]The only one of these transitions that will be used is the transition for $\tau = \sigma_j$. The remaining transitions are simply being chosen to make the transition function as predictable — and easy to generate — as possible.

$$\begin{tikzcd}\text{start} \arrow[r] & q_0 \arrow[r, "\sigma/\sqcup, \mathtt{R}"] & q_1\end{tikzcd}$$

- In the above picture, "σ" represents every symbol in Γ — so that the above picture shows a transition
$$\delta(q_0, \sigma_i) = (q_1, \sqcup, \mathtt{R})$$
for every integer i such that $0 \leq i \leq m$. This is the *first* sequence of transitions whose encodings should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega \rangle}$ when $|\omega| \geq 2$.

For each integer j such that $2 \leq j \leq n$, the machine should replace the symbol visible on the tape with α_j, moving right — so that the Turing machine also includes states $q_2, q_3, \dots, q_n$ and transitions with the form

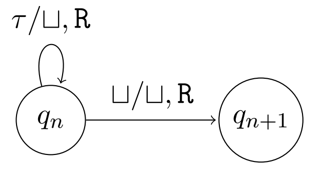
$$\begin{tikzcd}q_{j-1} \arrow[r, "\sigma/\alpha_j, \mathtt{R}"] & q_j\end{tikzcd}$$

for $2 \leq j \leq n$.

- Once again,"σ" represents every symbol in Γ. Thus — so that transitions are listed in dictionary order — one can think of the encoding of the transition function of $\mathcal{M}_{\langle M, \omega \rangle}$ updated as follows:

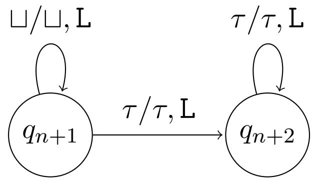
```
integer j := 2
while (j ≤ $n$) {
  integer i := 0
  while (i ≤ $m$) {
    Append an encoding of the transition

        $\delta(q_{\mathtt{j}-1}, \sigma_{\mathtt{i}}) = (q_{\mathtt{j}}, \alpha_{\mathtt{j}}, \mathtt{R})$

    onto the end of the encoding of the transition function
    for $\mathcal{M}_{\langle M, \omega \rangle}$ that is now being constructed.
    i := i + 1
  }
  j := j + 1
}
```

After the final symbol, α_n, in ω has been written, the sweep to the right should continue, with non-blank symbols replaced with copies of "⊔", until a copy of "⊔" is seen.



- In the above picture, "τ" represents every *non-blank* symbol in Γ — so that the above picture shows a transition
$$\delta(q_n, \sqcup) = (q_{n+1}, \sqcup, \mathtt{R})$$
as well as a transition
$$\delta(q_n, \sigma_i) = (q_n, \sqcup, \mathtt{R})$$
for every integer i such that $1 \leq i \leq m$. These are the *next* transitions whose encodings should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega\rangle}$.

Now, at this point, the tape does not, quite, store the desired string ω, with copies of "⊔" to the right — because the leftmost cell stores a copy of "⊔" instead of the first symbol, α_1, in ω. Since the tape head is now resting at a copy of "⊔" somewhere to the right of that, the update of the tape can be completed by sweeping left over zero or more copies of "⊔"; sweeping left over one or more non-blank symbols (that is, the symbols $\alpha_n, \alpha_{n-1}, \ldots, \alpha_3, \alpha_2$), and then replacing the next copy of "⊔" that is seen with a copy of α_1, moving left — and moving to the first state, q_{n+3}, in the next component:



- Once again, in the above picture, "τ" represents every *non-blank* picture — so that the above picture shows a transition
$$\delta(q_{n+1}, \sqcup) = (q_{n+1}, \sqcup, \mathtt{L})$$
as well as a transition
$$\delta(q_{n+1}, \sigma_i) = (q_{n+2}, \sigma_i, \mathtt{L})$$
for every integer i such that $1 \leq i \leq m$. These are the *next* transitions whose encodings should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega\rangle}$.

- While it is not shown in the picture — as noted above — the encoding of a transition

  $$\delta(q_{n+2}, \sqcup) = (q_{n+3}, \alpha_1, \mathtt{L})$$

  should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega\rangle}$ that is being constructed. The encoding of transitions

  $$\delta(q_{n+2}, \sigma_i) = (q_{n+2}, \sigma_i, \mathtt{L})$$

  should be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega\rangle}$, for every integer i such that $1 \leq i \leq m$, after that.

Implementation-Level Details: The Second Component

Note that $n + 2$ states have been included the first component, so that q_{n+3} is the new name for the first state for the second component, when $|\omega| = n$, for ***every*** integer n such that $n \geq 0$. Thus the sequence of transitions to be included in the encoding of the transition function for $\mathcal{M}_{\langle M, \omega\rangle}$ can now be completed by listing all the transitions included in the transition function for M (in order) — renaming states, so that each state q_j is renamed as q_{j+n+3}, for $0 \leq j \leq |Q| - 3$, in the process.

A Suggested Exercise

Exercise: Use the above information, adding details as you need them, in order to confirm that a function $f : \Sigma_{\mathsf{TM}}^{\star} \to \Sigma_{\mathsf{TM}}^{\star}$ such that, for all $\mu \in \Sigma_{\mathsf{TM}}^{\star}$,

- if $\mu \in$ TM+I, so that μ encodes a Turing machine M and input string ω for M, then $f(\mu)$ is the encoding of the corresponding Turing machine $\mathcal{M}_{\langle M, \omega\rangle}$; and
- if $\mu \notin$ TM+I then $f(\mu) = \lambda$

is a ***computable*** function.