Announcements

Reminders:

- Midterm 1 Solution Sessions - Saturday 1:30-2:30 in GHC 4301 and Sunday 1:30-2:30 in GHC 4215
- Remember to submit answers to the weekly quiz by 9 pm Sunday.

These Decidable Definitions Have Undecidable Ends

- A decider is a TM that halts on all inputs.
- A language $L$ is undecidable if there is no TM $M$ that halts on all inputs such that $M(x)$ accepts if and only if $x \in L$.
- A language $A$ reduces to $B$ if it is possible to decide $A$ using an algorithm that decides $B$ as a subroutine. Denote this as $A \leq B$ (read: $B$ can be used to solve $A$ so $A$ is at most as hard as $B$)

 Doesn’t Look Like Anything (Decidable) To Me

Prove that the following languages are undecidable (below, $M$, $M_1$, $M_2$ refer to TMs).

(a) $\text{REGULAR} = \{ \langle M \rangle : L(M) \text{ is regular} \}$.

We show that $\text{REGULAR}$ is undecidable via a reduction from $\text{HALTS}$. Suppose $M_{\text{REG}}$ decides $\text{REGULAR}$. We define a decider for $\text{HALTS}$ as follows

```python
def M_HALTS(<M, x>):
    <HELP> =
    """def HELP(w):
        if w in {0^n1^n | n a natural number}:
            ACCEPT
        else:
            M(x)
            ACCEPT"
    return M_REG(<HELP>)
```

Proof of correctness:

Suppose that $M(x)$ halts, then $L(HELP) = \Sigma^*$, so $M_{\text{REG}}(\langle HELP \rangle)$ accepts, as desired.

Suppose that $M(x)$ loops, then $L(HELP) = \{0^n1^n | n \in \mathbb{N} \}$, so $M_{\text{REG}}(\langle HELP \rangle)$ rejects, as desired.
Thus, we've shown that \textsc{halts} \leq \textsc{regular}, so \textsc{regular} is undecidable.

(b) \textit{TOTAL} = \{\langle M \rangle | M \text{ halts on all inputs} \}.

We show that \textit{TOTAL} is undecidable via a reduction from \textsc{halts}. Suppose \textit{m.total} decides \textit{TOTAL}. We define a decider for \textsc{halts} as follows

\begin{verbatim}
def M_HALTS(<M, x>):  
    <HELP> =  
        """def HELP(w):
            M(x)
            ACCEPT"
    return M_TOTAL(<HELP>)
\end{verbatim}

Proof of correctness:

Suppose that \texttt{M(x)} halts, then \texttt{HELP} halts on all inputs, so \texttt{m.total(\langle HELP \rangle)} accepts, as desired.

Suppose that \texttt{M(x)} loops, then \texttt{HELP} does not halt on all inputs, so \texttt{m.total(\langle HELP \rangle)} rejects, as desired.

Thus, we've shown that \textsc{halts} \leq \textit{TOTAL}, so \textit{TOTAL} is undecidable.

(c) \textit{Dolores} = \{\langle M_1, M_2 \rangle : \exists w \in \Sigma^* \text{ such that both } M_1(w) \text{ and } M_2(w) \text{ accept} \}.

We show that \textit{Dolores} is undecidable via a reduction from \textsc{halts}. Suppose \textit{m.dolores} decides \textit{Dolores}. We define a decider for \textsc{halts} as follows

\begin{verbatim}
def M_HALTS(<M, x>):  
    <HELP> =  
        """def HELP(w):
            M(x)
            ACCEPT"
    return M_DOLORES(<HELP, HELP>)
\end{verbatim}

Proof of correctness:

Suppose that \texttt{M(x)} halts, then \texttt{HELP} accepts all inputs, so \texttt{m.dolores(\langle HELP, HELP \rangle)} accepts, as desired.

Suppose that \texttt{M(x)} loops, then \texttt{HELP} rejects all inputs, so \texttt{m.dolores(\langle HELP, HELP \rangle)} rejects, as desired.

Thus, we've shown that \textsc{halts} \leq \textit{Dolores}, so \textit{Dolores} is undecidable.

(Extra) Lose All Scripted Responses. Improvisation Only

Let \textit{finite} = \{\langle M \rangle : M \text{ is a TM and } L(M) \text{ is finite} \}.

Show that \textit{TOTAL} \leq \textit{finite}.
Suppose that \textit{M\_FINITE} decides \textit{FINITE}.

We define a decider for \textit{TOTAL} as follows, where \textless{} means lexicographically smaller.

\begin{verbatim}
def M_TOTAL(<M>):
    <HELP> =
    """def HELP(w):
        for y <= w:
            M(y)
        ACCEPT"
    return not MFINITE(<HELP>)
\end{verbatim}

Proof of correctness:

Suppose \textit{M} is total and halts on all inputs, then \(L(HELP) = \Sigma^*\), so \textit{M\_FINITE(\langle HELP \rangle)} will reject, as desired.

Suppose \textit{M} is not total and let \(w\) be the lexicographically smallest string that \textit{M} does not halt on. Then \textit{HELP} will not accept any string lexicographically greater (or equal to) than \(w\), as \textit{M}(\(w\)) will not halt. Thus, since there are only finitely many strings lexicographically smaller than \(w\), \(L(HELP)\) is finite. Thus, \textit{M\_FINITE(\langle HELP \rangle)} accepts, as desired.