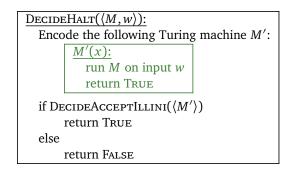
Prove that the following languages are undecidable.

1. ACCEPTILLINI := { $\langle M \rangle$ | *M* accepts the string **ILLINI**}

Solution: For the sake of argument, suppose there is an algorithm DECIDEACCEPTILLINI that correctly decides the language ACCEPTILLINI. Then we can solve the halting problem as follows:



We prove this reduction correct as follows:

 \implies Suppose *M* halts on input *w*.

- Then M' accepts *every* input string x. In particular, M' accepts the string **ILLINI**. So DecideAcceptIllini accepts the encoding $\langle M' \rangle$. So DecideHalt correctly accepts the encoding $\langle M, w \rangle$.
- \leftarrow Suppose *M* does not halt on input *w*.

Then M' diverges on *every* input string x.

- In particular, M' does not accept the string **ILLINI**.
- So DecideAcceptIllini rejects the encoding $\langle M' \rangle$.
- So DecideHalt correctly rejects the encoding $\langle M, w \rangle$.

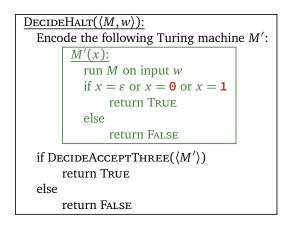
In both cases, DECIDEHALT is correct. But that's impossible, because HALT is undecidable. We conclude that the algorithm DECIDEACCEPTILLINI does not exist.

As usual for undecidablility proofs, this proof invokes *four* distinct Turing machines:

- The hypothetical algorithm DECIDEACCEPTILLINI.
- The new algorithm DECIDEHALT that we construct in the solution.
- The arbitrary machine *M* whose encoding is part of the input to DECIDEHALT.
- The special machine *M*['] whose encoding DECIDEHALT constructs (from the encoding of *M* and *w*) and then passes to DECIDEACCEPTILLINI.

2. AcceptThree := $\{\langle M \rangle \mid M \text{ accepts exactly three strings}\}$

Solution: For the sake of argument, suppose there is an algorithm DECIDEACCEPTTHREE that correctly decides the language ACCEPTTHREE. Then we can solve the halting problem as follows:



We prove this reduction correct as follows:

 \implies Suppose *M* halts on input *w*.

Then M' accepts exactly three strings: ε , 0, and 1. So DecideAcceptThree accepts the encoding $\langle M' \rangle$. So DecideHalt correctly accepts the encoding $\langle M, w \rangle$.

 $\begin{array}{ll} \longleftarrow & \text{Suppose } M \text{ does not halt on input } w. \\ & \text{Then } M' \text{ diverges on } every \text{ input string } x. \\ & \text{In particular, } M' \text{ does not accept exactly three strings (because } 0 \neq 3). \\ & \text{So DecideAcceptThree rejects the encoding } \langle M' \rangle. \\ & \text{So DecideHalt correctly rejects the encoding } \langle M, w \rangle. \end{array}$

In both cases, DECIDEHALT is correct. But that's impossible, because HALT is undecidable. We conclude that the algorithm DECIDEACCEPTTHREE does not exist.

3. ACCEPTPALINDROME := $\{\langle M \rangle \mid M \text{ accepts at least one palindrome}\}$

Solution: For the sake of argument, suppose there is an algorithm DECIDEACCEPTPALINDROME that correctly decides the language ACCEPTPALINDROME. Then we can solve the halting problem as follows:

DECIDEHALT($\langle M, w \rangle$):	
Encode the following Turing machine M' :	
$\underline{M'(x)}$:	
run <i>M</i> on input <i>w</i>	
return True	
if DecideAcceptPalindrome($\langle M' angle$)	
return True	
else	
return False	

We prove this reduction correct as follows:

 \implies Suppose *M* halts on input *w*.

Then M' accepts *every* input string x. In particular, M' accepts the palindrome **RACECAR**. So DECIDEACCEPTPALINDROME accepts the encoding $\langle M' \rangle$. So DECIDEHALT correctly accepts the encoding $\langle M, w \rangle$. \iff Suppose M does not halt on input w. Then M' diverges on *every* input string x. In particular, M' does not accept any palindromes. So DECIDEACCEPTPALINDROME rejects the encoding $\langle M' \rangle$. So DECIDEHALT correctly rejects the encoding $\langle M, w \rangle$.

In both cases, DECIDEHALT is correct. But that's impossible, because HALT is undecidable. We conclude that the algorithm DECIDEACCEPTPALINDROME does not exist.

Yes, this is *exactly* the same proof as for problem 1.

3