

Prove that the following languages are undecidable.

1. $\text{ACCEPTILLINI} := \{ \langle M \rangle \mid M \text{ accepts the string } \mathbf{ILLINI} \}$

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

<pre> DECIDEHALT($\langle M, w \rangle$): Encode the following Turing machine M': <table border="1"> <tr> <td> <pre> $M'(x)$: run M on input w return TRUE </pre> </td> </tr> </table> if $\text{DECIDEACCEPTILLINI}(\langle M' \rangle)$ return TRUE else return FALSE </pre>	<pre> $M'(x)$: run M on input w return TRUE </pre>
<pre> $M'(x)$: run M on input w return TRUE </pre>	

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 \mathbf{ILLINI} .
 So $\text{DECIDEACCEPTILLINI}$ accepts the encoding $\langle M' \rangle$.
 So DECIDEHALT correctly accepts the encoding $\langle M, w \rangle$.
- \impliedby Suppose M does not halt on input w .
 Then M' diverges on *every* input string x .
 In particular, M' does not accept the string \mathbf{ILLINI} .
 So $\text{DECIDEACCEPTILLINI}$ 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 $\text{DECIDEACCEPTILLINI}$ does not exist. ■

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

- The hypothetical algorithm $\text{DECIDEACCEPTILLINI}$.
- 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 $\text{DECIDEACCEPTILLINI}$.

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

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:

```

DECIDEHALT( $\langle M, w \rangle$ ):
  Encode the following Turing machine  $M'$ :
   $M'(x)$ :
    run  $M$  on input  $w$ 
    if  $x = \varepsilon$  or  $x = 0$  or  $x = 1$ 
      return TRUE
    else
      return FALSE
  if DECIDEACCEPTTHREE( $\langle M' \rangle$ )
    return TRUE
  else
    return FALSE

```

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 $\langle M, w \rangle$.

\impliedby Suppose M does not halt on input w .

Then M' diverges on every input string x .

In particular, M' does not accept exactly three strings (because $0 \neq 3$).

So DECIDEACCEPTTHREE 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 DECIDEACCEPTTHREE does not exist. ■

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

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

$\text{DECIDEHALT}(\langle M, w \rangle):$ Encode the following Turing machine M' : <table border="1"> <tr> <td> $M'(x):$ run M on input w return TRUE </td> </tr> </table> if $\text{DECIDEACCEPTPALINDROME}(\langle M' \rangle)$ return TRUE else return FALSE	$M'(x):$ run M on input w return TRUE
$M'(x):$ run M on input w return TRUE	

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 $\text{DECIDEACCEPTPALINDROME}$ accepts the encoding $\langle M' \rangle$.

So DECIDEHALT correctly accepts the encoding $\langle M, w \rangle$.

\impliedby 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 $\text{DECIDEACCEPTPALINDROME}$ 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 $\text{DECIDEACCEPTPALINDROME}$ does not exist.

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