

Prove that the following languages are undecidable.

See outline of how to solve such problems in the original problem set.

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

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:

<p>DecideHalt($\langle M, w \rangle$):</p> <p>Encode the following Turing machine M':</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 5px;"> <p>$M'(x)$:</p> <p>run M on input w</p> <p>return TRUE</p> </td> </tr> </table> <p>if $\text{DECIDEACCEPTILLINI}(\langle M' \rangle)$</p> <p>return TRUE</p> <p>else</p> <p>return FALSE</p>	<p>$M'(x)$:</p> <p>run M on input w</p> <p>return TRUE</p>
<p>$M'(x)$:</p> <p>run M on input w</p> <p>return TRUE</p>	

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 $\langle M, w \rangle$.
- \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 *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 undecidability 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 $\text{ACCEPTTHREE} := \{\langle M \rangle \mid M \text{ accepts exactly three strings}\}$

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:

\Rightarrow 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 $\langle M, w \rangle$.

\Leftarrow 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 **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  $M'$ :
   $M'(x)$ :
    run  $M$  on input  $w$ 
    return TRUE
  if DECIDEACCEPTPALINDROME( $\langle M' \rangle$ )
    return TRUE
  else
    return FALSE
```

We prove this reduction correct as follows:

- \implies Suppose M halts on input w .
Then M' accepts *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 $\langle M, w \rangle$.
- \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 **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.