

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  $\text{ACCEPTILLINI}$ . Then we can solve the halting problem as follows:

<pre> DECIDEHALT(<math>\langle M, w \rangle</math>):   Encode the following Turing machine <math>M'</math>:   <table border="1" style="margin: 5px auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;"><math>M'(x)</math>:</td> </tr> <tr> <td style="padding: 2px 5px;">run <math>M</math> on input <math>w</math></td> </tr> <tr> <td style="padding: 2px 5px;">return TRUE</td> </tr> </table>           if <math>\text{DECIDEACCEPTILLINI}(\langle M' \rangle)</math>             return TRUE           else             return FALSE         </pre>	$M'(x)$ :	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 string **ILLINI**.

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

So  $\text{DECIDEHALT}$  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  $\text{DECIDEACCEPTILLINI}$  rejects the encoding  $\langle M' \rangle$ .

So  $\text{DECIDEHALT}$  correctly rejects the encoding  $\langle M, w \rangle$ .

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

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

- The hypothetical algorithm  $\text{DECIDEACCEPTILLINI}$ .
- The new algorithm  $\text{DECIDEHALT}$  that we construct in the solution.
- The arbitrary machine  $M$  whose encoding is part of the input to  $\text{DECIDEHALT}$ .
- The special machine  $M'$  whose encoding  $\text{DECIDEHALT}$  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  $\text{DECIDEACCEPTTHREE}$  that correctly decides the language  $\text{ACCEPTTHREE}$ . 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  $x = \epsilon$  or  $x = 0$  or  $x = 1$ 
      return TRUE
    else
      return FALSE
  if  $\text{DECIDEACCEPTTHREE}(\langle M' \rangle)$ 
    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:  $\epsilon$ ,  $0$ , and  $1$ .

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

So  $\text{DECIDEHALT}$  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  $\text{DECIDEACCEPTTHREE}$  rejects the encoding  $\langle M' \rangle$ .

So  $\text{DECIDEHALT}$  correctly rejects the encoding  $\langle M, w \rangle$ .

In both cases,  $\text{DECIDEHALT}$  is correct. But that's impossible, because  $\text{HALT}$  is undecidable. We conclude that the algorithm  $\text{DECIDEACCEPTTHREE}$  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  $\text{ACCEPTPALINDROME}$ . 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  $\text{DECIDEACCEPTPALINDROME}$  accepts the encoding  $\langle M' \rangle$ .

So  $\text{DECIDEHALT}$  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  $\text{DECIDEHALT}$  correctly rejects the encoding  $\langle M, w \rangle$ .

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

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