# 1.

# 2. 3.15d, p.189

Let *M1* be a Turing machine that decides a language *L* and construct a Turing machine *M2* that decides :

*M2* = “On input *w*:

1. Simulate *M1* on *w*.
2. Accept if *M1* rejects, reject if *M1* accepts.”

Since *M1* decides *L* we know it halts on all inputs, therefore *M2* will also halt on all inputs. Additionally, *M2* will always produce the correct result because if , *M1* will accept and *M2* will reject, also if

, *M1* will reject, in which case *M2* will accept.

# 3. 4.2, p.211

where is a DFA equivalent to the regular expression

Let *M1* be a TM that decides *L*:

“On input

1. Convert R into an equivalent DFA
2. Run TM , from Theorem 4.5 which decides , on input
3. Accept if accepts, and reject if rejects.”

Since was proven to be a decidable language, is therefore also decidable.

# 4. 4.3, p.211

Let be a TM that decides :

“On input

1. Construct a DFA that recognizes
2. Run TM , from Theorem 4.4 which decides , on input
3. Accept if accepts, and reject if rejects.”

Since was proven to be a decidable language, is also decidable.

# 5. 4.7, p.211

The proof is by contradiction, that is suppose that is countable. Each element in is an infinite sequence where . We can define a correspondence between and . Let where and is the bit in the sequence. For example:

|  |  |
| --- | --- |
|  |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
|  |  |

Define a sequence , in which the bit in is opposite the bit in the sequence. So for the example above . Thus, differs from each sequence by at least one bit and for any , which is a contradiction and is uncountable.

# 6. 4.8, p.211