1.

Loop invariant: .

P(i): if there is an ith iteration of the loop, then and .

Prove by simple induction, ∀i ∈ N: P(i).

Base case, i = 0:

Already know that ∈ N. Before entering the loop,is guaranteed by line1 code. Then .

Then P (0) holds.

Inductive step:

Assume i is an arbitrary natural number and P (i) is true, that is and . We want to prove P (i +1), that is and .

If there is an i+1th loop iteration, then .

Then = #by code

Then , since by hypothesis.

Also #by code

= #by hypothesis,

=

=

=

=

= #

Then , since

Then P(i+1).

Therefore, ∀i ∈ N: P(i).

Now we want to prove termination:

If there is an (i + 1) th iteration of the loop,

#by code

#since

Then <> is a strictly decreasing sequence.

Also, since we already know that ∈ N and,

Then ∈ N.

Then we find that <> is a strictly decreasing sequence of natural numbers, by the Principle of Well-Ordering, <> must be finite.

Then there is a finite number of loop iterations.

Therefore, the loop terminates.

2.

a)

Here is my specification for that accepts:

,

,

 =

A,

F = {},

Define as the smallest set such that:

1)

2)

Prove that accepts:

Define P(s) as:

I will prove , P(s) by structural induction.

Basis case:

is an empty string, then is equivalent to repeated 0 times, .

Then , so the implication in the first line of the invariant is true in this case.

Also, since is of the form, the implication in the second line of invariant is vacuously true.

Then P ().

Inductive Step:

Let and assume P(s), I will show that P () and P () follow. There are two cases to consider:

Case :

Then # by P(s)

= #add one more a

Case :

Then

#by P(s)

=

# add one more b, cannot be in the form

# since is not in the form of then is vacuously true.

Then P () and P () follows.

The first line of the invariant ensures that all strings that can be expressed as are accepted.

The contrapositive of the second line of the invariant ensures that any strings that does not derive the machine to can be expressed in the form of . In other words, all strings that derive the machine to state can be expressed in the form .

Therefore, accepts .

b)

Here is my specification for that accepts:

,

,

 =

F =

Define as the smallest set such that:

1)

2)

Prove that accepts:

Define P(s) as:

I will prove , P(s) by structural induction.

Basis case:

is an empty string, then is equivalent to repeated 0 times, .

Then , so the implication in the first line of the invariant is true in this case.

Also, since is of the form, the implication in the second line of invariant is vacuously true.

Then P ().

Inductive Step:

Let and assume P(s), I will show that P () and P () follow.

There are two cases to consider:

Case :

Then # by P(s)

=

#add one more a, cannot be in the form of .

# since is not in the form of then is vacuously true.

Case :

Then

#by P(s)

= #add one more b

Then P () and P () follows.

The first line of the invariant ensures that all strings that can be expressed as are accepted.

The contrapositive of the second line of the invariant ensures that any strings that does not derive the machine to can be expressed in the form of . In other words, all strings that derive the machine to state can be expressed as the form

Therefore, accepts .

c)

Here is my specification for that accepts:

,

,

 =

F =

Define as the smallest set such that:

1)

2)

Prove that accepts:

Define P(s) as:

I will prove : P(s) by structural induction.

Basis case:

is an empty string, then || = 0, 0 is an even number.

Then , so the implication in the first line of the invariant is true in this case.

Also, since the length of , the implication in the second line of invariant is vacuously true.

Then P ().

Inductive Step:

Let and assume P(s), I will show that P () and P () follow.

Let c ,

Then # by P(s)

= #add one more c, the length increases by 1

Then P () follows.

The first line of the invariant ensures the length of x is even are accepted.

The contrapositive of the second line of the invariant ensures that any strings that does not derive the machine to O is not of the odd length, in other words, all strings that derive the machine to state E is of the even length.

Therefore, accepts .

d)

Here is my specification for that accepts:

,

,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| a |  |  |  |  |
| b |  |  |  |  |

F = {}

Now we want to prove accepts , denote the states for as , the states for as , their respective transition functions and , and their transition function for as. Inspection of shows that if (, , c) ,  then ( (, ) , c) = ( (, c ), (, c )).

Therefore, the following invariant follows by simply taking conjunctions of the invariants of the component machines, for any .

P(s) =

The implication on the first line ensures that all strings that can be expressed in the form of and end up in state The contrapositive of the implications on the other lines ensure that any strings that does not drive the machines to one of those 3 states can be expressed in the form of and . Hence accepts

e)

Here is my specification for that accepts:

,

,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| **a** |  | **, E)** | **, O)** | **E)** | **, O)** | **, E)** | **O)** | **E)** |
| **b** |  | **E)** | **, O)** | **E)** | **O)** | **E)** | **O)** | **E)** |

,

F = {}

Now we want to prove  accepts , denotes the states for as , the states for as , their respective transition functions and , and their transition function for as. Inspection of shows that if (, , c) ,  then ( (, ) , c) = ( (, c ), (, c )).

Therefore, the following invariant follows by simply taking conjunctions of the invariants of the component machines, for any .

P(s) =

The implication on the first line ensures that all strings that can be expressed in the form of and with even length end up in state **.** The contrapositive of the implications on the other lines ensure that any strings that does not drive the machines to one of those 7 states can be expressed in the form of and with even length. Hence accepts .

5.

a)

Suppose there exists DFA that accepts has 8 states.

According to Pigeonhole Principle, if we choose 9 strings over , then at least 2 strings must be end at the same state q.

We pick the following 9 strings: = , a, = ab, = aa, = cc, = aaa, = cbb, = bbb, = abba.

By pigeonhole Principle, for one of the pairs of the string, the supposed 8-state DFA is forced into the same state for both strings.

Let x = abba, then = abba, accepted; but all of to are rejected, since their length are not 4. Contradiction.

Let x = aaa, then = aaaa , accepted; but all of , , ,, , , are rejected, since their length are not 4. Contradiction.

Let x = ba, then = abba, accepted; then = aaba, rejected; then = ccba, rejected; then , , , are rejected since their length are not 4. Contradiction.

Let x = aa, then = aaaa, accepted; then = abaa, rejected; then = ccaa, rejected; then , ,, are rejected since their length are not 4. Contradiction.

Let x = cc, then = cccc, accepted; then = abcc, rejected; then = aacc, rejected; then , , , are rejected since their length are not 4. Contradiction.

Let x = a, then = aaaa, accepted; then = cbba, rejected; then = bbba, rejected; then , , ,, are rejected since their length are not 4. Contradiction.

Let x = c, then = cbbc, accepted; then = aaac, rejected; then = bbbc, rejected; then , , ,, are rejected since their length are not 4. Contradiction.

Let x = b, then = bbbb, accepted; then = aaab, rejected; then = cbbb, rejected; then , , ,, are rejected since their length are not 4. Contradiction.

Let x = , then = abba, accepted; but , , ,, , , are rejected since their length are not 4. Contradiction.

Then no pair of strings above end at the same state, which is contradiction with pigeonhole principle.

Then there does not exist DFA that accepts has 8 states.

Therefore, Any DFA that accepts has at least 9 states

b)

|x| is the length of x.

If |x| is even, then a DFA that accepts has at least states.

If |x| is odd, then a DFA that accepts has at least states.