Assignment 7

CSCE 4323: Formal Languages and Computability

Fall 2018

Solutions to the remaining problems should be typed and contained in a PDF, but may include ﬁgures neatly drawn by hand and included in the PDF. Proper notation and terminology must be used.

1. Let ALLDFA = {〈A〉| A is a DFA and L(A) = Σ∗}. Show that ALLDFA is decidable.

Let *M* = on input 〈A〉 where *A* is a DFA:

1. Let *C* be the DFA obtained by exchanging accept and reject states of *A*
2. Run TM *T* on input 〈C〉 to see where *L*(C) = 0
3. In this case accept, otherwise reject

2. Let B be the set of all inﬁnite sequences over {0,1}. Show that *B* is uncountable, using a proof by diagonalization.

Let *B* be an infinite sequence {*b1,b2,b3, …*}such that *bi* ∈ ∑\* and *B* is countable

Let *f* be the correspondence between *N* = {1,2,3,…} such that *n* ∈ *N*, and *B*

Let *f*(n) = {*bn,1,bn,2,bn,3, …, bn,i*}

Let *c =* {*c1, c2, c3, …*} ∈ *B* , where *ci =* 1 – *bi,I* for each *i* ∈ *N* and the *ith* bit in *c* is the opposite of the *ith* bit in the *ith* sequence

With this in mind, each *c* differs from the *nth* sequence in the *nth* bit. So *cn ≠ f(n)* for any *n*

This is a contradiction and therefore *B* is uncountable

3. Let *T* = {(*i,j,k*) | *i,j,k* ∈N}. Show that *T* is countable.

Assume *T* is countable and declare for each triple (*i,j,k*), call *i+j+k* in order to calculate the sum of the triple.

Define a variable *n* ∈ *N*, where *s* is a number belonging to the natural numbers and is used to calculate the sum of (*i,j,*k). The variable *n* is a finite set of the sums.

Now compute the sum of the triples equal to 0, then equal to 1, then equal to 2, and so on for all triples in *T*

In order to prove *T* is countable, make a one-to-one and onto function such that:

*f* : *T* → *N*

We know *P =* {(*i,j*) | *i,j ∈ N*} is countable

Now imagine that *f*{(*I,j,k*)} = *f*{(*i’,j’,k’*)} and declare another one-to-one function *h* such that:

*h*{(*h*(*i,j*),*k*)} = *h*{(*h*(*i’,j’*),*k*’)}

Since *h*(*i,j*) = *h*(*i’,j’*) and *k* = *k*’ then (*i,j*) = (*i’,j’*) since *h* is also one-to-one

Therefore (*i,j,k*)= (*i’,j’,k’*) means that *f* is one-to-one as well

Since *n* ∈ *N* then *h*{(*m,k*)} = *n* for some *m,k* ∈ N because *h* is onto

*h*{(*i,j*)} = *m* for some *i,j* ∈ N because *h* is onto

Therefore, *f*{(*i,j,k*)} = *h*{(*m,k*)} = *h*{(*h*{(*i,j*)},*k*} which makes *f* onto, which also makes *f* a bijective function

Therefore, *T* is countable

4. Let *S* = {〈M〉 | *M* is a DFA that accepts *wR* whenever it accepts *w*}. Show that *S* is decidable.

Assume *S* is undecidable: on input *D*, where *M* is a TM and *w* is a string:

­1) If 〈M,w〉 is a valid encoding for *M* and *w* then accept, else reject

2) Construct a TM ­*M2­­* from *M* and *w*:

a. *M2* on input *x*:

i. If x ∈ L(00\*11\*) then accept

ii. If x ∉ L(00\*11\*) then run *M* on input *w*. Accept when *M* accepts, else reject

3) Run *S* on input 〈M2­〉 and accept is *S* accepts, otherwise reject

5. Let *T* = {〈M〉 | *M* is a TM that accepts *wR* whenever it accepts *w*}. Show that *T* is undecidable.

Consider a TM *A* = {〈M,w〉 | *M* is a TM that accepts *w*}: on input 〈M,w〉 where *M* is a TM and *w* is a string

1. If 〈M,w〉 is a valid encoding for *M* and *w* then accept, else reject
2. Construct a TM *M2­­* from *M* and *w*:
   1. *M2­* on input *x*:
      1. If *x* ∈ L(00\*11\*) then accept
      2. If x ∉ L(00\*11\*) then run *M* on input *w*. Accept if *M* accepts *w*, otherwise reject
3. Run *T* on input 〈M2〉 and if *T* accepts then accept, otherwise reject

6. Consider the problem of determining whether a Turing machine *M* on an input *w* ever attempts to move its head left at any point during its computation on *w*. Formulate this problem as a language and show that it is decidable.

Construct a TM *K* that decides *KTM*

*KTM* = {〈M,w〉 | *M* attempts to move it’s head left when its head is in the left position while computing *w*}

Define TM *n* to evaluate *KTM*

Each step to the left takes *w+nm+*1 steps, where *nm* is the total number of steps taken by TM *M* and *w* is the size of the input string

Assume *M* wants to move its head to the left and build the TM *K* as such:

Turing Machine *K*: on input 〈M,w〉:

1. Simulate *M* on input string *w* for *w+nm+*1 steps
2. If *M* moves its head left then accept, otherwise reject

When the user selects *K*, *K* has decider *KTM* which always halts

Therefore, this proves that *KTM* has a decider

7. Consider the problem of determining whether a Turing machine *M* on an input *w* ever attempts to move its head left when its head is on the left-most tape cell. Formulate this problem as a language and show that it is undecidable.

Construct a TM *L* that decides *LTM*

*LTM* = {〈M,w〉 | *M* attempts to move it’s head left when its head is in the left position while computing *w*}

Define TM *n* to evaluate *LTM*

Each step to the left takes *w+nm+*1 steps, where *n­m* is the total number of steps taken by TM *M* and *w* is the size of the input string. *L* can only move its head to the left

Define *q­o* and *qs* to be the start and accept state for the TM and evaluate the shortest path possible as *Px*, with *x* = {*qw,q0,q1,q2, …, qs*}. The length of *Px* is undetermined since it could contain a cycle. The length of *Px* is *w+nm+*1

Construct a TM for *n*, *n­TM*, that simulates *LTM*. Simulate it on string *w* for *w+nm+*1 steps

If *n* moves to the left then accept, otherwise reject

The length of the path of *Px* is undetermined so it stops at *HALTTM*

Therefore, *LTM* is undecidable