3 Prove that the problem associated with language A_{TM} defined below is undecidable. You are given that HALT = {<*M*, *w*>: *M* is a Turing machine and *M* halts on *w*} is undecidable. Use the template provided to perform a mapping reduction. You must give your answer on this exam question sheet.

The language L_3 is defined as
A_{TM} = {<*M, w*>: *M* is a TM that accepts *w*}.

Note, in the template below, some blanks have a small subscript number. Blanks with the same subscript number must have the same value.

<u>Proof</u>

Optional: $\overline{L3}$ = ...

We will use a mapping reduction to prove the reduction

. . . HALT. . . . . ≤ . . . A_{TM}. . . . .

Assume that . . . . A_{TM} . . . .$_{\underline{1}}$ is decidable.

The transition function *f* that maps instances of . . . HALT . . . . to instances of

. A_{TM}. . . . . is given by TM F given by the following pseudocode.

F = "On input <. . . M, w. . . . .$_{\underline{2}}$>:

1. Construct the following N given by the following pseudocode.

   N = "On input u:. . . . .

   . .1. Run M on w. . . . .

   . .2. Accept . . . .

   . . . .

   . . . .”

2. Output <. . . N, u . . . .$_{\underline{3}}$>."

Now, <. . . .N, u . . . .$_{\underline{3}}$> is an element of . . . A_{TM} . . . . . iff <. . M,w. . .$_{\underline{2}}$> is an element of . .HALT. . ..

So, using *f* and the assumption that . . A_{TM}. .$_{\underline{1}}$ is decidable, we can decide . . HALT. ..

A contradiction.

Therefore, . .A_{TM}. .$_{\underline{1}}$ is undecidable. (This also means that the complement of . . A_{TM}. . .$_{\underline{1}}$ is undecidable; the complement of any undecidable language is itself undecidable.)

3 Prove that the problem associated with language L_3 defined below is undecidable. You are given that HALT = {<*M*, *w*>: *M* is a Turing machine and *M* halts on *w*} is undecidable. Use the template provided to perform a mapping reduction. You must give your answer on this exam question sheet.

The language L_3 is defined as
L_3 = {<*M*>: *M* is a Turing machine and $|\mathcal{L}(M)| \geq 5$, i.e. *M* accepts at least five words}.

Note, in the template below, some blanks have a small subscript number. Blanks with the same subscript number must have the same value.

Note, $\mathcal{L}(M)$ is a well-known notation to describe the set of words recognised by machine M.

Proof

Optional: $\overline{\text{L3}}$ = ...

We will use a mapping reduction to prove the reduction

. . HALT . . . ≤ . . . L3 . . . .

Assume that . . . . L3. . . .$_1$ is decidable.

The transition function *f* that maps instances of . . . HALT . . . . to instances of . L3 . . is given by TM F given by the following pseudocode.

F = "On input <. . . .*M*, *w*. . . .$_2$>:

1. Construct the following *N* given by the following pseudocode.

   N = ". On input *u*: . . . .

   . . . . if *u* is in the set {0, 00, 10, 000}:

   . . . . Accept. . . . .

   . . . . Run *M* on *w*. . . . .

   . . . . if *u* == 111:. . . . .

   . . . . Accept . . . ."

2. Output <. . . .*N* . . . .$_3$>."

Now, <. . . . *N*. . . . .$_3$> is an element of . . . . L3 . . . . . . iff <. . *M*, *w* . . . . .$_2$> is an element of . . . HALT. . ..

So, using *f* and the assumption that . . . L3. . . . .$_1$ is decidable, we can decide . HALT. . . .. A contradiction.

Therefore, . . L3 . . . .$_1$ is undecidable. (This also means that the complement of . . . .L3 . . . .$_1$ is undecidable; the complement of any undecidable language is itself undecidable.)

3 Prove that the problem associated with language L_3 defined below is undecidable. You are given that HALT = {<*M*, *w*>: *M* is a Turing machine and *M* halts on *w*} is undecidable. Use the template provided to perform a mapping reduction. You must give your answer on this exam question sheet.

The language L_3 is defined as
L_3 = {<*M, a, b*>: *M* is a Java program, *a* and *b* are integer variables declared in *M*, and when *M* is run, *a* and *b* have the same value at least once}.

Note, in the template below, some blanks have a small subscript number. Blanks with the same subscript number must have the same value.

<u>Proof</u>

Optional: $\overline{\text{L3}}$ = ...

We will use a mapping reduction to prove the reduction

. . HALT . . . ≤ . . . . L3. . ..

Assume that . . . .L3 . . . .$_1$ is decidable.

The transition function *f* that maps instances of . . .HALT . . . . to instances of . . L3 . . . is given by TM F given by the following pseudocode.

F = "On input <. . . *M*, *w*. . .$_2$>:

1. Construct the following *N* given by the following pseudocode.

   N = "void main(void) {. . . . .

   . . . . int x=5, y=6; . . . .

   . . . . Run *M* on *w*;. . . . .

   . . . . x++;. . . . .

   . . . . } . . . .

   . . . .”

2. Output <. . *N*, *x*, *y* . . .$_3$>."

Now, <. . . *N*, *x*, *y*. . . .$_3$> is an element of . . . L3. . . . . iff <. . . *M*, *w*. . .$_2$> is an element of . . HALT. . ..

So, using *f* and the assumption that . . . L3. . .$_1$ is decidable, we can decide . HALT. . . . . .. A contradiction.

Therefore, . . L3. . . . .$_1$ is undecidable. (This also means that the complement of . . . .L3 . . . .$_1$ is undecidable; the complement of any undecidable language is itself undecidable.)

3 Prove that the problem associated with language L_3 defined below is undecidable. You are given that HALT = {<*M*, *w*>: *M* is a Turing machine and *M* halts on *w*} is undecidable. Use the template provided to perform a mapping reduction. You must give your answer on this exam question sheet.

The language L_3 is defined as
L_3 = {<*M, q*>: *M* is a TM that never goes into state *q* ~~when *M* is run~~}.

Note, in the template below, some blanks have a small subscript number. Blanks with the same subscript number must have the same value.

<u>Proof</u>

Optional: $\overline{L_3}$ = . . {<*M, q*>: *M* is a TM that when run goes into state *q* at least once}.. . . .

We will use a mapping reduction to prove the reduction

. . HALT. . . ≤ . . $\overline{L_3}$ . . . . ..

Assume that . . . .$\overline{L_3}$. . .$_{\underline{1}}$ is decidable.

The transition function *f* that maps instances of . . . HALT . . . . to instances of . . . $\overline{L_3}$. . . . is given by TM F given by the following pseudocode.

F = "On input <. .*M*, *w*. . .$_{\underline{2}}$>:

1. Construct the following *N* given by the following pseudocode.

   N = ". .On any input:. . . . .

   . . . . . Let *q* be some state not in *M*. . . . .

   . . . . . Run *M* on *w* . . . .

   . . . . . Go into state *q*. . . . .

   . . . .

   . . . .”

2. Output <. . . *N*, *q*. .$_{\underline{3}}$>."

Now, <. . . . *N*, *q* . . . .$_{\underline{3}}$> is an element of . . $\overline{L_3}$ . . . . iff <. . . *M*,*w*. . .$_{\underline{2}}$> is an element of . .HALT. . ..

So, using *f* and the assumption that . . $\overline{L_3}$. .$_{\underline{1}}$ is decidable, we can decide

. HALT. .. A contradiction.

Therefore, . . $\overline{L_3}$ . . .$_{\underline{1}}$ is undecidable. (This also means that the complement of . . . $\overline{L_3}$ . . . .$_{\underline{1}}$ is undecidable; the complement of any undecidable language is itself undecidable.)

3 Prove that the problem associated with language L_3 defined below is undecidable. You are given that HALT = {<*M*, *w*>: *M* is a Turing machine and *M* halts on *w*} is undecidable. Use the template provided to perform a mapping reduction. You must give your answer on this exam question sheet.

The language L_3 is defined as
L_3 = {<*J, i*>: *J* is a Java program and *i* is a nonnegative integer, and when *J* is run it never executes line number *i*}.

Note, in the template below, some blanks have a small subscript number. Blanks with the same subscript number must have the same value.

Proof

Optional: $\overline{L}_3$ = {<*J, i*>: *J* is a Java program and *i* is a nonnegative integer, and when *J* is run it executes line number *i* at least once}

We will use a mapping reduction to prove the reduction

. . .HALT . . . ≤ . . . $\overline{L}_3$ . . . . ..

Assume that . . . $\overline{L}_3$. . . .$_{1}$ is decidable.

The transition function *f* that maps instances of . . HALT. . . . . to instances of . . .$\overline{L}_3$. . . . is given by TM *F* given by the following pseudocode.

F = "On input <. .*M*, *w* . .$_{2}$>:

1. Construct the following *N* given by the following pseudocode.

   N = ".0: .void main(void) { . . . .(line number 0). . . . .

   . . 1: . . Run *M* on *w* . . . . (line number 1). . . . .

   . . 2: . . .int *x* = 10;. . . . . . (line number 2)

   . . 3:. . . }. . . . . (line number 3) . . . .

   . . . .

   . . . ."

2. Output <. . *N*, 2 . . .$_{3}$>."

Now, <. . .N, *i* . . . .$_{3}$> is an element of . . $\overline{L}_3$ . . . . iff <. . . *M*, *w*. . .$_{2}$> is an element of . . HALT . ..

So, using *f* and the assumption that . . $\overline{L}_3$. . . .$_{1}$ is decidable, we can decide . . HALT . . .. A contradiction.

Therefore, . . $\overline{L}_3$. . . .$_{1}$ is undecidable. (This also means that the complement of . . $\overline{L}_3$ . . .$_{1}$ is undecidable; the complement of any undecidable language is itself undecidable.)