Homework 8 - 10

Menu

HW#8 Problem 1

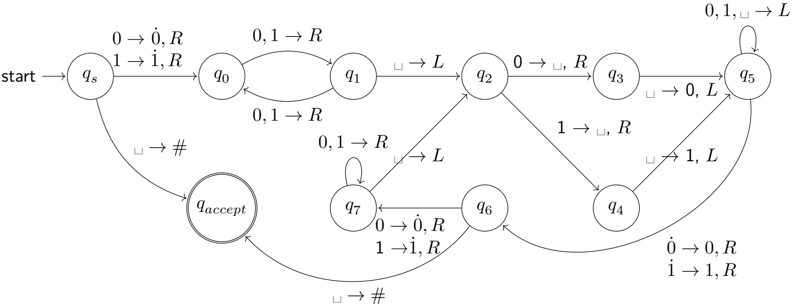
(10 points) Give a formal definition (with a state diagram) of a Turing machine that, given a string of an even length as the input, splits the input string into two halves and add a # in the middle to separate the two substrings. The input alphabet is $\{0, 1\}$.

HW#8 Problem 1

The Turing machine TM for the problem is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_s, q_{accept}, q_{reject})$, where

- Q is the set of states,
- $\Sigma = \{0, 1\}$,
- $\Gamma = \{0, 1, \dot{0}, \dot{1}, \#, \sqcup\}$,
- q_s is the start state,
- q_{accept} is the accept state, and
- q_{reject} is the reject state.

HW#8 Problem 1



HW#8 Problem 2

(Problem 3.10; 20 points) Let $c_1x^n + c_2x^{n-1} + \cdots + c_nx + c_{n+1}$ be a polynomial with a root at $x = x_0$. Let $c_{\max}$ be the largest absolute value of a c_i. Show that

$$|x_0| < (n+1)\frac{c_{\max}}{|c_1|}.$$

HW#8 Problem 2

Since x_0 is a root of the polynomial,
$c_1x_0^n + c_2x_0^{n-1} + ... + c_nx_0 + c_{n+1} = 0$

By triangle inequality,
$|c_1x_0^n + c_2x_0^{n-1} + ... + c_nx_0 + c_{n+1}| = 0 \leq$
$|c_1x_0^n| + |c_2x_0^{n-1}| + ... + |c_nx_0| + |c_{n+1}|$

Case 1 $|x_0| < 1$:
Since $(n+1)\frac{c_{max}}{|c_1|} \geq 1$, $|x_0| < (n+1)\frac{c_{max}}{|c_1|}$

HW#8 Problem 2

Case 2 $|x_0| \geq 1$:
We can get the upper bound of each term by c_{max}:
$|c_1 x_0^n| + |c_2 x_0^{n-1}| + ... + |c_n x_0| + |c_{n+1}| \leq n \cdot c_{max}|x_0^{n-1}| + c_{max}$
Divide $|c_1|$ on both sides:
$|x_0^n| + |\frac{c_2}{c_1}||x_0^{n-1}| + ... + |\frac{c_n}{c_1}||x_0| + |\frac{c_{n+1}}{c_1}| \leq (n+1) \cdot \frac{c_{max}}{|c_1|}|x_0^{n-1}|$

Since $|\frac{c_2}{c_1}||x_0^{n-1}| + ... + |\frac{c_n}{c_1}||x_0| + |\frac{c_{n+1}}{c_1}|$ are all positive,

$|x_0^n| \leq (n+1) \cdot \frac{c_{max}}{|c_1|}|x_0^{n-1}|$

$|x_0| \leq (n+1)\frac{c_{max}}{|c_1|}$

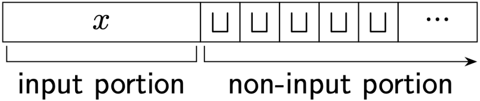
HW#8 Problem 3

(Problem 3.11; 20 points) Show that single-tape TMs that cannot write on the portion of the tape containing the input string recognize only regular languages.

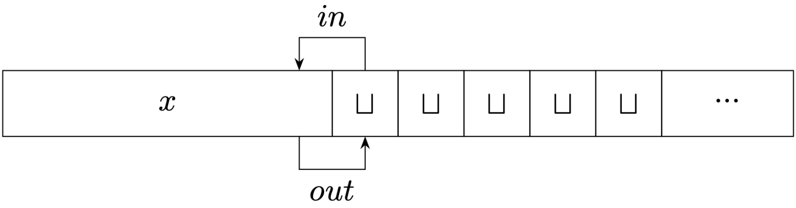
HW#8 Problem 3

Let $M = (Q, \Sigma, \Gamma, q_0, q_{accept}, q_{reject})$ be a single-tape TM that cannot write on the input portion of the tap. A typical case when M works on an input string x is as follows:

the tape head will stay in the input portion for some time, and then enter the non-input portion (i.e., the portion of the tape on the right of the $|x|^{th}$ cells) and stay there for some time, then go back to the input portion, and stay there for some time, and then enter the non-input portion, and so on.



HW#8 Problem 3



We call the event that the tape head switches from input portion to non-input portion an *out* event, and the event that the tape head switches from non-input portion to input-portion an *in* event.

HW#8 Problem 3

Let $first_x$ denote the state that M is in just after its first "out" event (i.e., the state of M when it first enters the non-input portion).

In case M never enters the non-input portion, we assign $first_x = q_{accept}$ if M accepts x, and assign $first_x = q_{reject}$ if M does not accept x.

HW#8 Problem 3

Next, we define a characteristic function f_x such that for any $q \in Q$, $f_x(q) = q'$ implies that if M is at state q just after its "in" event, M will move to state q' after its next "out" event.

In case M never enters the non-input portion again, we assign $f_x(q) = q_{accept}$ if M enters the accept state inside the input portion, and q_{reject} otherwise.

HW#8 Problem 3

Now we can define the binary relation R_L over Σ^* for the language L of TM M as follows:

$x \ R_L \ y$ iff

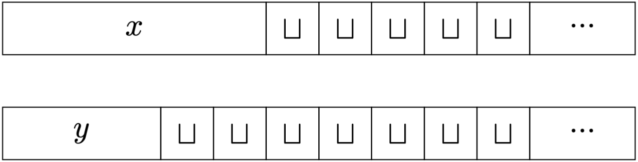
- $first_x = first_y$, and
- for all q, $f_x(q) = f_y(q)$.

We can observe the following property (requirements for Myhill-Nerode Theorem):

$x \ R_L \ y$ iff x and y are indistinguishable by L
(namely, $x \ R_L \ y$ iff $\forall z \in \Sigma^*(xz \in L \leftrightarrow yz \in L)$)

$Why?$

HW#8 Problem 3

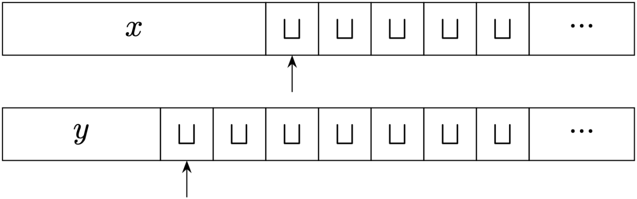


Let we consider two strings x and y with the same $first$ and f:

$Situation\ 1$:

If $first_x = first_y = (q_{accept}$ or $q_{reject})$, x and y will both be accepted or rejected at the same time before "out" event happens.

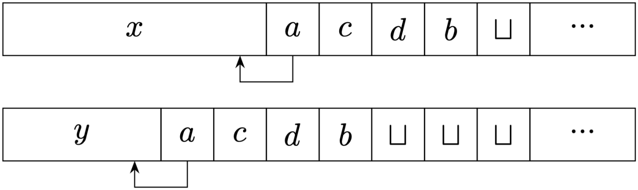
HW#8 Problem 3



$Situation$ 2:

If $first_x = first_y = q \neq (q_{accept}$ or $q_{reject})$, M_x and M_y will stay in the same state q and the heads of them stay in the same position of empty portion of two tapes ,which means that M_x and M_y will take the same actions in this portion (write the same symbol and move to the same state, i.e. if M_x accepts, M_y accepts at the same time).

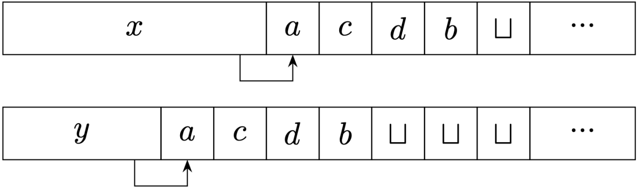
HW#8 Problem 3



Situation 2 (*cont.*):
How about "in" event happens?

Situation 2-1:
Because for all q, $f_x(q) = f_y(q)$, and M_x and M_y stay at the same state q when they are about to perform the "in" event, if $f_x(q) = f_y(q) = (q_{accept}$ or $q_{reject})$, similarly, x and y will both be accepted or rejected at the same time inside the input portion.

HW#8 Problem 3



Situation 2-2:
If $f_x(q) = f_y(q) = q' \neq (q_{accept}$ or $q_{reject})$, M_x and M_y will stay in the same state q' and the heads of them stay in the same position of non-input portion of two tapes (not empty now, but with the same string). Similarly, M_x and M_y will take the same actions in this portion.

If "in" event happens again, *Situation* 2 will happen repeatedly until M_x and M_y accept or reject.

HW#8 Problem 3

x	z	$\sqcup$	$\cdots$

y	z	$\sqcup$	$\sqcup$	$\sqcup$	$\cdots$

Now consider the strings xz and yz, you may notice that it is similar to $Situation$ 2-2, the non-input portion is not empty doesn't affect M_x and M_y to take the same actions in this portion.

So, M accepts xz if and only if M accepts yz, i.e. x and y are indistinguishable by M.

HW#8 Problem 3

In this situation, we say that x and y are in the same equivalence class (all strings in an equivalence class are indistinguishable to each other).

How many possibilities are there at most for the equivalence classes of M?

- $first_x$ has $|Q|$ possibilities.
- $f_x(q)$ has $|Q|$ possibilities for each $q \in Q$, i.e. $|Q|^{|Q|}$ possibilities totally.

So, there are at most $|Q|^{|Q|+1}$ equivalence classes, that is, the number of distinguishable strings are finite (R_L is of finite index). By Myhill-Nerode theorem, the language L is regular.

HW#8 Problem 4

(Problem 3.12; 20 points) Show that every infinite Turing-recognizable language has an infinite decidable subset.

HW#8 Problem 4

Let A be an infinite Turing-recognizable language, then there exists an enumerator E that enumerates all strings in A.
We can construct an enumerator E' that prints a subset of A in lexicographic order:

1. Simulate E, when E prints its first string w_1, print w_1 and let $w_r = w_1$.
2. Continue simulating E.
3. When E is ready to print a new string w, check if w is longer than w_r (this ensures w occurs after w_r in standard order(order by length)). If so, then print w and let $w_r = w$, otherwise do not print w.
4. Go to 2.

HW#8 Problem 4

The language of E' is infinite since A is infinite, there exist strings in A longer than the current w_r , which means E will eventually print one of these and so will E'.
The language of E' language is decidable since it prints strings in standard order.
Thus, the language of E' is an infinite decidable subset of A.

HW#8 Problem 5

(20 points) Let $A = \{\langle M, N\rangle \mid M \text{ is a PDA and } N \text{ is a DFA such that } L(M) \subseteq L(N)\}$. Show that A is decidable.

HW#8 Problem 5

Use the property: $A \subseteq B \Leftrightarrow A \cap \overline{B} = \emptyset$.
Let TM R decides E_{CFG}, we can construct a decider D as follows:

$D =$ "On input $\langle M, N \rangle$, where M is a PDA and N is a DFA:
1. Construct the complement $\overline{N}$ of N.
2. Construct a PDA P that recognizes the intersection of M and $\overline{N}$ (the intersection of a context-free language and a regular language is context free).
3. Let G_P be the context-free grammar that recognized by P, run R on input $\langle G_P \rangle$.
4. If R accepts, *accept*; otherwise, *reject*."

HW#8 Problem 6

(Exercise 4.4; 10 points) Let $A\varepsilon_{\mathrm{CFG}} = \{\langle G \rangle \mid G \text{ is a CFG that generates } \varepsilon\}$. Show that $A\varepsilon_{\mathrm{CFG}}$ is decidable.

HW#8 Problem 6

We can construct a decider D as follows:

$D =$ "On input $\langle G \rangle$, where G is a CFG:
1. Convert G to an equivalent grammar in Chomsky normal form G'.
2. If $(S_0 \to \epsilon) \in G'$, $accept$ (in Chomsky normal form, only S_0 can generate ϵ); otherwise, $reject$."

HW#8 Problem 6

Reduction method:

Let TM S decides A_{CFG}, we can construct a decider D as follows:

$D =$ "On input $\langle G \rangle$, where G is a CFG:
1. Run S on input $\langle G, \epsilon \rangle$.
2. If S accepts, $accept$; otherwise, $reject$."

HW#9 Problem 1

(Exercise 4.7; 10 points) Let B be the set of all infinite sequences over $\{0, 1\}$. Show that B is uncountable, using a proof by diagonalization.

HW#9 Problem 1

Suppose B is countable, we can draw a table of $n \in N$ and $f(n) \in B$.
For $n \in N$, $f(n) = b_{n1}b_{n2}b_{n3}...$

n	$f(n)$
1	110...
2	001...
3	100...
$\vdots$	$\vdots$

HW#9 Problem 1

Then define a infinite sequence $c = c_1c_2c_3... \in B$, where $c_i = 1 - b_{ii}$. Since c differs from the i-th sequence in the i-th bit, c doesn't equal to any $f(n)$, contradiction!
Therefore, B is uncountable.

HW#9 Problem 2

(Problem 4.12; 10 points) Let A be a Turing-recognizable language consisting of descriptions of Turing machines, $\{\langle M_1\rangle, \langle M_2\rangle, \ldots\}$, where every M_i is a decider. Prove that some decidable language D is not decided by any decider M_i whose description appears in A. (Hint: you may find it helpful to consider an enumerator for A.)

HW#9 Problem 2

A 是 Turing-recognizable language，包含了某些 Deciders
說明必然存在一個 decidable language D，它不能被 A 裡頭的任何 Decider 給 decide

HW#9 Problem 2

使用對角論證法：

題目提示告訴我們，既然 A 是 Turing-recognizable，就表示有一個 Enumerator E 可以生成 A
將 E 生成的第 i 個 TM 標記爲 M_i
而因爲 Σ^* 是可數集，存在一種排序法使對於任一個字串 $s \in \Sigma^*$ 而言，都能標記它出現的順序
於是可以做出一張表

HW#9 Problem 2

	s_1	s_2	...	s_i	...
M_1	<u>accept</u>	accept	...	reject	...
M_2	accept	<u>reject</u>	...	accept	...
$\vdots$	...	...	...	...	...
M_i	reject	accept	...	<u>reject</u>	...
$\vdots$	...	...	...	...	...

依照這張表，建構一個 TM M_D

$M_D =$ "On input s:

1. 計算出 s 在 Σ^* 當中的順位 i
2. 將 s 丟入 M_i 當中計算
3. If M_i accepts, reject; otherwise, accept."

這樣就能建構出一台與 A 當中的任何圖靈機都不一樣的機器

而且 M_i 本身是 Decider，這台機器一定會停機，所以 M_D 是 Decider，並且 M_D 在 input s_i 下會得到和 M_i 不同的結果

因此存在 decidable language D 不能被 A 中的任何 decider 所 decide

HW#9 Problem 2

這題能告訴我們什麼

一個存著「所有」Deciders 的語言

$D_{ALL} = \{\langle D \rangle \mid \text{D decides a language over } \Sigma^*\}$ 不可能是 Turing-recognizable

HW#9 Problem 3

(Problem 4.14; 20 points) Let $C = \{\langle G, x\rangle \mid G$ is a CFG and x is a substring of some $y \in L(G)\}$. Show that C is decidable. (Hint: an elegant solution to this problem uses the decider for E_{CFG}.)

HW#9 Problem 3

存在一個 decider，可以判斷 CFG G 是否會生成某個字串 y 使得 x 是它的子字串
那麼就是要把 $L(G)$ 與 $\Sigma^* x \Sigma^*$ 這兩個 language 取交集
一個 CFL 與 RL 的交集也是 CFL (將 PDA 與 DFA 的狀態合在一起做成新的 PDA)
再把交集出來的語言丟到 E_{CFG} 的 Decider 裡頭即可

HW#9 Problem 3

M = "On input $\langle G, x\rangle$ where G is a CFG:
1. Construct a CFG G ' s.t. $L(G') = L(G) \cap \Sigma^* x \Sigma^*$
2. Run $M_{E_{CFG}}$ on input $\langle G'\rangle$
3. If $M_{E_{CFG}}$ accept, reject; otherwise, accept."

上面每個步驟都能在有限時間完成，所以得 C 是 decidable

HW#9 Problem 4

(Problem 4.16; 10 points) Let $PAL_{\text{DFA}} = \{\langle M \rangle \mid M$ is a DFA that accepts some palindrome$\}$. Show that PAL_{DFA} is decidable. (Hint: Theorems about CFLs are helpful here.)

HW#9 Problem 4

Suppose TM R decides E_{CFG}, and P is a PDA which $L(P) = \{w | w$ is a palindrome$\}$.
We can construct a decider D that decides PAL_{DFA},

D $=$ " On input $\langle M \rangle$, M is a DFA
1. Construct a PDA P' which $L(P') = L(P) \cap L(M)$ (the intersection of a regular language and a context-free language is context-free).
2. Convert P' into an equivalent CFG G.
3. Run R on $\langle G \rangle$.
4. If R accepts, reject; otherwise, accept."

Since R is a decider and D runs in finite steps, PAL_{DFA} is decidable.

HW#9 Problem 5

(Problem 4.31; 20 points) Let $\mathit{INFINITE}_{\mathrm{PDA}} = \{\langle M \rangle \mid M \text{ is a PDA and } L(M) \text{ is infinite}\}$. Show that $\mathit{INFINITE}_{\mathrm{PDA}}$ is decidable.

HW#9 Problem 5

判定 PDA 辨識的字串是否有無限多個
由 pumping lemma 可以知道，只要 CFL 內有個字串 s 長度有 pumping length p 以上，就可以生成無限多個字串也在 CFL 內
而且此時一定會有一個長度介於 p 與 2p 之間的字串:
如果 $p \leq |s| \leq 2p$ 那就有了

HW#9 Problem 5

Y = "On input $\langle M \rangle$ where M is a PDA:
1. Convert M to a CFG G and compute G ' s pumping length p.
2. Construct a regular expression E that contains all strings of length p or more.
3. Construct a CFG H such that $L(H) = L(G) \cap L(E)$
4. Test $L(H) = \emptyset$, using the E_{CFG} decider R.
5. If R accepts, reject; if R rejects, accepts."

HW#9 Problem 6

(Exercise 5.1; 10 points) Show that EQ_{CFG} is undecidable.

HW#9 Problem 6

The idea is to reduce ALL_{CFG} to EQ_{CFG}.
Assume that a TM R decides EQ_{CFG}.
Construct a CFG G' which $L(G') = \Sigma^*$.
We could then construct a decider S for ALL_{CFG} as follows:

$S =$ " On input $\langle G \rangle$, G is a CFG:
1. Run TM R on input $\langle G, G' \rangle$.
2. If R rejects, reject; if R accepts, accepts. "

But we've known that ALL_{CFG} is undecidable, so EQ_{CFG} is undecidable.

HW#9 Problem 7

(Exercise 5.4; 20 points) If A is reducible to B and B is a regular language, does that imply that A is a regular language? Why or why not?

HW#9 Problem 7

利用反例說明 A 不一定是 regular language

假設 A 是 context-free language, 對應的 CFG 爲 G,
B 是 regular language, $B = \{1\}$,
建構一個 computable function f 使得 $w \in A \iff f(w) \in B$,
令 $M_{A_{CFG}}$ decides A_{CFG},

f = "On input w:
1. Run $M_{A_{CFG}}$ on input $\langle G, w \rangle$
2. If $M_{A_{CFG}}$ accepts, output 1; otherwise, output 0"

HW#10 Problem 1

(Problem 5.9; 10 points) Let $AMBIG_{\mathrm{CFG}} = \{\langle G\rangle \mid G \text{ is an ambiguous CFG}\}$. Show that $AMBIG_{\mathrm{CFG}}$ is undecidable. (Hint: use a reduction from PCP. Given an instance

$$P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_k}{b_k}\right] \right\}$$

of PCP, construct a CFG G with the rules:

$$\begin{array}{lcl} S & \to & T \mid B \\ T & \to & t_1 T a_1 \mid \cdots \mid t_k T a_k \mid t_1 a_1 \mid \cdots \mid t_k a_k \\ B & \to & b_1 B a_1 \mid \cdots \mid b_k B a_k \mid b_1 a_1 \mid \cdots \mid b_k a_k, \end{array}$$

where $a_1, \ldots, a_k$ are new terminal symbols. Prove that this reduction works.)

HW#10 Problem 1

Assume that a TM D_{AMBIG} decides $AMBIG_{CFG}$, we can construct a decider D that decides PCP as follows: $D =$ "On input $\langle P \rangle$, where $R = \{\}$: 1. Construct a CFG G with the rules:

$$S \rightarrow T \mid B$$

$$T \rightarrow t_1 T a_1 \mid ... \mid tkTa_k \mid t_1 a_1 \mid ... \mid t_k a_k$$

$$B \rightarrow b_1 B a_1 \mid ... \mid bkBa_k \mid b_1 a_1 \mid ... \mid b_k a_k$$

2. Run D_{AMBIG} on input $\langle G \rangle$.
3. If D_{AMBIG} accepts, accept; otherwise, reject."

But we've known that PCP is undecidable, so $AMBIG_{CFG}$ is undecidable.

HW#10 Problem 2

(Problem 5.14(b); 20 points) Define a *two-headed finite automaton* (2DFA) to be a deterministic finite automaton that has two read-only, bidirectional heads that start at the left-hand end of the input tape and can be independently controlled to move in either direction. The tape of a 2DFA is finite and is just large enough to contain the input plus two additional blank tape cells, one on the left-end and one on the right-hand end, that serve as delimiters. A 2DFA accepts its input by entering a special accept state. For example, a 2DFA can recognize the language $\{a^n b^n c^n \mid n \geq 0\}$.

Let $E_{\text{2DFA}} = \{\langle M \rangle \mid M \text{ is a 2DFA and } L(M) = \emptyset\}$. Show that E_{2DFA} is undecidable.

HW#10 Problem 2

We can reduce E_{TM} to $E_{2\text{DFA}}$.

The idea is to construct a 2DFA that recognizes the accept computational history $c_1\#c_2\#\ldots\#c_n$ of a TM M.

To do so, the 2DFA checks if c_1 consists q_{start} and a symbol in Σ, and then checks if c_n consists q_{accept} and symbols in Σ.

For middle transitions, let one head on c_i and the other on c_{i+1} and check each states and symbols.

HW#10 Problem 2

Assume that a TM $D_{2\text{DFA}}$ decides $E_{2\text{DFA}}$, we can construct a decider D that decides E_{TM} as follows:

$D =$ "On input $\langle M \rangle$, where M is a TM:
1. Construct a 2DFA N from M as described in previous slide.
2. Run $D_{2\text{DFA}}$ on input $\langle N \rangle$.
3. If $D_{2\text{DFA}}$ accepts, *accept*; otherwise, *reject*."

But we've known that E_{TM} is undecidable, so $E_{2\text{DFA}}$ is undecidable.

HW#10 Problem 3

(Problem 5.18 adapted; 20 points) Please discuss briefly the applicability of Rice's theorem to proving the undecidability of each of the following languages.

(a) $REGULAR_{\mathrm{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is regular}\}$.

(b) $E_{\mathrm{LBA}} = \{\langle M \rangle \mid M \text{ is an LBA where } L(M) = \emptyset\}$.

HW#10 Problem 3

To show if Rice's theorem is applicable, check two things:

(1) The property is of a language recognized by a TM.

(2) The property is non-trivial.

The property is non-trivial if there exists a TM satifies it and one does not.

(a) Applicable. M is a TM and "$L(M)$ is regular "is a non-trivial property of a language.

(b) Not applicable. M is not a TM

HW#10 Problem 4

(Problem 5.22; 20 points) Let $X = \{\langle M, w\rangle \mid M$ is a single-tape TM that never modifies the portion of the tape that contains the input $w\}$. Is X decidable? Prove your answer.

HW#10 Problem 4

We can try to reduce A_{TM} to X.

Assume that a TM D_X decides X, we can construct a decider D that decides A_{TM} as follows:

$D =$ "On input $\langle M, w\rangle$, where M is a TM and w is a string:

1. Construct $M' =$ "On input u:
    1. Move to the right of u and put \$.
    2. Copy w after \$.
    3. Simulate M on the portion of w.
    4. If M accepts and u is not empty, modify any character of u and $accept$; otherwise, $reject$."
2. Run D_X on input $\langle M', u\rangle$ for any non-empty string u.
3. If D_X accepts, $reject$; otherwise, $accepts$."

But we've known that A_{TM} is undecidable, so X is undecidable.

HW#10 Problem 5

(Problem 5.29; 10 points) A *useless state* in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a Turing machine has any useless states. Formulate this problem as a language and show that it is undecidable.

HW#10 Problem 5

$USELESS_{TM} = \{\langle M, q\rangle \mid q$ is a useless state in TM $M\}$.
Suppose that $USELESS_{TM}$ is decidable and that TM R decides it.
For any Turing machine M with accept state q_{accept}, q_{accept} is useless if and only if $L(M) = \emptyset$.
Since $E_{TM} = \{\langle M\rangle \mid M$ is a TM and $L(M) = \emptyset\}$
We can use R to check if q_{accept} is a useless state to decide E_{TM}.

HW#10 Problem 5

S = "On input $\langle M \rangle$, where M is a TM:
1. Run TM R on input $\langle M, q_{accept} \rangle$, where q_{accept} is the accept state of M.
2. If R accepts, accept. If R rejects, reject."

But we've known that E_{TM} is undecidable, so the problem of determining whether a TM has any useless states is undecidable.

HW#10 Problem 6

(10 points) Let $ALL_{\mathrm{DFA}} = \{\langle A\rangle \mid A \text{ is a DFA and } L(A) = \Sigma^*\}$. Prove that $ALL_{\mathrm{DFA}} \in \mathrm{P}$.

HW#10 Problem 6

We can construct a deterministic decider D that decides ALL_{DFA} in polynomial time as follows:

$D =$ "On input $\langle A \rangle$, where A is a DFA with n states:

$(O(1))$ 1. Mark the initial state of A.

$(O(n^2))$ 2. Mark the states of A that can be arrived from any marked states until no state can be marked.

$(O(n))$ 3. If there is any non-accepting state marked, *reject*; otherwise, *accepts*."

The decider D will decide ALL_{DFA} in $(O(n^2))$, so $ALL_{\text{DFA}} \in P$.

HW#10 Problem 7

(10 points) Two graphs G and H are said to be *isomorphic* if the nodes of G may be renamed so that it becomes identical to H. Let $ISO = \{\langle G, H\rangle \mid G \text{ and } H \text{ are isomorphic}\}$. Prove that $ISO \in \mathrm{NP}$, using the definition $\mathrm{NP} = \bigcup_k \mathrm{NTIME}(n^k)$.

HW#10 Problem 7

We can construct an nondeterministic polynomial time decider N decides ISO as follows:

$N =$ "On input $\langle G, H\rangle$ where $G(V, E)$ and $H(V', E')$ are undirected graphs:
1. If $|V| \neq |V'|$ or $|E| \neq |E'|$, $reject$.
2. Nondeterministically select a permutation π of m elements.
3. For all $\{(x, y)|x, y \in V\}$, check whether "$(x, y) \in E$ iff $(\pi(x), \pi(y)) \in E'$" is satisfied . If all agree, $accepts$. If any differ, $reject$.

Stage 2 can be implemented in polynomial time nondeterministically. (arbitrary pop a node x from V and repeat until V is empty)
Stages 1 and 3 takes polynomial time. Hence ISO $\in$ NP.