Write your answers in the separate answer booklet.

You have 180 minutes (after you get the answer booklet) to answer seven questions.

1. For each statement below, there are two boxes in the answer booklet labeled "Yes" and "No". Check "Yes" if the statement is ***always*** true and "No" otherwise, and give a ***brief*** (at most one short sentence) explanation of your answer. **Assume $P \neq NP$.** If there is any other ambiguity or uncertainty about an answer, check "No". For example:

   - $x + y = 5$

     [ ] Yes [X] No Suppose $x = 3$ and $y = 4$.

   - 3SAT can be solved in polynomial time.

     [ ] Yes [X] No 3SAT is NP-hard.

   - If P = NP, then Emily can beat Super Mario Brothers in four minutes and 53 seconds.

     [X] Yes [ ] No The hypothesis is false, so the implication is true.

   Read each statement *very* carefully; some of these are deliberately subtle!

   (a) Which of the following statements are *always* true?

   - The problem of deciding whether M accepts x for a given Turing machine M and a given string x is decidable.
   - The problem of deciding whether M accepts x within $2^{|x|}$ steps for a given Turing machine M and a given string x is decidable.
   - If $P = NP$, then all decidable problems are solvable in polynomial time.
   - If L is undecidable, then $\overline{L}$ is decidable.
   - If L is not context-free, then it is undecidable.

   (b) Suppose there is a ***polynomial-time*** many-one reduction from some language A over the alphabet $\{0, 1\}$ to some other language B over the alphabet $\{0, 1\}$. Which of the following statements are ***always*** true, assuming P $\neq$ NP?

   - A is a subset of B.
   - If $B \in P$, then $A \in P$.
   - If B is NP-hard, then A is NP-hard.
   - If B is regular, then A is regular.
   - If B is regular, then A is decidable.

Problems 2–7 are on the following pages.

2. ***More of the same.*** For each statement below, there are two boxes in the answer booklet labeled "Yes" and "No". Check "Yes" if the statement is ***always*** true and "No" otherwise, and give a ***brief*** (at most one short sentence) explanation of your answer. **Assume $P \neq NP$.** If there is any other ambiguity or uncertainty about an answer, check "No".

   (a) Which of the following statements are true?

   - The solution to the recurrence $T(n) = 8T(n/2) + O(n^2)$ is $T(n) = O(n^2)$.
   - The solution to the recurrence $T(n) = 2T(n/8) + O(n^2)$ is $T(n) = O(n^2)$.
   - Every directed acyclic graph contains at least one sink.
   - Given *any* undirected graph G, we can compute a spanning tree of G in $O(V+E)$ time using depth-first search.
   - Suppose $A[1..n]$ is an array of integers. Consider the following recursive function:

$$What(i,j) = \begin{cases} 0 & \text{if } i < 0 \text{ or } i > n \\ 0 & \text{if } j < 0 \text{ or } j > n \\ \max \left\{ \begin{array}{c} What(i, j-1) \\ What(i-1, j) \\ A[i] \cdot A[j] + What(i+1, j+1) \end{array} \right\} & \text{otherwise} \end{cases}$$

     We can memoize this function into an array $What[0..n, 0..n]$ in $O(n^2)$ time, by increasing i in the outer loop and increasing j in the inner loop.

   (b) Consider the following pair of languages:

   - DirHamPath $:= \{G \mid G \text{ is a directed graph with a Hamiltonian path}\}$
   - Acyclic $:= \{G \mid G \text{ is a directed acyclic graph}\}$

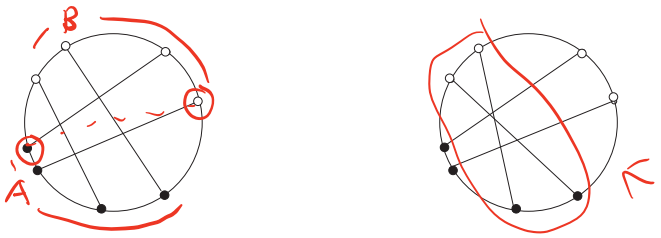
   (For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.)

   - Acyclic $\in$ NP
   - Acyclic $\cap$ DirHamPath $\in$ P
   - DirHamPath is decidable.
   - A polynomial-time many-one reduction from DirHamPath to Acyclic would imply P=NP.
   - A polynomial-time many-one reduction from Acyclic to DirHamPath would imply P=NP.

Problems 3–7 are on the following pages.

3. We are given a set A of n points on the lower half of a circle, and a set of B of n points on the upper half of the circle. (The points are given in arbitrary order, not necessarily sorted.) We want to find a set of n pairs $S = \{(a_1, b_1), \ldots, (a_n, b_n)\}$, with $\{a_1, \ldots, a_n\} = A$ and $\{b_1, \ldots, b_n\} = B$, maximizing the total length $\ell(S) = \sum_{i=1}^{n} d(a_i, b_i)$. Here, $d(u, v)$ denotes the Euclidean distance between u and v (which you may assume can be computed in $O(1)$ time).

   The example below shows two different pairings between A (shown in black) and B (shown in white), neither of which are optimal, but the pairing on the right has a larger total length.



   (a) Consider the following greedy strategy: pick a pair $(a, b) \in A \times B$ maximizing $d(a, b)$, remove a from A and b from B, and recursively pick the remaining pairs. Show that this greedy strategy does not always correctly find an optimal solution. (Your counterexample may be given in the form of a picture.)

   (b) Let a be the leftmost point in A (i.e., with the smallest x-coordinate). Let b be the rightmost point in B (i.e., with the largest x-coordinate). Prove that any optimal pairing T^* must use the pair (a, b).

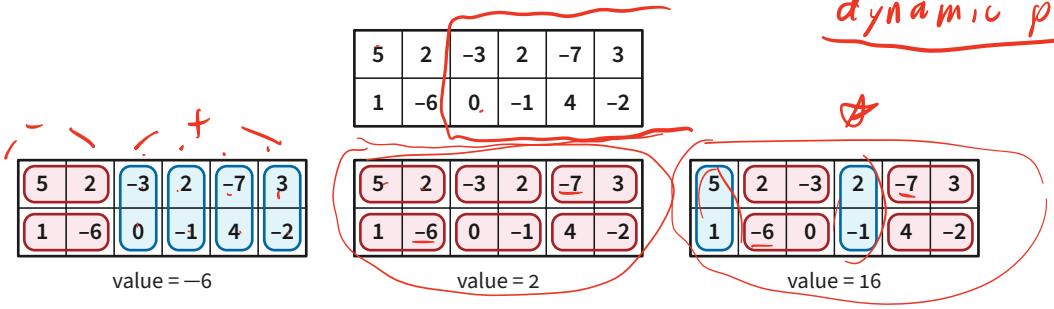
   [Hint: You may use the following geometric fact: for any 4 distinct points s, t, u, v on a circle in counterclockwise order, $d(s,u) + d(t,v) > d(t,u) + d(s,v)$.]

4. Suppose you are asked to tile a $2 \times n$ grid of squares with dominos (1×2 rectangles). Each domino must cover exactly two grid squares, either horizontally or vertically, and each grid square must be covered by exactly one domino.

   Each grid square is worth some number of points, which could be positive, negative, or zero. The ***value*** of a domino tiling is the sum of the points in squares covered by vertical dominos, ***minus*** the sum of the points in squares covered by horizontal dominos.

   Describe an algorithm to compute the largest possible value of a domino tiling of a given $2 \times n$ grid. Your input is an array $Points[1..2, 1..n]$ of point values.

   As an example, here are three domino tilings of the same 2×6 grid, along with their values. The third tiling is optimal; no other tiling of this grid has larger value. Thus, given this 2×6 grid as input, your algorithm should return the integer 16.



(not complete this time!)

5. ***Prove*** that the following problem (which we call MATCH) is NP-hard. The input is a finite set S of strings, all of the same length n, over the alphabet $\{0, 1, 2\}$. The problem is to determine whether there is a string $w \in \{0,1\}^n$ such that for every string $s \in S$, the strings s and w have the same symbol in at least one position.

   For example, given the set $S = \{01220, 21110, 21120, 00211, 11101\}$, the correct output is TRUE, because the string $w = 01001$ matches the first three strings of S in the second position, and matches the last two strings of S in the last position. On the other hand, given the set $S = \{2002, 2112, 2012, 2102\}$, the correct output is FALSE. w = 10?

   [Hint: Describe a reduction from SAT (or 3SAT)] ← free point!

6. Describe and analyze an algorithm to determine whether the language accepted by a given DFA is finite or infinite. You can assume the input alphabet of the DFA is $\{0,1\}$. *[Hint: DFAs are directed graphs.]*

7. Recall that a ***run*** in a string $w \in \{0,1\}^*$ is a maximal substring of w whose characters are all equal. For example, the string 00011111110000 is the concatenation of three runs:

$$00011111110000 = 000 \bullet 1111111 \bullet 0000$$

   (a) Let L_a denote the set of all strings in $\{0,1\}^*$ where every 0 is followed immediately by at least one 1.

   For example, L_a contains the strings 010111 and 1111 and the empty string ε, but does not contain either 001100 or 1111110.

   - Describe a DFA or NFA that accepts L_a ***and***
   - Give a regular expression that describes L_a.

   (You do not need to prove that your answers are correct.)

   (b) Let L_b denote the set of all strings in $\{0,1\}^*$ whose run lengths are increasing; that is, every run except the last is followed immediately by a *longer* run.

   For example, L_b contains the strings 0110001111 and 1100000 and 000 and the empty string ε, but does not contain either 000111 or 100011.

   Prove that L_b is not a regular language. infinite fooling set

CS/ECE 374 A ✧ ~~Fall 2025~~ Spring 2026

Practice Final Exam 3

May 10, 2026

Name:	Emily Fox
NetID:	ekfox

- ***Don't panic!***
- You have 180 minutes (3 hours) to answer seven questions. The questions are described in more detail in a separate handout.
- If you brought anything except your writing implements, your two **hand-written** double-sided 8½" × 11" cheat sheets, and your university ID, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
- Please clearly print your name and your NetID in the boxes above.
- Please also print your name at the top of every page of the answer booklet, except this cover page. We want to make sure that if a staple falls out, we can reassemble your answer booklet. (It doesn't happen often, but it does happen.)
- Greedy algorithms require formal proofs of correctness to receive any credit, even if they are correct. Otherwise, proofs or other justifications beyond items listed in the standard rubrics are required for full credit if and only if we explicitly ask for them, using the word ***prove*** or ***justify*** in bold italics.

- **Please do not write outside the black boxes on each page.** These indicate the area of the page that our scanners can actually scan. If the scanner can't see your work, we can't grade it.
- If you run out of space for an answer, please use the overflow/scratch pages at the back of the answer booklet, but **please clearly indicate where we should look.** If we can't find your work, we can't grade it.
- **Only work that is written into the stapled answer booklet will be graded.** In particular, you are welcome to detach scratch pages from the answer booklet, but any work on those detached pages will not be graded. We will provide additional scratch paper on request, but any work on that scratch paper will not be graded.
- Please return ***all*** paper with your answer booklet: your question sheet, your cheat sheet, and all scratch paper. **Please put all loose paper *inside* your answer booklet.**
- Breathe in. Breathe out. You've got this.

CS/ECE 374 A ✧ Fall 2025 Practice Final 1 Problem 1	Name: Emily Fox

(a) Which of the following statements are *always* true?

- The problem of deciding whether M accepts x for a given Turing machine M and a given string x is decidable.

  [Yes] [X No] If we can decide if a machine accepting Halt accepts, we've decided Halt.

- The problem of deciding whether M accepts x within $2^{|x|}$ steps for a given Turing machine M and a given string x is decidable.

  [X Yes] [No] Run M on x for $2^{|x|}$ steps. If it halts, return M(x), o.w. reject.

- If $P = NP$, then all decidable problems are solvable in polynomial time.

  [X Yes] [No] $P \neq NP$ (would have been No if we weren't assuming P≠NP)

- If L is undecidable, then $\overline{L}$ is decidable.

  [Yes] [X No] Could decide L by returning $\neg\overline{L}$'s decision on any string

- If L is not context-free, then it is undecidable.

  [Yes] [X No] $L = 3SAT$

(b) Suppose there is a ***polynomial-time*** (many-one) reduction from some language A over the alphabet $\{0,1\}$ to some other language B over the alphabet $\{0,1\}$. Which of the following statements are ***always*** true, assuming P ≠ NP?

- A is a subset of B.

  [Yes] [X No] Consider Ind Set ⇒ Clique reduction or ($\Sigma^* \to \{0,1\}$)

- If $B \in P$, then $A \in P$.

  [X Yes] [No] Given input x of A, reduce to B & return answer.

- If B is NP-hard, then A is NP-hard.

  [Yes] [X No] wrong direction. -or- Tree ∈P reduces to 3SAT.

- If B is regular, then A is regular.

  [Yes] [X No] Let A = Tree & B = {1}.

- If B is regular, then A is decidable.

  [X Yes] [No] $B \in P \Rightarrow A \in P \Rightarrow$ A is decidable

CS/ECE 374 A ✧ Fall 2025 Practice Final 1 Problem 2	Name: Emily Fox

(a) For each statement below, check "YES" if the statement is ***always*** true and "NO" otherwise, and give a ***brief*** (at most one short sentence) explanation of your answer. **Assume $P \neq NP$.** If there is any other ambiguity or uncertainty about an answer, write "NO".

- The solution to the recurrence $T(n) = 8T(n/2) + O(n^2)$ is $T(n) = O(n^2)$.

  [ ] Yes [X] No — increasing geometric => $\omega(n^2)$ ($\#\text{leaves} = 8^{\lg n} = n^{\lg 8} = n^3$)

- The solution to the recurrence $T(n) = 2T(n/8) + O(n^2)$ is $T(n) = O(n^2)$.

  [X] Yes [ ] No — decreasing geometric = O(root)

- Every directed acyclic graph contains at least one sink.

  [X] Yes [ ] No — If we repeatedly follow outgoing edges, we hit a sink or create a cycle

- Given *any* undirected graph $G = (V, E)$, we can compute a spanning tree of G in $O(V + E)$ time using depth-first search.

  [ ] Yes [X] No — G may not be connected.

- Suppose $A[1..n]$ is an array of integers. Consider the following recursive function:

$$What(i,j) = \begin{cases} 0 & \text{if } i < 0 \text{ or } i > n \\ 0 & \text{if } j < 0 \text{ or } j > n \\ \max \begin{Bmatrix} What(i, j-1) \\ What(i-1, j) \\ A[i] \cdot A[j] + What(i+1, j+1) \end{Bmatrix} & \text{otherwise} \end{cases}$$

We can memoize this function into an array $What[0..n, 0..n]$ in $O(n^2)$ time, by increasing i in the outer loop and increasing j in the inner loop.

[ ] Yes [X] No — What(i+1, j+1) is not yet computed.

Problem 2 continues on the next page.

(b) Consider the following pair of languages:

- DIRHAMPATH := $\{G \mid G$ is a directed graph with a Hamiltonian path$\}$
- ACYCLIC := $\{G \mid G$ is a directed acyclic graph$\}$

(For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.) Which of the following statements are true, assuming P $\neq$ NP?

- ACYCLIC $\in$ NP

  [X] Yes [ ] No Can check if topo sort exists in poly time. P $\subseteq$ NP.

- ACYCLIC $\cap$ DIRHAMPATH $\in$ P

  [X] Yes [ ] No Check if G in Acyclic. If so, see if G' topological order is a path.

- DIRHAMPATH is decidable.

  [X] Yes [ ] No DirHamPath $\in$ NP (certificate is the path)

- A polynomial-time reduction from DIRHAMPATH to ACYCLIC would imply P=NP.

  [X] Yes [ ] No Acyclic $\in$ P

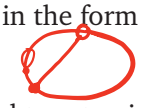
- A polynomial-time reduction from ACYCLIC to DIRHAMPATH would imply P=NP.

  [ ] Yes [X] No wrong direction for Acyclic $\in$ P to matter

CS/ECE 374 A ✧ Fall 2025 Practice Final 1 Problem 3	Name: Emily Fox

(See the questions handout for the problem description.)

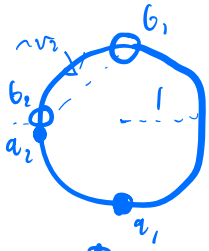
(a) Consider the following greedy strategy: pick a pair $(a, b) \in A \times B$ maximizing $d(a, b)$, remove a from A and b from B, and recursively pick the remaining pairs. Show that this greedy strategy does not always correctly find an optimal solution. (Your counterexample may be given in the form of a picture.)



(b) Let a be the leftmost point in A (i.e., with the smallest x-coordinate). Let b be the rightmost point in B (i.e., with the largest x-coordinate). Prove that any optimal pairing T^* must use the pair (a, b).

[Hint: You may use the following geometric fact: for any 4 distinct points s, t, u, v on a circle in counterclockwise order, $d(s,u) + d(t,v) > d(t,u) + d(s,v)$.]

(a)



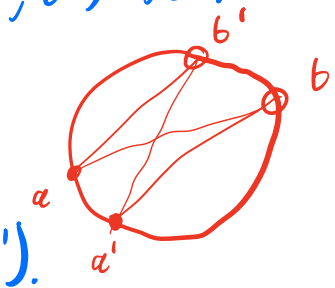
greedy: (a_1, b_1), (a_2, b_2) for total ~ 2
better: (a_1, b_2), (a_2, b_1) total $\sim 2 \cdot \sqrt{2} > 2$

(b) Let T^* be an optimal pairing.

Suppose $(a,b) \notin T^*$.

Let $(a, b') + (a', b) \in T^*$.

Replace $(a, b') + (a', b)$ with $(a, b) + (a', b')$ to make T.



Going CCW, we have a, a', b, b' in that order.

So $d(a,b) + d(a',b') > d(a',b) + d(a,b')$.

So T is a better solution. ↯

$(a,b) \in T^*$.

CS/ECE 374 A ✧ Fall 2025 Practice Final 1 Problem 4	Name: Emily Fox

Suppose you are asked to tile a $2 \times n$ grid of squares with dominos (1×2 rectangles). Each domino must cover exactly two grid squares, either horizontally or vertically, and each grid square must be covered by exactly one domino.

Each grid square is worth some number of points, which could be positive, negative, or zero. The ***value*** of a domino tiling is the sum of the points in squares covered by vertical dominos, ***minus*** the sum of the points in squares covered by horizontal dominos.

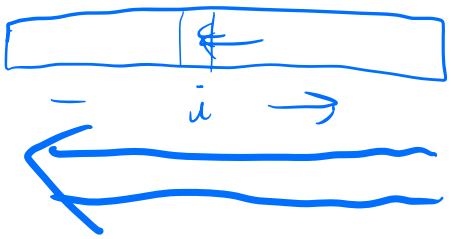
Describe an algorithm to compute the largest possible value of a domino tiling of a given $2 \times n$ grid. Your input is an array $Points[1..2, 1..n]$ of point values.

MaxValue(i): Max value from tiling the subarray of columns i through n.

Need to compute MaxValue(1).

$$MaxValue(i) = \begin{cases} 0 & \text{if } i > n \\ Points[1,n] + Points[2,n] & \text{if } i = n \\ \max \begin{cases} (Points[1,i] + Points[2,i] + MaxValue(i+1)) \\ (-Points[1,i] - Points[1,i+1] - Points[2,i] - Points[2,i+1] + MaxValue(i+2)) \end{cases} & \text{o.w.} \end{cases}$$

Memoize into a 1D array MaxValue[1..n+1].



-or- Fill in decreasing order of i.

Time: $O(1) \cdot O(n) = \underline{O(n)}$

CS/ECE 374 A ✧ Fall 2025 Practice Final 1 Problem 5	Name: Emily Fox

Prove that the following problem (which we call MATCH) is NP-hard. The input is a finite set S of strings, all of the same length n, over the alphabet $\{0, 1, 2\}$. The problem is to determine whether there is a string $w \in \{0,1\}^n$ such that for every string $s \in S$, the strings s and w have the same symbol in at least one position.

truth values of variables? $(a \vee \bar{b} \vee d)$

By a reduction from 3SAT:

Given a 3CNF Boolean formula F will construct an instance of MATCH.

F is over variable $x_1, x_2, \ldots, x_n$.

Let length of MATCH strings be n.

Start with $S \leftarrow \emptyset$.

For each clause C over variables $x_{i_1}, x_{i_2}, x_{i_3}$

add to S the string $s_C = 2^n$ with

the i_jth position set to 0

if x_{i_j} is False/negated in C or

1 if x_{i_j} is True/"pure" in C for

all $j \in \{1,2,3\}$.

Takes polynomial time $O(mn)$ where m: # clauses in F.

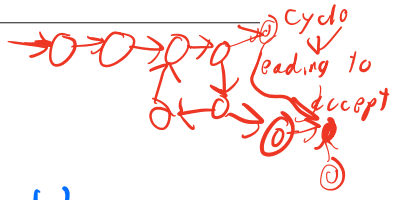
Need to show F can be satisfied iff we made a Yes instance of MATCH.

cont. on page 8

CS/ECE 374 A ✧ Fall 2025 Practice Final 1 Problem 6	Name: Emily Fox

Describe and analyze an algorithm to determine whether the language accepted by a given DFA is finite or infinite. You can assume the input alphabet of the DFA is {0, 1}. *[Hint: DFAs are directed graphs.]*

start → directed cycle → leading to accept



Let $M = (Q, \delta, s, A)$ be our DFA.

Build a graph $G = (V, E)$.

V: $Q \cup \{t\}$ $t \notin Q$

E: $\{(p, q) \mid \delta(p, 0) = q \text{ or } \delta(p, 1) = q\} \cup \{(r, t) \mid r \in A\}$.

DFA M accepts an infinite language iff

There is a vertex v on a cycle s.t. s can reach v & v can reach t.

Suffices to find one vertex v per cycle.

So run DFSAll(G) & for each back edge $(v, \text{post}[v], \text{post}[u])$ (u, v), mark v as on a cycle.

Run DFS(s) & note which vertices are marked & reachible from s.

G': reversal of G

Run DFS(t) in G' & note which vertices are reachible from t in G' / can reach t in G.

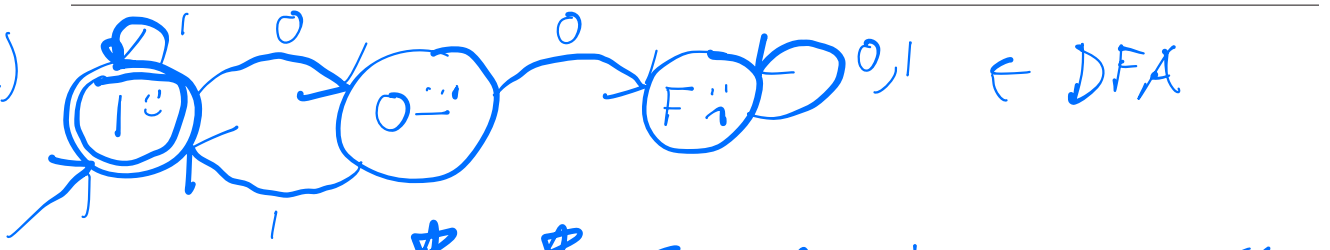
Return infinite if any vertex v noted in all three steps, o.w. finite. Time is $O(|V| + |E|) = O(|Q|)$

CS/ECE 374 A ✧ Fall 2025 Practice Final 1 Problem 7	Name: Emily Fox

(a) Let L_a denote the set of all strings in $\{0,1\}^*$ where every 0 is followed immediately by at least one 1. Describe a DFA or NFA that accepts L_a ***and*** give a regular expression that describes L_a. (You do not need to prove that your answers are correct.)

(b) Let L_b denote the set of all strings in $\{0,1\}^*$ whose run lengths are increasing; that is, every run except the last is followed immediately by a *longer* run. ***Prove*** that L_b is not a regular language.

(a)



← DFA

$(1^*01)^*1^*$ ← Regular expression

(b) Let $F = 0^*$.

Let $x \neq y$ be any distinct members of F.

$x = 0^i$ & $y = 0^j$ for $i, j \geq 1$ & $i \neq j$.

Assume $j > i$ (relabel x & y if necessary)

Let $z = 1^j$.

$xz = 0^i 1^j \in L_b$, because $j > i$.

$yz = 0^j 1^j \notin L_b$, because $j \not> j$.

So z distinguishes x & y. They were arbitrary, so F is an infinite fooling set.

$\Rightarrow$ Suppose there is a satisfying assignment A for F.
Let $w \in \{0,1\}^n$ s.t. the ith bit is 1 iff x_i is True in A.
For every string $s_c \in S$, we satisfied a literal x_i or $\bar{x}_i$ in C. The ith position of the string s_c matches our choice for w's ith bit.

$\Leftarrow$ Suppose MATCH got a Yes instance that uses a string $w \in \{0,1\}^n$.
Make an assignment A for F where x_i is True iff ith bit of w is 1.
For every clause C in F, we could only have matched on of three positions in s_c that correspond to variables of C. The matched position's literal in C is True by choice of that position's character.

(overflow / scratch paper)

(overflow / scratch paper)

(overflow / scratch paper)

(overflow / scratch paper)

Some useful NP-hard problems. You are welcome to use any of these in your own NP-hardness proofs, except of course for the specific problem you are trying to prove NP-hard.

3SAT: Given a boolean formula in conjunctive normal form, with exactly three distinct literals per clause, does the formula have a satisfying assignment?

INDEPENDENTSET: Given an undirected graph G and an integer k, is there a subset of k vertices in G that have no edges among them?

CLIQUE: Given an undirected graph G and an integer k, is there a subset of k vertices in G where there is an edge between every pair of them?

VERTEXCOVER: Given an undirected graph G and an integer k, is there a subset of k vertices that touch every edge in G?

DOMINATINGSET: Given an undirected graph G and an integer k, is there a subset of k vertices such that every vertex of G is either in the set or adjacent to a member of the set?

SETCOVER: Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set S and an integer k, is there a subcollection of k of these subsets whose union is S?

HITTINGSET: Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set S and an integer k, is there a subset of S of size k that intersects every subset S_i?

3COLOR: Given an undirected graph G, can its vertices be colored with three colors so that every edge touches vertices with two different colors?

HAMILTONIANPATH: Given graph G (either directed or undirected), is there a path in G that visits every vertex exactly once?

HAMILTONIANCYCLE: Given a graph G (either directed or undirected), is there a cycle in G that visits every vertex exactly once?

TRAVELINGSALESMAN: Given a graph G (either directed or undirected) with weighted edges and a number L, is there a Hamiltonian path/cycle of weight at most L in G?

LONGESTPATH: Given a graph G (either directed or undirected, possibly with weighted edges) and a number L, is there a simple path of length at least L in G?

STEINERTREE: Given an undirected graph G with some of the vertices marked and an integer k, is there a subtree of G with at most k edges that contains every marked vertex?

SUBSETSUM: Given a set X of positive integers and an integer k, does X have a subset whose elements sum to k?

PARTITION: Given a set X of positive integers, can X be partitioned into two subsets with the same sum?

3PARTITION: Given a set X of $3n$ positive integers, can X be partitioned into n three-element subsets, all with the same sum?

DRAUGHTS: Given an $n \times n$ international draughts configuration and an integer k, is there a move that can (and therefore must) capture at least k pieces in a single move?