	CS 341 Practice Final
	1. Short answers:
CS 341 Practice Final Marvin K. Nakayama Computer Science Dept., NJIT	 (a) Define the following terms and concepts: i. Union, intersection, set concatenation, Kleene-star, set subtraction, complement Answer: Union: S ∪ T = { x x ∈ S or x ∈ T } Intersection: S ∩ T = { x x ∈ S and x ∈ T } Concatenation: S ∘ T = { xy x ∈ S, y ∈ T } Kleene-star: S* = { w₁w₂ ··· w_k k ≥ 0, w_i ∈ S ∀ i = 1, 2,, k} Subtraction: S − T = { x x ∈ S, x ∉ T } Complement: S = { x ∈ Ω x ∉ S } = Ω − S, where Ω is the universe of all elements under consideration. ii. A set S is closed under an operation f Answer: S is closed under f if applying f to members of S characterize a member f G
CS 341 Practice Final 3	CS 341 Practice Final
iii. Regular language	vii. Church-Turing Thesis
Answer: A regular language is defined by a DFA.	Answer: The informal notion of algorithm corresponds exactly to a Turing machine that always halts (i.e., a decider).
Answer: A language is regular if and only if it has a regular expression. v. Context-free language Answer: A CFL is defined by a context-free grammar (CFG). vi. Chomsky normal form Answer: A CFG is in Chomsky normal form if each of its rules has one of 3 forms: $A \rightarrow BC, A \rightarrow x, \text{or} S \rightarrow \varepsilon,$ where A, B, C are variables, B and C are not the start variable, x is a terminal, and S is the start variable.	 viii. Turing-decidable language Answer: A language A that is decided by a Turing machine; i.e., there is a Turing machine M such that M halts and accepts on any input w ∈ A, and M halts and rejects on input input w ∉ A. Looping cannot happen. ix. Turing-recognizable language Answer: A language A that is recognized by a Turing machine; i.e., there is a Turing machine M such that M halts and accepts on any input w ∈ A, and



x. co-Turing-recognizable language

Answer: A language whose complement is Turing-recognizable.

xi. Countable and uncountable sets

Answer:

- A set S is countable if it is finite or we can define a correspondence between the positive integers and S.
- \bullet In other words, can create (possibly infinite) list of all elements in S and each specific element will eventually appear in list.
- An uncountable set is a set that is not countable.
- A common approach to prove a set is uncountable is by using a diagonalization argument.

CS 341 Practice Final

5

7

xii. Language A is mapping reducible to language $B,\,A\leq_{\rm m}B$

Answer:

- Suppose A is a language defined over alphabet Σ₁, and B is a language defined over alphabet Σ₂.
- Then $A \leq_{m} B$ means there is a computable function $f: \Sigma_{1}^{*} \to \Sigma_{2}^{*}$ such that $w \in A$ iff $f(w) \in B$.



 \iff

YES instance for problem A

 $w \in A$

 $f(w) \in B$ YES instance for problem B

CS 341 Practice Final

xiii. Function f(n) is O(g(n))

Answer: There exist constants c and n_0 such that $|f(n)| \le c \cdot g(n)$ for all $n \ge n_0$.

xiv. Classes P and NP

Answer:

CS 341 Practice Final

- P is the class of languages that can be decided by a **deterministic** Turing machine in polynomial time.
- NP is the class of languages that can be verified in (deterministic) polynomial time.
- Equivalently, NP is the class of languages that can be decided by a **nondeterministic** Turing machine in polynomial time.

xv. Language A is polynomial-time mapping reducible to language B , $A \leq_{\rm P} B.$

Answer:

- Suppose A is a language defined over alphabet Σ₁, and B is a language defined over alphabet Σ₂.
- Then $A \leq_{\mathsf{P}} B$ means \exists polynomial-time computable function $f: \Sigma_1^* \to \Sigma_2^*$ such that $w \in A$ iff $f(w) \in B$.



 $w \in A \iff f(w) \in B$ YES instance for problem $A \iff$ YES instance for problem B





(c) Explain the "P vs. NP" problem.

Answer:

- $\bullet\ P$ is class of languages that can be solved in deterministic poly time.
- $\bullet~\rm NP$ is class of languages that can be verified in deterministic poly time (equivalently, solved by poly-time NTM).
- We know that $P \subseteq NP$.
- Each poly-time DTM is also a poly-time NTM.
- But it is currently unknown if P = NP or $P \neq NP$.



CS 341 Practice Final

15

Detailed Proof:

- Suppose there exists a TM H that decides A_{TM} .
- Consider language
 - $L = \{ \langle M \rangle \mid M \text{ is a TM that does not accept } \langle M \rangle \}.$
- Now construct a TM D for L using TM H as a subroutine:
 - D = "On input \langle M \rangle, where M is a TM:
 1. Run H on input \langle M, \langle M \rangle.
 2. If H accepts, reject. If H rejects, accept."
- If we run TM D on input $\langle D \rangle$, then D accepts $\langle D \rangle$ if and only if D doesn't accept $\langle D \rangle$.
- \bullet Since this is impossible, TM H must not exist.

CS 341 Practice Final

13

- 2. Recall that $A_{\mathsf{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM that accepts string } w \}.$
- (a) Prove that $A_{\rm TM}$ is undecidable. You may not cite any theorems or corollaries in your proof.

Overview of Proof:

• Suppose A_{TM} is decided by some TM H, taking input $\langle M, w \rangle \in \Omega = \{ \langle M, w \rangle \mid M \text{ is a TM and } w \text{ a string } \}.$

• Define another TM (decider) D using H as a subroutine.

$$\langle M \rangle \longrightarrow \begin{bmatrix} D \\ \langle M, \langle M \rangle \rangle \longrightarrow \end{bmatrix} H \xrightarrow{accept} cecept ceceept cecept cecept cecept cecept cecept cecept cecep$$

- \bullet What happens when we run D with input $\langle D \rangle$?
 - $\blacksquare~D$ accepts $\langle D\rangle$ iff D doesn't accept $\langle D\rangle,$ which is impossible.

CS 341 Practice Final

(b) Show that A_{TM} is Turing-recognizable.

Answer: Universal TM (UTM) U recognizes A_{TM} :

- $U = \text{``On input } \langle M, w \rangle \in \Omega, \text{ where } M \text{ is a TM and } w \text{ is a string:} \\ 1. \text{ Run } M \text{ on } w.$
 - 2. If M accepts w, accept; if M rejects w, reject."
- U recognizes $A_{\rm TM}$ but does not decide $A_{\rm TM}$
- When we run M on w, there is the possibility that M neither accepts nor rejects w but rather loops on w.

CS 341 Practice Final 17	CS 341 Practice Final 18
 3. Each of the languages below in parts (a), (b), (c), (d) is of one of the following types: Type REG. It is regular. Type CFL. It is context-free, but not regular. Type DEC. It is Turing-decidable, but not context-free. For each of the following languages, specify which type it is. Also, follow these instructions: If a language L is of Type REG, give a regular expression and a DFA (5-tuple) for L. If a language L is of Type CFL, give a context-free grammar (4-tuple) and a PDA (6-tuple) for L. Also, prove that L is not regular. If a language L is of Type DEC, give a description of a Turing machine that decides L. Also, prove that L is not context-free. 	(a) $A = \{ w \in \Sigma^* w = \text{reverse}(w) \text{ and} $ the length of w is divisible by 4 }, where $\Sigma = \{0, 1\}$. Answer: A is of type CFL. A CFG $G = (V, \Sigma, R, S)$ for A has • $V = \{S\}$, • $\Sigma = \{0, 1\}$, • starting variable S , • rules $R = \{ S \rightarrow 00S00 01S10 10S01 11S11 \varepsilon \}$.
CS 341 Practice Final 19	CS 341 Practice Final 20
PDA for $A = \{w \in \Sigma^* \mid w = w^{\mathcal{R}}, w \text{ divisible by 4}\}:$ $\overbrace{q_1}^{(q_1)} \xrightarrow{\varepsilon, \varepsilon \to \$} \xrightarrow{q_2} \xrightarrow{\varepsilon, \varepsilon \to \varepsilon} \xrightarrow{q_4} \xrightarrow{\varphi} \xrightarrow{\varepsilon, \$ \to \varepsilon} \xrightarrow{q_6} \xrightarrow{\varphi} \xrightarrow{\varphi} \xrightarrow{\varphi} \xrightarrow{\varphi} \xrightarrow{\varphi} \xrightarrow{\varphi} \xrightarrow{\varphi} \varphi$	Prove $A = \{w \in \Sigma^* \mid w = w^{\mathcal{R}}, \text{length of } w \text{ is divisible by } 4\}$ nonregular. • For a contradiction, suppose that A is regular. • Pumping Lemma (Theorem 1.1): If L is regular language, then \exists number p where, if $s \in L$ with $ s \ge p$, then can split $s = xyz$ satisfying properties (1) $xy^{i}z \in L$ for each $i \ge 0$, (2) $ y > 0$, (3) $ xy \le p$ • Let $p \ge 1$ be the pumping length of the pumping lemma. • Consider string $s = 0^p 1^{2p} 0^p \in A$, and note that $ s = 4p > p$, so conclusions of pumping lemma must hold. • Since all of the first p symbols of s are 0s, (3) implies that x and y must only consist of 0s. Also, z must consist of rest of 0s at the beginning, followed by $1^{2p}0^p$. • Hence, we can write $x = 0^j$, $y = 0^k$, $z = 0^m 1^{2p} 0^p$, where $j + k + m = p$ since $s = 0^{p}1^{2p}0^p = xyz = 0^j 0^k 0^m 1^{2p} 0^p$. • Moreover, (2) implies that $k > 0$. • Finally, (1) states that $xyyz$ must belong to A . However, $xyyz = 0^j 0^k 0^k 0^m 1^{2p} 0^p = 0^{p+k} 1^{2p} 0^p$ since $j + k + m = p$. • But, $k > 0$ implies reverse $(xyyz) \neq xyyz$, which means $xyyz \notin A$, which contradicts (1).

CS 341 Practice Final	21CS 341 Practice Final22			
 (b) B = {bⁿaⁿbⁿ n ≥ 0}. Answer: B is of type DEC. Below is a description of a Turing machine that decides B. M = "On input string w ∈ {a, b}*: Scan input to check if it's in b*a*b*; reject if not. Return tape head to left-hand end of tape. Repeat following until no more b's left on tape. Replace the leftmost b with x. Scan right until a occurs. If no a's, reject. Replace the leftmost b (after the a's) with x. Replace the leftmost b (after the a's) with x. Replace the leftmost b (after the a's) with x. Menow prove that B is not context-free by contradiction. 	• Suppose that $B = \{b^n a^n b^n \mid n \ge 0\}$ is context-free. • PL for CFL (Thm 2.D): For every CFL L , \exists pumping length p such that $\forall s \in L$ with $ s \ge p$, can split $s = uvxyz$ with (1) $uv^i xy^i z \in L \forall i \ge 0$, (2) $ vy \ge 1$, (3) $ vxy \le p$. • Let p be pumping length of CFL pumping lemma • Consider string $s = b^p a^p b^p \in B$. Note that $ s = 3p > p$, so the pumping lemma will hold. • Thus, can split $s = b^p a^p b^p = uvxyz = \text{satisfying (1)-(3)}$ • We now consider all of the possible choices for v and y : • Suppose strings v and y are both uniform (e.g., $v = b^j$ for some $j \ge 0$, and $y = a^k$ for some $k \ge 0$). Then $ vy \ge 1$ implies that $v \ne \varepsilon$ or $y \ne \varepsilon$ (or both), so uv^2xy^2z won't have the correct number of b 's at the beginning, a 's in the middle, and b 's at the end. Hence, $uv^2xy^2z \notin B$. • Now suppose strings v and y are not both uniform . Then uv^2xy^2z won't have form $b \cdots ba \cdots ab \cdots b$, so $uv^2xy^2z \notin B$. • Every case gives contradiction, so B is not a CFL.			
CS 341 Practice Final (c) $C = \{ w \in \Sigma^* n_a(w) \mod 4 = 1 \}$, where $\Sigma = \{a, b\}$ and $n_a(w)$ is the number of a 's in string w . For example, $n_a(babaabb) = 3$. Also, $3 \mod 4 = 3$, and $9 \mod 4 = 1$. Answer: C is of type REG. A regular expression for C is $(b^*ab^*ab^*ab^*ab^*)^*b^*ab^*$	23 CS 341 Practice Final $C = \{ w \in \Sigma^* \mid n_a(w) \mod 4 = 1 \}$ DFA 5-tuple $(Q, \Sigma, \delta, q_1, F)$ • $Q = \{q_1, q_2, q_3, q_4\}$ • $\Sigma = \{a, b\}$ • q_1 is start state • $F = \{q_2\}$ • transition fcn $\delta : Q \times \Sigma \rightarrow Q$ $\frac{ a b }{q_1 q_2 q_1}$ $q_2 q_3 q_2$ $q_3 q_4 q_3$ $q_4 q_1 q_4$ $(Q = \{ w \in \Sigma^* \mid n_a(w) \mod 4 = 1 \}$			

(d) $D = \{ b^n a^n b^k c^k \mid n \ge 0, k \ge 0 \}.$ [Hint: Recall that the class of CFLs is closed under concatenation.] **Answer:** D is of type CFL.

A CFG $G = (V, \Sigma, R, S)$ for D has

$$\bullet V = \{S, X, Y\}$$

•
$$\Sigma = \{a, b, c\}$$

- \bullet starting variable S
- Rules R:

S	\rightarrow	XY	
X	\rightarrow	$bXa \mid$	ε
Y	\rightarrow	$bYc \mid $	ε

CS 341 Practice Final

PDA for
$$D = \{ b^n a^n b^k c^k \mid n \ge 0, k \ge 0 \}$$
:

Important: q_3 to q_4 pops and pushes \$ to make sure stack is empty. PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where $Q = \{q_1, q_2, \dots, q_6\}, \Sigma = \{a, b, c\}, \Gamma = \{b, \$\},$ q_1 is the start state, $F = \{q_6\}$, and the transition function $\delta : Q \times \Sigma_{\varepsilon} \times \Gamma_{\varepsilon} \to \mathcal{P}(Q \times \Gamma_{\varepsilon})$ is defined by

Input:	a			b	с			ε	
Stack:	b	\$ ε	b	\$ ε	b	\$ ε	b	\$	ε
q_1									$\{(q_2, \$)\}$
q_2				$\{(q_2, b)\}$					$\{(q_3,\varepsilon)\}$
q_3	$\{(q_3,\varepsilon)\}$							$\{(q_4, \$)\}$	
q_4				$\{(q_4, b)\}$					$\{(q_5,\varepsilon)\}$
q_5					$\{(q_5,\varepsilon)\}$			$\{(q_6,\varepsilon)\}$	
q_6									

Blank entries are \emptyset .

CS 341 Practice Final

CS 341 Practice Final

27

25

Prove $D = \{ b^n a^n b^k c^k \mid n \ge 0, k \ge 0 \}$ not regular.

- Suppose that D is regular. Let $p \ge 1$ be pumping length of pumping lemma (Theorem 1.I).
- Consider string $s = b^p a^p b^p c^p \in D$, and note that |s| = 4p > p, so conclusions of pumping lemma must hold.
- Thus, can split s = xyz satisfying (1) $xy^i z \in D$ for all $i \ge 0$, (2) |y| > 0, (3) $|xy| \le p$.
- Since all of the first p symbols of s are b's,
 (3) implies that x and y must consist of only b's.
- Also, z is rest of b's at beginning, followed by $a^p b^p c^p$.
- Hence, we can write $x = b^j$, $y = b^k$, $z = b^m a^p b^p c^p$, where j + k + m = p since
- $s = b^p a^p b^p c^p = xyz = b^j b^k b^m a^p b^p c^p.$
- Moreover, (2) implies that k > 0.
- Finally, (1) states that xyyz must belong to D, but $xyyz = b^j b^k b^k b^m a^p b^p c^p = b^{p+k} a^p b^p c^p$ since j + k + m = p. Also k > 0, so $xyyz \notin D$, which

contradicts (1). Therefore, D is a nonregular language.

4. Each of the languages below in parts (a), (b), (c), (d) is of one of the following types:

Type DEC. It is Turing-decidable.

Type TMR. It is Turing-recognizable, but not decidable.

Type NTR. It is not Turing-recognizable.

For each of the following languages, specify which type it is. Also, follow these instructions:

- \bullet If a language L is of Type DEC, give a description of a Turing machine that decides L.
- If a language L is of Type TMR, give a description of a Turing machine that recognizes L. Also, prove that L is not decidable.
- If a language L is of Type NTR, give a proof that it is not Turing-recognizable.

CS 341 Practice Final

In each part below, if you need to prove that the given language L is decideable, undecidable, or not Turing-recognizable, you must give an explicit proof of this; i.e., do not just cite a theorem that establishes this without a proof. However, if in your proof you need to show another language L' has a particular property for which there is a theorem that establishes this, then you may simply cite the theorem without proof.

- (a) $\overline{A_{\text{TM}}}$, where $A_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM that accepts string } w \}$. **Answer:** $\overline{A_{\text{TM}}}$ is of type NTR, which is just Theorem 4.M. Proof:
 - If $\overline{A_{\text{TM}}}$ were Turing-recognizable, then A_{TM} would be both Turing-recognizable (see slide 4-25) and co-Turing-recognizable.
 - But then Theorem 4.L would imply that A_{TM} is decidable, which we know is not true by Theorem 4.I.
 - Hence, $\overline{A_{\mathsf{TM}}}$ is not Turing-recognizable.

20

CS 341 Practice Final

(c) $HALT_{TM} = \{ \langle M, w \rangle | M \text{ is a TM that halts on input } w \}.$ [Hint: modify universal TM to show $HALT_{TM}$ is TM-recognizable.

Answer: $HALT_{TM}$ is of type TMR (see Theorem 5.A).

- Decision problem: Given TM M and string w, does M halt on input w?
- Universe: $\Omega_H = \{ \langle M, w \rangle \mid \mathsf{TM} \ M, \text{ string } w \}.$
- Consider following Turing machine T:
- $T = \text{``On input } \langle M, w \rangle \in \Omega_H \text{, where } M \text{ is TM and } w \text{ is string:} \\ 1. \text{ Run } M \text{ on } w.$
 - 2. If M halts (i.e., accepts or rejects) on w, accept."
- TM T recognizes $HALT_{TM}$
- ${\scriptstyle \blacksquare}$ accepts each $\langle M,w\rangle \in H\!ALT_{\mathsf{TM}}$
- \blacksquare loops on each $\langle M,w\rangle \not\in H\!ALT_{\mathsf{TM}}$

	(b) $EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are TMs with } L(M_1) = L(M_2) \}.$ [Hint: show $\overline{A_{TM}} \leq m EQ_{TM}]$
is	$[\text{Fine: show } T_{[M]} \leq M \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{j=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{j=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{j=1}^{M} \sum_{j=1}^{M} \sum_{i=1}^{M} \sum_{j=1}^{M} $
	Answer: EQ_{TM} is of type NTR (see Theorem 5.K).
t	Prove by snowing $A_{\text{TM}} \leq_{\text{m}} EQ_{\text{TM}}$ and applying Corollary 5.1.
C	• $A_{TM} \subseteq \Omega_1 = \{ \langle M, w \rangle \mid M \text{ is a TM}, w \text{ is a string } \},$
	$EQ_{TM} \subseteq \Omega_2 = \{ \langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are TMs } \}.$
	• Define reducing function $f(\langle M, w \rangle) = \langle M_1, M_2 \rangle$, where
•.	• $M_1 = "reject$ on all inputs."
	$M_2 = "On input x:$
	1. Ignore input x , and run M on w .
	2. If M accepts w , $accept$; if M rejects w , $reject$."
	• $L(M_1) = \emptyset$.
	• If M accepts w (i.e., $\langle M, w \rangle \notin \overline{A_{\text{TM}}}$), then $L(M_2) = \Sigma^*$.
	If M doesn't accept w (i.e., $\langle M, w \rangle \in \overline{A_{TM}}$), then $L(M_2) = \emptyset$.
	• Thus $\langle M, w \rangle \in \overline{A_{TM}} \iff f(\langle M, w \rangle) = \langle M_1, M_2 \rangle \in EQ_{TM}$
	so $\overline{A_{\text{TM}}} \leq EQ_{\text{TM}}$
	• But $\overline{A_{TM}}$ is not TM-recognizable (Corollary 4 M)
	• But A_{TM} is not TM-recognizable (Corollary 4.10),
	so D Q IM is not I wirecognizable by coronary s.n.
31	CS 341 Practice Final 32
	We now prove that $HALT_{TM}$ is undecidable, which is Theorem 5.A.
e.]	• We will show that A_{TM} reduces to $HALT_{\text{TM}}$, where
-	
	• $A_{\text{TM}} \subset \Omega_A = \{ \langle M w \rangle \mid \text{TM} M \text{string} w \}$
	• $A_{TM} \subseteq \Omega_A \equiv \{ \langle M, w \rangle \mid TM \ M, \text{ string } w \}$ - $HAIT_{TM} \subseteq \Omega_{TM} \equiv \{ \langle M, w \rangle \mid TM \ M, \text{ string } w \}$
	• $A_{TM} \subseteq \Omega_A \equiv \{ \langle M, w \rangle \mid TM \ M, \text{ string } w \}$ • $HALT_{TM} \subseteq \Omega_H \equiv \{ \langle M, w \rangle \mid TM \ M, \text{ string } w \}.$
	• $A_{TM} \subseteq \Omega_A \equiv \{ \langle M, w \rangle \mid TM \ M, \text{ string } w \}$ • $HALT_{TM} \subseteq \Omega_H \equiv \{ \langle M, w \rangle \mid TM \ M, \text{ string } w \}.$ • Suppose $\exists TM \ R$ that decides $HALT_{TM}.$
	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying
	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w.
<u>ъ</u> :	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w. S = "On input ⟨M, w⟩ ∈ Ω_A, where M is TM and w is string:
p.	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w. S = "On input ⟨M, w⟩ ∈ Ω_A, where M is TM and w is string: Run R on input ⟨M, w⟩.
5:	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w. S = "On input ⟨M, w⟩ ∈ Ω_A, where M is TM and w is string: R un R on input ⟨M, w⟩. R rejects. reject.
p.	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w. S = "On input ⟨M, w⟩ ∈ Ω_A, where M is TM and w is string: Run R on input ⟨M, w⟩. If R rejects, reject.
b:	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w. S = "On input ⟨M, w⟩ ∈ Ω_A, where M is TM and w is string: Run R on input ⟨M, w⟩. If R rejects, reject. If R accepts, simulate M on input w until it halts.
5 .	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w. S = "On input ⟨M, w⟩ ∈ Ω_A, where M is TM and w is string: Run R on input ⟨M, w⟩. If R rejects, reject. If R accepts, simulate M on input w until it halts. If M accepts, accept; otherwise, reject."
b.	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w. S = "On input ⟨M, w⟩ ∈ Ω_A, where M is TM and w is string: Run R on input ⟨M, w⟩. If R rejects, reject. If R accepts, simulate M on input w until it halts. If M accepts, accept; otherwise, reject." Since TM R is a decider, TM S always halts and decides A_{TM}.
p.	 A_{TM} ⊆ Ω_A ≡ { ⟨M, w⟩ TM M, string w } HALT_{TM} ⊆ Ω_H ≡ { ⟨M, w⟩ TM M, string w }. Suppose ∃ TM R that decides HALT_{TM}. Then could use R to build a TM S to decide A_{TM} by modifying UTM to first use R to check if it's safe to run M on w. S = "On input ⟨M, w⟩ ∈ Ω_A, where M is TM and w is string: Run R on input ⟨M, w⟩. If R rejects, reject. If R accepts, simulate M on input w until it halts. If M accepts, accept; otherwise, reject." Since TM R is a decider, TM S always halts and decides A_{TM}. However, A_{TM} is undecidable (Theorem 4.1),

33 CS 341 Practice Final	34
 5. • Let L₁, L₂, L₃, be an infinite sequence of regular language each of which is defined over a common input alphabet Σ. • Let L = ∪_{k=1}[∞] L_k be the infinite union of L₁, L₂, L₃, • Is it always the case that L is a regular language? • If your answer is YES, give a proof. • If your answer is NO, give a counterexample. • Explain your answer. • Hint: Consider, for each k ≥ 1, the language L_k = {a^kb^k}. 	Σ,
35 CS 341 Practice Final	36
ing of 6. Let L_1 , L_2 , and L_3 be languages defined over the alphabet $\Sigma = \{a, b\}$, where	
 F. L₁ consists of all possible strings over Σ except the strings w₁, w₂,, w₁₀₀; i.e., start with all possible strings over the alphabet take out 100 particular strings the remaining strings form the language L₁; L₂ is recognized by an NFA; and L₃ is recognized by a PDA. 	
M eŋ	 33 CS 341 Practice Final 5. • Let L₁, L₂, L₃, be an infinite sequence of regular languages each of which is defined over a common input alphabet Σ. (a) Let L = ∪_{k=1}[∞] L_k be the infinite union of L₁, L₂, L₃, • Is it always the case that L is a regular language? • If your answer is YES, give a proof. • If your answer is NO, give a counterexample. • Explain your answer. • Hint: Consider, for each k ≥ 1, the language L_k = {a^kb^k}. 35 CS 341 Practice Final 6. Let L₁, L₂, and L₃ be languages defined over the alphabet Σ = {a, b}, where • L₁ consists of all possible strings over Σ except the strings w₁, w₂,, w₁₀₀; i.e., • start with all possible strings over the alphabet • take out 100 particular strings • the remaining strings form the language L₁; • L₂ is recognized by an NFA; and

Answer:

- $\overline{L_1} = \{w_1, w_2, \dots, w_{100}\}$, so $|\overline{L_1}| = 100$. Thus, $\overline{L_1}$ is a regular language since it is finite by Theorem 1.F.
- Then Theorem 1.H implies that the complement of $\overline{L_1}$ must be regular, but the complement of $\overline{L_1}$ is L_1 . Thus, L_1 is regular.
- \bullet Language L_2 has an NFA, so it also has a DFA by Theorem 1.C. Therefore, L_2 is regular.
- Since L_1 and L_2 are regular, $L_1 \cap L_2$ must be regular by Theorem 1.G. Theorem 2.B then implies that $L_1 \cap L_2$ is CFL.
- Since L_3 has a PDA, L_3 is CFL by Theorem 2.C.
- Hence, since $L_1 \cap L_2$ and L_3 are both CFLs, their concatenation is CFL by Theorem 2.F.

- CS 341 Practice Final
 - 7. Write Y or N in the entries of the table below to indicate which classes of languages are closed under which operations.

	Regular		Decidable	Turing-recognizable
Operation	languages	CFLs	languages	languages
Union				
Intersection				
Complementation				

Answer:

CS 341 Practice Final

				T :
	Regular		Decidable	luring-recog
Ор	languages	CFLs	languages	languages
U	Y (Thm 1.A)	Y (Thm 2.E)	Y (HW 7, prob 2a)	Y (HW 7, prob 2b)
\cap	Y (Thm 1.G)	N (HW 6, prob 2a)	Y	Y
Compl.	Y (Thm 1.H)	N (HW 6, prob 2b)	Y (swap acc/rej)	N (e.g., A_{TM})

CS 341 Practice Final

39

37

8. Consider the following CFG G in Chomsky normal form:

 $\begin{array}{l} S \rightarrow a \mid YZ \\ Z \rightarrow ZY \mid a \\ Y \rightarrow b \mid ZZ \mid YY \end{array}$

Use CYK (dynamic programming) algorithm to fill in following table to determine if G generates string babba. Does G generate babba?

S	\rightarrow	$a \mid$	YZ
Z	\rightarrow	Z	$Y \mid a$



G does not generate babba because S is not in (1, 5) entry



 ${\scriptstyle \blacktriangle}$ within same triple

- \blacktriangle between contradictory literals, e.g., x_1 and $\overline{x_1}$
- Prove $\langle \phi \rangle \in 3SAT$ iff $\langle G, k \rangle \in CLIQUE$.



CS 341 Practice Final	9 CS	<i>5 341 Practice Final</i> 50			
3SAT ≤ _m ILP		3SAT ≤ _m ILP			
• Reductn $f: \Omega_3 \to \Omega_I$, $\langle \phi \rangle \in 3SAT$ iff $f(\langle \phi \rangle) = \langle A, b \rangle \in ILP$. • Consider 3cnf-formula with $m = 4$ variables and $k = 3$ clauses: $\phi = (x_1 \lor x_2 \lor \overline{x_3}) \land (\overline{x_1} \lor \overline{x_2} \lor x_4) \land (\overline{x_2} \lor \overline{x_4} \lor \overline{x_3})$ • Define integer linear program with • $2m = 8$ variables $y_1, y'_1, y_2, y'_2, y_3, y'_3, y_4, y'_4$ • y_i corresponds to x_i • y'_i corresponds to $\overline{x_i}$ • 3 sets of inequalities for each of pair y_i, y'_i : $0 \le y_1 \le 1, 0 \le y'_1 \le 1, y_1 + y'_1 = 1$ $0 \le y_2 \le 1, 0 \le y'_2 \le 1, y_2 + y'_2 = 1$ $0 \le y_3 \le 1, 0 \le y'_3 \le 1, y_3 + y'_3 = 1$ $0 \le y_4 \le 1, 0 \le y'_4 \le 1, y_4 + y'_4 = 1$ all hold with y_i, y'_i integer iff one of y_i, y'_i is 1, and other 0. • $0 \le y_i \le 1 \iff -y_i \le 0 \And y_i \le 1$ $y_i + y'_i = 1 \iff y_i + y'_i \le 1 \And y_i + y'_i \ge 1$		 Recall 3cnf-formula with m = 4 variables and k = 3 clauses: φ = (x₁ ∨ x₂ ∨ x₃) ∧ (x₁ ∨ x₂ ∨ x₄) ∧ (x₂ ∨ x₄ ∨ x₃) φ satisfiable iff each clause evaluates to 1. A clause evaluates to 1 iff at least one literal in the clause equals if For each clause (x_i ∨ x_j ∨ x_ℓ), create inequality y_i + y'_j + y_ℓ ≥ 1. For our example, ILP has inequalities y₁ + y₂ + y₃ ≥ 1 y'₁ + y'₂ + y₄ ≥ 1 y'₂ + y'₄ + y'₃ ≥ 1 which guarantee that each clause evaluates to 1. 			
CS 341 Practice Final	1 CS	5 341 Practice Final 52			
$3SAT \leq_m ILP$		Reducing 3SAT to ILP Takes Polynomial Time			
• Given 3cnf-formula:		$ullet$ Given 3cnf-formula ϕ with			
$\phi = (x_1 \lor x_2 \lor \overline{x_3}) \land (\overline{x_1} \lor \overline{x_2} \lor x_4) \land (\overline{x_2} \lor \overline{x_4} \lor \overline{x_3})$		• m variables: x_1, x_2, \ldots, x_m			
Constructed ILP:		■ <i>k</i> clauses			
$\begin{array}{lll} 0 \leq y_1 \leq 1, & 0 \leq y_1' \leq 1, & y_1 + y_1' = 1\\ 0 \leq y_2 \leq 1, & 0 \leq y_2' \leq 1, & y_2 + y_2' = 1\\ 0 \leq y_3 \leq 1, & 0 \leq y_3' \leq 1, & y_3 + y_3' = 1\\ 0 \leq y_4 \leq 1, & 0 \leq y_4' \leq 1, & y_4 + y_4' = 1\\ & y_1 + y_2 + y_3' \geq 1\\ & y_1' + y_2' + y_4 \geq 1\\ & y_2' + y_4' + y_3' \geq 1 \end{array}$ • Note that: $\phi \text{ satisfiable } \iff \text{ constructed ILP has solution}\\ (\text{with values of variables} \in \{0, 1\})\end{array}$		 Constructed ILP has 2m variables: y₁, y'₁, y₂, y'₂,, y_m, y'_m 6m + k inequalities: 3 sets of inequalities for each pair y_i, y'_i: 0 ≤ y_i ≤ 1, 0 ≤ y'_i ≤ 1, y_i + y'_i = 1, so total of 6m inequalities of this type. For each clause in φ. ILP has corresponding inequality e g 			
		$(x_1 \lor x_2 \lor \overline{x_3}) \longleftrightarrow y_1 + y_2 + y_3' \ge 1,$			
		so total of k inequalities of this type. • Thus, size of ILP is polynomial in m and k.			