

CS/ECE374 Cramming Carnival Worksheet

This worksheet aims to ask problems similar to what you might see on the exam, but wasn't created with knowledge of exam content. Answers are available on the HKN website (<https://hkn.illinois.edu/services>).

1 Regular Expressions

Write a regular expression for the following languages:

1. Strings that contain 01 as a substring, but not 10
2. Strings where every run of 1s has odd length
3. Strings that contain 101 as a subsequence
4. Strings representing a binary number divisible by 2
5. Strings representing a binary number that's a power of 2
6. Strings where runs of 1s alternate between odd and even length (starting with odd length)

2 Short Answer

Answer true or false for each question. Justify your answers.

1. If AB is not a context-free language, then at least one of A and B is not a CFL.
2. If A is regular and B is a CFL, then AB may not be context free.
3. If L is not regular, then L^* is not regular.
4. When reading a string of length n , an NFA with ϵ -transitions will be in at most $n + 1$ different states.
5. A DFA recognizing strings with 101 as a subsequence requires at least $2^3 = 8$ states.
6. For a regular language L , there exists an upper bound on the number of states required to represent L as a DFA.

3 Standalone NFAs and DFAs

Provide an NFA or DFA that decides each language. Consider why you chose to use one representation over the other.

1. $L = \{w \in \Sigma^* : 101 \text{ is a subsequence of } w\}$
2. $L = \{w \in \Sigma^* : 101 \text{ is a substring of } w\}$
3. $L = \{w \in \{0, 1, 2\}^* : \sum_i w_i \equiv 0 \pmod{5}\}$
4. $L = \{w \in \Sigma^* : w = x111 \text{ for some } x \in \Sigma^*\}$
5. $L_z = \{w \in \Sigma^* : z \text{ is a prefix of } w\}$

4 Product Construction

Let L_1, L_2, L_3 be regular languages. Provide an NFA or DFA that decides each language. Your final answer should use tuple notation, and not be a drawing.

1. $L_{eo} = \{w \in \Sigma^* : \#(1, w) \equiv \#(0, w) \equiv 1 \pmod{2}\}$
2. $L = \{w : w \in \overline{L_1 \cup L_2}\}$
3. $L = \{w : w \text{ is in at least two of } L_1, L_2, L_3\}$
4. $L = \{w : w \text{ has an odd number of 1s and contains the subsequence } 101\}$
5. $L_{ps} = \{w : w \text{ ends in } 111 \wedge \text{ the digits in } w \text{ sum to a number not divisible by } 5\}$

5 Language Transformations

Prove that each of the following languages are regular by creating an NFA, DFA, or regular expression for them.

1. Let $\Sigma = \{0, 1, 2\}$ and let $\text{replace}(w)$ replace every 1 in w with a 2 (ex $\text{replace}(120) = 220$). Define $\text{Two}(L) = \{\text{replace}(w) : w \in L\}$.
2. Using replace as above, define $\text{UnTwo}(L) = \{w \in \Sigma^* : \text{replace}(w) \in L\}$
3. Let $\text{prepend}(w)$ prepend 11 to w (ex $\text{prepend}(10) = 1110$). Define $\text{Prepend}(L) = \{\text{prepend}(w) : w \in L\}$.
4. Let $\text{add}(w)$ append 111 to w (ex $\text{add}(10) = 10111$). Define $\text{Add}(L) = \{\text{add}(w) : w \in L\}$.
5. Let $\text{d2}(w)$ delete the second-to-last character of w (ex $\text{d2}(1010) = 100$). Define $\text{D2}(L) = \{\text{d2}(w) : w \in L\}$.
6. Using d2 as above, define $\text{A2}(L) = \{w : \text{d2}(w) \in L\}$.
7. Let $\text{insert0}(w)$ be a function that generates all the ways to insert a single 0 into w at an arbitrary location (ex $\text{insert0}(10) = \{010, 100\}$). Define $\text{AddZero}(L) = \cup_{w \in L} \text{insert0}(w)$.
8. Using insert0 as above, define $\text{DeleteZero}(L) = \{w \in \Sigma^* : \text{insert0}(w) \cap L \neq \emptyset\}$

6 Fooling Sets

Find the largest fooling set you can for each of the given languages.

1. $L = \{0^n w 1^n : w \in \Sigma^* \wedge |w| \leq 3 \wedge n \geq 0\}$
2. $L = \{0^n w 1^n : w \in \Sigma^* \wedge n \geq 0\}$
3. $L = \{0^n 1^m : n \text{ is divisible by } m \wedge n, m \geq 0\}$
4. $L = \{z 0 z^r : z \in \Sigma^*\}$
5. (Hard.) $L_k = \{w : \text{the } k\text{th to last character of } w \text{ is } 1\}$

7 Context Free Languages

Write a context-free grammar for each of the provided languages.

1. $L = \{1^n 0^{3n} : n \geq 0\}$
2. $L = \{w \in \Sigma^* : w \text{ has exactly two more 1s than 0s}\}$
3. $L = \{w 0^n 1^n w^r : w \in \Sigma^* \wedge n \geq 0\}$

Describe the languages generated by each context-free grammar.

1.

$$S \rightarrow 00S1 \mid \epsilon$$

2.

$$\begin{aligned} S &\rightarrow 2S3 \mid 3S2 \mid R \\ R &\rightarrow 11R \mid R00 \mid \epsilon \end{aligned}$$

3.

$$S \rightarrow 0S1 \mid 1S0 \mid 1 \mid 0$$

8 Sums and Recurrences

Provide a tight asymptotic upper bound for each of the following sums and recurrences.

1. $\sum_{i=1}^n i\sqrt{i}$
2. $A(n) = 4A\left(\frac{n}{2}\right) + n^2, \quad n \geq 2 \quad \text{and} \quad A(1) = 1$
3. $T(n) = 2T\left(\frac{n}{2}\right) + n^2$
4. $T(n) = 4T\left(\frac{n}{4}\right) + cn$
5. $T(n) = \sqrt{n} \cdot T(\sqrt{n}) + 1 \text{ for } n \geq 2, T(2) = 1$

9 Graphs - Conceptual

1. Draw a small connected graph G and label a node s of G such that the following properties hold: (i) G has a cycle (ii) G does not have a cycle containing s and (iii) degree of s is at least 2
2. Let $G = (V, E)$ be an undirected graph and let $s \in V$. Describe an efficient algorithm that checks if there is a cycle in G that contains s .
3. Makefiles specify the dependencies of programs during compilation. How can a Makefile be modeled as a graph? How can one compute an ordering of programs so that the programs are compiled under the dependency constraints.
4. (CS374 SP23) Let $G = (V, E)$ be a directed graph. Call a vertex of G good if it is in some cycle. Describe a linear-time algorithm to count the number of good vertices in G .
5. Let $G = (V, E)$ be a directed graph where the length of each edge is either 2 or 3. Describe a linear-time algorithm to find the shortest paths from a given node s .

10 DP - Subsequences

1. Given arrays A and B , find the subsequences A' and B' such that $A' \cdot B'$ is maximized.
2. Find the longest common palindromic subsequence of arrays A , B , and C .
3. The *hamming distance* of two arrays A_1 , A_2 of size k is the number of bits that must be flipped to turn A_1 into A_2 . Given array A and array B , find the subsequence of A with the smallest hamming distance to B .
4. An element of array A at index i is *proper* if $A[i] \geq \sum_{j=1}^{i-1} A[j]$. Find the longest sequence of proper elements.
5. (Hard.) Find the subsequence of A with the largest geometric mean (product of all the elements divided by count), and output the geometric mean of that sequence.

11 DP - Subarrays

1. Compute the longest common subarray of A and B .
2. Given an array A , partition A into as few subarrays as possible such that the concatenation of digits in each subarray is prime. Assume you have access to an $O(1)$ black box for `IsPrime`.

12 DP - Optimization

1. At Willy Wonka's Chocolate Factory, Charlie—who is now remarkably jaded—wants to take as much chocolate as he can to resell. On his tour, Charlie will visit n rooms, each with chocolate worth $C[i] > 0$. Rooms that are risky to steal from have $R[i] = 1$, else $R[i] = 0$. Given Charlie can only steal from k risky rooms, what is the maximum amount of candy he can obtain?
2. (Hard.) Aaron wants to open k gyms in Champaign, which can be represented as a 1D line of n houses. Each gym will replace one house. Each house has a demand $D[i]$ for how much they want to use a gym. When a house has demand $D[i]$ and a gym is m houses away, the house's demand for that gym is $D[i]/m$. Find the maximum demand Aaron can obtain.

13 DP - Graphs

1. Sania is trying to infiltrate the Siebel Center for Computer Science to take all the free food she can. She starts off with a single key—Ryan's stolen iCard—which will get her inside the building. Once inside, the building is represented as a directed graph $G = (V, E)$. We can write $V = P \cup S$, where P are publicly accessible rooms, and S are staff accessible rooms. Some vertices may also contain a staff key left out that Sania can steal ($s(v) = 1$ if this is the case, else $s(v) = 0$) to gain access to staff accessible rooms. Each vertex offers $f(v)$ free food. What is the maximum amount of food Sania can steal?
2. An NFA (with no ϵ -transitions) can be represented as a (not necessarily acyclic) directed graph $G = (V, E)$, where edges are labeled with one or more symbols and represent transitions and vertices represent states. Let $\ell(e)$ denote the set of characters corresponding to the transition given by e . Given a start state s , string w , and array A of accepting states, does the NFA accept w ?

14 Divide and Conquer

1. In a sorted array of distinct positive integers, is there an index i such that $A[i] = i$?

2. Given array A , compute the 75th percentile value. Extend your algorithm to compute the p th percentile.
3. (Kinda hard.) Given an array A , two indices i and j are k -close if $|A[i] - A[j]| \leq k$. Report the number of k -close indicies.

15 Graphs - Transformations and Shortest Paths

1. Given a directed graph $G = (V, E)$, a vertex is lighted if $\ell(v) = 1$. A path is safe if it has no more than three consecutive unlit vertices. Given a vertex s and vertex t , what is the shortest lighted path from s to t ?
2. Given a vertex v in a graph with positive weights, what is the lightest cycle that includes v , if one exists?

16 Reductions - Concepts

For each statement below, select all claims that must be true if the statement is true regardless of the choice of the problem X .

1. $X \leq_p \text{SAT}$: $X \in P$, $X \in NP$, X is decidable, X is undecidable
2. $\text{SAT} \leq_p X$: $X \in P$, $X \in NP$, X is decidable, X is undecidable
3. $X \leq_p \text{ShortestPath}$: $X \in P$, $X \in NP$, X is decidable, X is undecidable
4. $X \implies \text{HALT}$: $X \in P$, $X \in NP$, X is decidable, X is undecidable
5. $\text{HALT} \implies X$: $X \in P$, $X \in NP$, X is decidable, X is undecidable

Determine whether each of the following claims are true or false.

1. If X is regular, then $X \in NP$
2. If X is decidable, then $X \in NP$
3. If X is undecidable, then $X \notin NP$
4. If X is in P , then $X \leq_p Y$ for *any* $Y \in P$
5. If X is NP -complete, then $Y \leq_p X$ for *any* $Y \in NP$

Briefly describe reductions to show each of the following statements.

1. $3\text{SAT} \implies \text{HALT}$
2. $\text{SAT} \leq_p \text{CircuitSAT}$
3. $\{\langle G, s, t \rangle : G \text{ has a path from } s \text{ to } t\} \leq_p \text{SAT}$

Select which of the following decision problems are NP -hard, and provide a short justification for your answer.

1. Given a graph G , is there a simple (no repeated vertices) path of length k ?
2. Given a graph G , is there a path from s to t ?
3. Given an acyclic circuit C and its inputs, what is the output?
4. Given an acyclic circuit C , are there some inputs so that it will output 1?

5. Given a graph with one node colored red, can the remainder of the graph be three-colored?

Select which of the following problems are decidable or undecidable, and provide a short justification for your answer.

1. $\{\langle M \rangle : 0 \in L(M)\}$
2. $\{\langle M \rangle : M \text{ has at most 374 states}\}$
3. $\{\langle M \rangle : |L(M)| = 374\}$
4. $\{\langle M \rangle : M \text{ halts on all inputs}\}$
5. $\{\langle M, w, t \rangle : M \text{ halts on } w \text{ in at most } t \text{ steps}\}$

Did you notice any pattern about the undecidability questions above?

17 Undecidability Reductions

Provide a proof that each of the following languages is undecidable (assume M is always a TM). Do not use Rice's theorem.

1. $L = \{\langle M \rangle : M \text{ accepts all even-length strings}\}$
2. $L = \{\langle M \rangle : |L(M)| \geq 1\}$
3. $L = \{\langle M, w \rangle : M \text{ accepts } w\}$
4. $L = \{\langle M, w, x \rangle : M \text{ accepts one of } w \text{ or } x, \text{ but not both}\}$
5. $L = \{\langle M \rangle : M \text{ runs for more than } n \text{ steps on some input of size } n\}$
6. $L = \{\langle M \rangle : L(M) \text{ is regular}\}$
7. $L = \{\langle M \rangle : L(M) \text{ is not regular}\}$

We've seen that questions about the languages of Turing machines are typically undecidable. This is not the case for NFAs and DFAs: most questions are decidable. Provide an algorithm to decide each of the following languages (assume M is always a DFA and N an NFA).

1. $L = \{\langle M, n \rangle : M \text{ accepts all binary strings of length } n\}$
2. $L = \{\langle M, w \rangle : M \text{ accepts } w\}$
3. $L = \{\langle M, w \rangle : M \text{ accepts some string ending in 012}\}$
4. $L = \{\langle N \rangle : \text{All strings accepted by } N \text{ begin with a 1}\}$
5. $L = \{\langle N, w \rangle : \text{All computations by } N \text{ on } w \text{ lead to acceptance}\}$

18 NP-Hardness Reductions

Prove that each problem is NP-Hard. Problems denoted (*) are more challenging than average.

1. A SAT formula is *mostly-satisfiable* if there is exactly one variable which, after being removed, results in the formula becoming satisfiable. Determine if a formula Φ is mostly-satisfiable.

2. (*) Xavier thinks microwaves are too complicated, so he's set out to design his own. He doesn't want to worry about starting a timer on the microwave, so he's devised the following strategy: each food's box will specify any number of variables v_1, \dots, v_n , as well as any number of linear constraints (i.e. $v_1 + v_2 \leq 2$ or $v_1 \geq v_n$) on a QR code the microwave will scan. Xavier's microwave will try to assign each v_i to either 0 or 1 while satisfying the provided constraints and maximizing $\sum v_i$, the cooking time (since it's better to overcook than undercook!). If the microwave can't figure out how to cook the food, it will just turn off.
3. A boolean circuit C is *kinda-ok* if, after switching exactly one or-gate to an and-gate (or vice versa), the circuit is satisfiable. Given a circuit C , is it kinda-ok?
4. The University of Illinois has decided to open its own zoo! The can choose from a list of n animals, and have access to a list of tuples (i, j) , which mean animal i would eat animal j if given the opportunity. Unfortunately, the only animals available to them have a tendency to escape, so they want to find the largest subset of animals that will not eat one another if they escape (called a *safe* subset).
5. A set of vertices is *mostly independent* if each vertex in the set is adjacent to at most one other vertex in the set. Compute the size of the largest mostly independent set of vertices.
6. A clique is *light* if it contains at most $n/2$ vertices. Find the size of the largest light clique in G .
7. Hans is planning his wedding and wants to seat guests with similar interests at the same table. His guests are eccentric, and each have exactly two interests. Compute the minimum number of tables Hans must rent, assuming each guest must sit down.
8. (*) A vertex cover C of size k is *balanced* if it can be partitioned into sets C_1 and C_2 so that $C_1 \cup C_2 = C$, $C_1 \cap C_2 = \emptyset$, $|C_1| = |C_2|$ and there are no edges between vertices in C_1 and C_2 . Compute the size of the smallest balanced vertex cover.
9. Given a graph G with red and white edges, *candy cane path* alternates between red and white edges. Compute the length of the longest candy cane path.
10. A *half-assed Hamiltonian cycle* visits at least $(1/100)$ th of the vertices of a graph G . Compute the length of the longest half-assed Hamiltonian cycle. (Hint: start with $1/2$).
11. Given a directed graph G , is there a cycle starting at node s which contains k vertices? (Hint: use a directed graph)
12. When compiling code, variables must be stored in registers and/or the stack. Two variables which are used at the same point in a program cannot both be stored in the same register. For example, if variables a , b , and c are used in the body of a loop, and there are two registers, one must be on the stack. Given an integer k , a list of variables, and a list of tuples (a, b) indicating that variables a and b are used at the same time, can the program be compiled to use only k registers and store no variables on the stack?
13. A graph is almost three-colorable if there is only one pair of neighboring vertices with the same color. Determine whether a graph G can be almost three-colored.
14. A graph is k -nearly three-colorable if there are at most k pairs of neighboring vertices with the same color. Determine whether a graph G can be k -nearly three-colored.