University of Nevada, Las Vegas Computer Science 456/656 Spring 2025

Answers to Assignment 8: Due Saturday May 3, 2025

1. Work Exercise 1 on Handout/reglrNC.pdf.

   Exercise 1: Let M be the NFA with transition matrix

	a	b
0	$0,1$	$\emptyset$
1	$\emptyset$	$1,2$
2	0	0

   .

   where 2 is the final state.

   For $w \in \{a,b\}^*$, $w \in L(M)$ if and only if $T_w[0][2] = 1$, since 0 is the start state and the only final state is 2. Use the tournament method shown above to determine whether $aabab \in L(M)$.

$$\underset{T_\lambda}{\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}} \quad \underset{T_a}{\begin{bmatrix}1&1&0\\0&0&0\\1&0&0\end{bmatrix}} \quad \underset{T_b}{\begin{bmatrix}0&0&0\\0&1&1\\1&0&0\end{bmatrix}}$$

$$\underset{T_a}{\begin{bmatrix}1&1&0\\0&0&0\\1&0&0\end{bmatrix}} \quad \underset{T_a}{\begin{bmatrix}1&1&0\\0&0&0\\1&0&0\end{bmatrix}} \quad \underset{T_b}{\begin{bmatrix}0&0&0\\0&1&1\\1&0&0\end{bmatrix}} \quad \underset{T_a}{\begin{bmatrix}1&1&0\\0&0&0\\1&0&0\end{bmatrix}} \quad \underset{T_b}{\begin{bmatrix}0&0&0\\0&1&1\\1&0&0\end{bmatrix}} \quad \underset{T_\lambda}{\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}} \quad \underset{T_\lambda}{\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}} \quad \underset{T_\lambda}{\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}}$$

$$\underset{T_{aa}}{\begin{bmatrix}1&1&0\\0&0&0\\1&1&0\end{bmatrix}} \quad \underset{T_{ba}}{\begin{bmatrix}0&0&0\\1&0&0\\1&1&0\end{bmatrix}} \quad \underset{T_b}{\begin{bmatrix}0&0&0\\0&1&1\\1&0&0\end{bmatrix}} \quad \underset{T_\lambda}{\begin{bmatrix}1&0&0\\0&1&9\\9&0&1\end{bmatrix}}$$

$$\underset{T_{aaba}}{\begin{bmatrix}1&0&0\\0&0&0\\1&0&0\end{bmatrix}} \quad \underset{T_b}{\begin{bmatrix}0&0&0\\0&1&1\\1&0&0\end{bmatrix}}$$

$$\underset{T_{aabab}}{\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}}$$

   Since $T_{aabab}[0][2] = 0$, We conclude that $abaab \notin L(M)$.

2. Work Exercies 1, 2, 3, 4 on Handout/indSS.pdf.

3. Give a definition of the class $\mathcal{P}$–SPACE.

   A language L is in the class $\mathcal{P}$–SPACE if there is a $\mathcal{P}$–SPACE algorithm which decides L.

4. Give a definition of the class $\mathcal{P}$–SPACE–complete. L is $\mathcal{P}$–SPACE complete if there is a $\mathcal{P}$–TIME reduction of any $\mathcal{P}$–SPACE language to L.

5. Let $L = \{a^n b^n : n \geq 0\}$. Prove that L is $\mathcal{NC}$. (Do not cite the fact that all CF languages are $\mathcal{NC}$.)

   Given a string $w \in \{a, b\}^*$:

   1. Assign a processor to each of the approximately $2n$ blocks of size a power of 2 used during the Tournament procedure. Each processor connects with its two parents and its one child. The resulting network has diameter $O(\log n)$.

   2. If any block contains symbol other than a or b, then $w \notin L$, and that fact is broadcast to all processors in $O(\log n)$ time.

   3. If any processor detects that its block contains the substring ba, $w \notin L$, and that fact is broadcast to all processors in $O(\log n)$ time.

   4. Assign the value 1 each a and 0 to each b. In $O(\log n)$ time, compute and broadcast the numbers of a's and b's in w. If the root processor detects that these numbers are different, $w \notin L$, and that fact is broadcast to all processors. Otherwise, the root processor will detect that the numbers are equal and thus $w \in L$, and that fact is broadcast, concluding the algorithm in logarithmic time.

Pseudopolynomial Algorithm for Subset Sum

We define an instance of *Subset_Sum*, the *subset sum problem* to be a string $\langle X \rangle \langle K \rangle$ where $X = x_1, x_2, \ldots x_n$. is a list of positive integers and K is an integer. A *solution* to that instance is a subsequence of X whose total is K. Without loss of generality, $x_i \leq K$ for each i, since otherwise x_i could not be part of the solution.

We have already proved that Subset_Sum is $\mathcal{NP}$ complete. We now give a dynamic programming algorithm $\mathcal{A}$ of time complexity $O(nK)$.

Definition of $\mathcal{A}$. Let $A[n+1][K+1]$ be the Boolean matrix where $A[i][k]$ means there is a subsequence of $x_1, \ldots x_i$ whose sum is k.

```
    For k from 1 to K
        A[0][k] = 0
A[0][0] = 1
For i from 1 to n
    For k from 0 to K
        A[i][k] = A[i-1][k]
        If k ≥ x_i and A[i][k − x_i]
            A[i][k] = 1
Return A[n][K]
```

Exercise 2: Use the algorithm $\mathcal{A}$ to decide whether there is a sublist of (6, 10, 7, 17, 3, 7, 10, 3, 4) whose sum is 25. (In a sequence, there can be duplicate terms.)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0
17	1	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	1	1	0
3	1	0	0	1	0	0	1	1	0	1	1	0	0	1	0	0	1	1	0	1	1	0	0	1	1	0
7	1	0	0	1	0	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	1	0	0	1	1	0
10	1	0	0	1	0	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1	1	0	0	1	1	0
3	1	0	0	1	0	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	1	0
4	1	0	0	1	1	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0

Since there is a 0 in the last row last column, there is no solution.

Exercise 3: Why can't we call $\mathcal{A}$ "polynomial"? The input size is $O(\sum_{i=1}^{n} \log_2 x_i + \log_2 K)$, and that value is not, in general, a polynomial function of nK.

Exercise 4: However, $\mathcal{A}$ can be made $\mathcal{P}$–TIME, if the terms of the sequence are restricted to numbers with at most 2 digits. Explain.

The number of bits of input is $O(\sum_{i=1}^{n} \log_2 x_i + \log_2 K)$. If K is greater than the sum of the terms of the sequence, then there is no solution, hence the answer can be given in $O(n)$ time. This means that $\log_2 K = O(n)$. and thus the input size is $O(n)$. Thus, we can assume that $K < 99n$. Thus $\log_2 K) = O(n)$, and hence the input size is $O(n)$. The time complexity of the algorithn is $O(nK) = O(nY_2)$,