CS208 Tutorial 3: Finite Automata Theory

1. Often, we are required to find ways to accept sets of strings with various properties. We've seen in class that DFAs and NFAs are ways to achieve this, with an equivalence between NFAs and DFAs. In this question and the next, we'll specify some properties of sets of strings and you will be required to construct (small) DFAs/NFAs to accept these sets (or languages). We aren't insisting that you should find the smallest (in terms of number of states) automata, but try to use as few states as you can.

   Let $\Sigma = \{0, 1, 2\}$. Construct DFAs for recognizing each of the following languages.

   1. $L_1 = \{w \in \Sigma^* \mid w$ doesn't have any pair of consecutive decreasing letters (numbers)$\}$. For example $010 \notin L$ but $00012 \in L$ and $222 \in L$.
   2. $L_2 = \{w \in \Sigma^* \mid w = u.v,\ u \in \Sigma^+,\ v \in \Sigma^*,\ \ n_0(u) > n_1(u) + 2$ or $n_1(u) > n_0(u) + 2\}$, where $n_i(u)$ denotes the count of i's in u, for $i \in \{0, 1\}$. For example, $0010221020012 \in L$ but $012012012 \notin L$.
   3. $L_3 = \{0^n \mid n > 0, n^3 + n^2 + n + 1 = 0 \mod 3\}$

2. Let $\Sigma = \{a, b\}$. Construct NFAs, possibly with ε-transitions for each of the following languages.

   1. $L_4 = \{w \in \Sigma^* \mid n_{ab}(w) \text{ is even}\}$, where $n_{ab}(w)$ denotes the count of times ab appears in w as consecutive letters.

   2. $L_5 = \{w \in \Sigma^* \mid w$ contains the "pattern" abba (as consecutive letters) followed by the "pattern" baba, possibly in an overlapping manner$\}$. For example, $abababbaabb, bababbabb \notin L$ but $abababbaba, ababbaabbaba \in L$.

   Now try constructing a DFA that recognizes L_5 using the subset construction and ε-edge removal on the NFA constructed above. Do you see an exponential blow-up in the count of states as you do this conversion?

3. **Take-away question: Propositional formulas and NFAs**

   For a CNF formula φ with m variables and c clauses, show that you can construct in polynomial time an NFA with $O(cm)$ states that accepts all **falsifying or non-satisfying assignments**, represented as boolean strings of length m. You can assume that the formula φ is over variables $x_1 x_2 \ldots x_m$, and you can assume that the NFA is fed as input the word $v_1 \cdot v_2 \cdot \cdots \cdot v_m$, where $v_i \in \{0, 1\}$ and v_i is interpreted as the value of propositional variables x_i, for all $i \in \{1, \ldots m\}$.

   Can you construct an NFA of size polynomial in c and m that accepts all **satisfying assignments** of the CNF formula φ?