Task 4 . NFA. Context Free Grammar.

Scott Weeden

Software Verification and Validation was undertaken in collaboration with Professor Namin. See the software-vv repository for related work.

Submit PDF file (and Latex if you use it) of your solution to Canvas by 11:59pm Fri Feb 27. .

1. You must use Latex.

2. (25 - basics) (NFA) Consider the NFA $M = (\Sigma, K, s, F, \Delta)$ where $\Sigma = \{a, b, c\}$, $K = \{p_1, p_2, p_3\}$, $s = p_1$, $F = \{p_3\}$ and Δ is a transition relation.

   (a) What is the alphabet of M?
   Solution. The alphabet of M is $\Sigma = \{a, b, c\}$.

   (b) What are the states of M?
   Solution. The states of M are $K = \{p_1, p_2, p_3\}$.

   (c) What is the start state of M?
   Solution. The start state is $s = p_1$.

   (d) Is p_2 a final state of M?
   Solution. No. The set of final states is $F = \{p_3\}$, so $p_2 \notin F$.

   (e) Give an example string over alphabet of M.
   Solution. For example, abc, a, bb, or ε (if we allow the empty string over Σ).

   (f) Give an example of a configuration of M with state p_1 in the configuration.
   Solution. A configuration is a pair (state, remaining input). One example is (p_1, abc) or (p_1, ε).

   (g) Give an example of Δ with at least 1 element and at most 3 elements (recall that Δ is simply a set).
   Solution. Δ is a subset of $K \times (\Sigma \cup \{\varepsilon\}) \times K$. For example: $\Delta = \{(p_1, a, p_2), (p_2, b, p_3)\}$, or $\Delta = \{(p_1, a, p_1)\}$ (one element).

3. (10 - basics) (Foundation) Study the *pumping lemma*.

   (a) List all concepts used in the lemma. Each concept should be in the format of *concept name* and *arguments* of the concept.
   Solution. (Typical formulation for regular languages.) Concepts: *language* (no arguments or language name), *regular* (language), *string* (e.g. s), *length*

(string), *positive integer* (e.g. pumping length p), *decomposition/concatenation* (string into xyz), *substring* (e.g. y), *repetition* (e.g. xy^iz for $i \geq 0$), *member* (string, language).

(b) List all logical symbols used in the lemma.
Solution. Typically: ∀ (for all), ∃ (there exists), ⇒ or → (implies), ∧ (and), ¬ (not), ∈ (member of), ≥, =. In the statement: "If L is regular then there exists $p \geq 1$ such that for every string $s \in L$ with $|s| \geq p$, there exist x, y, z such that $s = xyz$, $|xy| \leq p$, $|y| > 0$, and for all $i \geq 0$, $xy^iz \in L$."

4. (10 - medium) (Ideas underlying Pumping Lemma) Given an NFA M and a computation over it: $(p_1, abc) \vdash_M (p_2, bc) \vdash_M (p_2, c) \vdash_M (p_3, e)$ where p_1 is the start state and p_3 a final state. (Hint. Not only recall what *accept* intuitively means, but also its formal definition. Note also the repetition of state p_2 in the computation.)

   (a) is string *abc* accepted by M?
   Solution. Yes. The given computation shows $(p_1, abc) \vdash_M^* (p_3, \varepsilon)$ with $p_3 \in F$, so by definition abc is accepted by M.

   (b) is string *ac* accepted by M?
   Solution. We are not told whether there is a computation from (p_1, ac) to some (q, ε) with $q \in F$. So we cannot conclude from the given information alone; the answer depends on Δ. If there is a path $p_1 \xrightarrow{a} \cdots \xrightarrow{c} p_3$, then ac is accepted; otherwise not.

   (c) is string ab^2c accepted by M?
   Solution. Yes. We can extend the same idea: $(p_1, abbc) \vdash_M (p_2, bbc) \vdash_M (p_2, bc) \vdash_M (p_2, c) \vdash_M (p_3, \varepsilon)$ (reading one b in the middle step). So ab^2c is accepted.

   (d) Explain why for any non-negative number n, ab^nc is accepted by M?
   Solution. The computation shows that from p_1 we can read a and reach p_2, then stay in p_2 while reading any number of b's (including zero), then read c and reach $p_3 \in F$. So for any $n \geq 0$, we have $(p_1, ab^nc) \vdash_M^* (p_3, \varepsilon)$. By the formal definition of accept, ab^nc is accepted for every $n \geq 0$.

5. (5 - advanced) (Comprehensive). Using pumping lemma, prove the language $\{0^n1^n | n \geq 0\}$ over alphabet $\Sigma = \{0, 1\}$ is not regular.

   Solution. Assume for contradiction that $L = \{0^n1^n \mid n \geq 0\}$ is regular. Then the pumping lemma holds: there exists a pumping length $p \geq 1$ such that for every $s \in L$ with $|s| \geq p$, there exist x, y, z with $s = xyz$, $|xy| \leq p$, $|y| > 0$, and $xy^iz \in L$ for all $i \geq 0$. Choose $s = 0^p1^p$. Then $s \in L$ and $|s| = 2p \geq p$. So $s = xyz$ with $|xy| \leq p$, $|y| > 0$. Hence xy lies entirely within the block of 0's, so $y = 0^k$ for some $k \geq 1$. Then $xy^0z = xz = 0^{p-k}1^p$ has fewer 0's than 1's, so $xz \notin L$. This contradicts $xy^iz \in L$ for all $i \geq 0$. So L is not regular.

6. (35 - basics) (Context free grammar). Use the problem decomposition method to

(a) write a precise definition AND a context free grammar for the language of all strings whose number of 1's is one less than its number of 0's.

Solution. *Definition:* $L = \{w \in \{0,1\}^* \mid \#_0(w) - \#_1(w) = 1\}$ (over alphabet $\Sigma = \{0,1\}$). In words: every string in L has exactly one more 0 than 1.
Grammar (decomposition): Decompose into "one extra 0" and "equal 0s and 1s." Let E generate strings with $\#_0 = \#_1$. Then S adds exactly one 0: either before or after such a string.

$$S \to 0E \mid E0 \qquad E \to 0E1E \mid 1E0E \mid \varepsilon$$

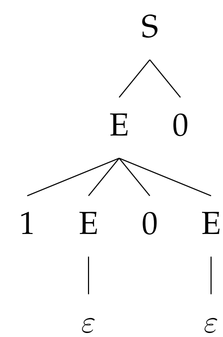
and based on the grammar,

(a) Write the right most derivation for string 100.
Solution. Rightmost derivation (always expand the rightmost nonterminal):

$$S \Rightarrow E0 \Rightarrow 1E0E0 \Rightarrow 1E00 \Rightarrow 100.$$

(First $S \to E0$; then rightmost nonterminal $E \to 1E0E$; then rightmost $E \to \varepsilon$ giving $1E0\varepsilon0 = 1E00$; then $E \to \varepsilon$ giving 100.)

(b) Draw a parse tree for 100.
Solution. Parse tree: root S with children E, 0. The subtree for E has children 1, E, 0, E (from $E \to 1E0E$); both child E's derive ε. So the tree yields $1 \cdot \varepsilon \cdot 0 \cdot \varepsilon \cdot 0 = 100$.



7. (5 - advanced) (Context free grammar). Following the method discussed in class, write context free grammar for the language (over alphabet $\{0,1\}$) consisting of strings that have equal number 0's and 1's. If you follow ideas and methods other than the ones discussed in class, you will get at most 90% for this quesiton.

   Solution. $L = \{w \in \{0,1\}^* \mid \#_0(w) = \#_1(w)\}$. Grammar (start symbol E):

$$E \to 0E1E \mid 1E0E \mid \varepsilon$$

   Each rule preserves the balance of 0s and 1s; ε gives the base case.

8. (10 - medium) (Foundation) Give a precise definition of *context free language*.

   Solution. A language L over an alphabet Σ is a *context-free language* (CFL) if there exists a context-free grammar $G = (V, \Sigma, P, S)$ such that the language generated by G equals L, i.e. $L(G) = L$. Equivalently: L is a CFL iff $L = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$ for some CFG G with start symbol S.