CS 121, Section 2

Week of September 16, 2013

1 Concept Review

1.1 Overview

In the past weeks, we have examined the *finite automaton*, a simple computational model with limited memory. We proved that DFAs, NFAs, and regular expressions are equal in computing power and recognize the *regular languages*. We also showed that the regular langages are closed under *union, concatenation, Kleene Star, intersection, difference, complement, and reversal.* We then used a counting argument to show that there are indeed languages which are non-regular.

This week in section we will become a little more comfortable with these topics by working with regular expressions, making arguments about countability, and exploring some more closure properties of regular languages.

1.2 Cardinalities

We classify the cardinality of a set S as follows.

- Finite, if there is a bijection between S and $\{1, 2, \cdots, n\}$ for some $n \geq 0$.
- Coutably infinite, if there is a bijection between S and $\mathbb{N}$.
- Countable, if it is finite or countably infinite.
- Uncountable, otherwise.

Examples include the following.

- Finite: Σ (alphabet), states of a DFA, students in CS121, finite unions of finite sets.
- Countably infinite: Σ^* (strings), $\mathbb{Z}$, DFAs, countable unions of countably infinite sets.
- Uncountable: $\mathcal{P}(\mathbb{N})$, set of all languages.

Since there are only countably many regular languages and uncountably many languages, 'most' languages are non-regular.

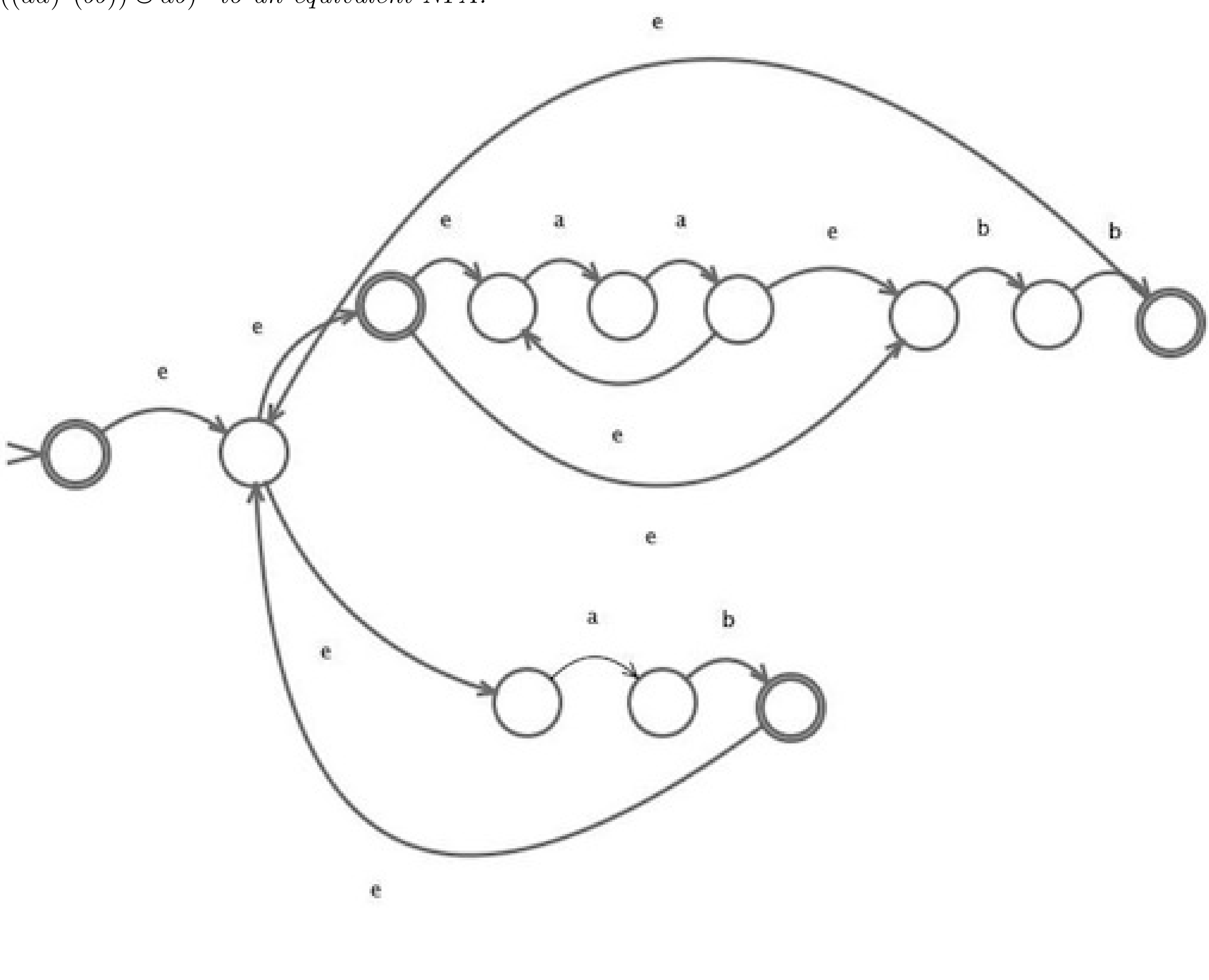
2 Exercises

Exercise 2.1. *Describe in plain English the language represented by the following regular expressions.*

1. $a^* \cup b^*$
2. $(aaa)^*$
3. $\Sigma^* a \Sigma^* b \Sigma^* a \Sigma^*$

1. Strings that do not contain both a's and b's.
2. Strings of a's that have length equal to some multiple of 3.
3. Strings that contain the subsequence aba.

Exercise 2.2. *Using the procedure outlined in class, convert the regular expression* $(((aa)^*(bb)) \cup ab)^*$ *to an equivalent NFA.*



Exercise 2.3. *Let L be a language over the alphabet $\Sigma = \{a, b\}$. Define* $\mathsf{PigLatin}(L) = \{w\sigma : \sigma \in \Sigma, w \in \Sigma^*, \sigma w \in L\}$. *Informally,* $\mathsf{PigLatin}(L)$ *is the language containing all strings in L except that each string has had its first character moved to its end. (For example,* $\mathsf{PigLatin}(\{abc, a, aab\}) = \{bca, a, aba\}$*.)*

Show that if L is regular, then $\mathsf{PigLatin}(L)$ *is regular. Specifically, given a DFA for L, show how to construct an NFA for* $\mathsf{PigLatin}(L)$*. (Your proof for this problem should involve finite automata and not regular expressions.)*

Let $D = (Q, \Sigma, \delta, s, F)$ be the DFA accepting L. We will construct an NFA $N = (Q', \Sigma, \delta', s', F')$ accepting $\mathsf{PigLatin}(L)$.

Formally,

$$
\begin{aligned}
Q' &= \{s', f'\} \cup (Q \times \{a, b\}) \\
\delta'(q, \sigma) &= \begin{cases}
\{(\delta(q_0, a), a), (\delta(q_0, b), b)\} & \text{if } q = s' \text{ and } \sigma = \varepsilon, \\
\{f', (\delta(p, a), a)\} & \text{if } q = (p, a) \text{ for some } p \in F \text{ and } \sigma = a, \\
\{f', (\delta(p, b), b)\} & \text{if } q = (p, b) \text{ for some } p \in F \text{ and } \sigma = b, \\
\{(\delta(p, \sigma), a)\} & \text{if } q = (p, a) \text{ for some } p \notin F \text{ and } \sigma \neq \varepsilon, \text{ or} \\
 & \text{if } q = (p, a) \text{ for some } p \in F \text{ and } \sigma = b, \\
\{(\delta(p, \sigma), b)\} & \text{if } q = (p, b) \text{ for some } p \notin F \text{ and } \sigma \neq \varepsilon, \text{ or} \\
 & \text{if } q = (p, b) \text{ for some } p \in F \text{ and } \sigma = a, \\
\emptyset & \text{otherwise.}
\end{cases} \\
F' &= \{f'\}
\end{aligned}
$$

Informally, our NFA consists of 2 copies of the original DFA, one for a and one for b. We represent the states of the a-copy with $Q \times \{a\}$, respectively for b. There will be a new start state s' which will have ε-transitions to two states: to $(\delta(q_0, a), a)$ and to $(\delta(q_0, b), b)$– that is, to the state in the a-copy that the DFA would be in after reading an a as the first character, and to the state in the b-copy it would be in after reading b as the first character. The transitions within each copy then mimic that of the DFA. Finally, we have a transition on an a (resp. b) from any old final state in the a-copy (resp. b) to the new final state f'.

Informal justification: Given that σw is accepted by D, we accept $w\sigma$ by non-deterministically jumping to either the state it might be in after reading σ. We then process all of w, and then accept the string only if w puts the NFA in an original final state, and then the appropriate σ is the last character left. If the string is in $\mathsf{PigLatin}(L)$, then there is an appropriate place for it to jump. Conversely, if the NFA accepts a string, then there must have been some place it jumped to, and so for it to get to the final state it must have looked like $w\sigma$ for some $\sigma w \in L$.

Exercise 2.4. *Prove or disprove the following statements about regular expressions:*

1. $L((RS \cup R)^*R) = L(R(SR \cup R)^*)$.
2. $L((RS \cup R)^*RS) = L((RR^*S)^*)$.

1. $L((RS \cup R)^*R) = L(R(SR \cup R)^*)$. True. Suppose $w \in (RS \cup R)^*R$, then $w = r_1s_1r_2s_2 \ldots r_ns_nr_{n+1}$ where $r_1, \ldots, r_{n+1} \in R$ and $s_1, \ldots s_n \in S \cup \{\varepsilon\}$. Thus $w \in R(SR \cup R)^*$. The other direction is similar.

2. $L((RS \cup R)^*RS) = L((RR^*S)^*)$. True. Suppose $w \in (RS \cup S)^*RS$, then $w = r_1s_1r_2s_2 \ldots r_ns_nr_{n+1}s_{n+1}$ where $r_1, \ldots, r_{n+1} \in R$, $s_1, \ldots, s_n \in S \cup \{\varepsilon\}$ and $s_{n+1} \in S$. Let k be the smallest number such that $s_k \in S$; k exists because $s_{n+1} \in S$. Then $r_1s_1 \ldots r_ks_k \in RR^*S$. Now repeating for the next smallest k, and we can show by induction that $w \in (RR^*S)^*$. The other direction is similar.

Exercise 2.5. *Are the following sets finite (if so, how large), countably infinite, or uncountably infinite? Justify your answer.*

1. *The set of all infinite binary sequences* $\{0,1\}^{\mathbb{N}}$
2. *The set of real numbers* $\mathbb{R}$.
3. *The set of rational numbers* $\mathbb{Q}$.
4. *The set of all English words.*
5. *The set of all English sentences.*

1. *Uncountable.* Define $f : \mathcal{P}(\mathbb{N}) \to \{0,1\}^{\mathbb{N}}$ by

$$f(S)_i = 1 \iff i \in S.$$

   We claim that f is one-to-one. Since $\mathcal{P}(\mathbb{N})$ is uncountable, the result follows. Let S and S' be distinct subsets of $\mathbb{N}$. Then there exists i such that $i \in S$ and $i \notin S'$ or vice versa. Either way, $f(S)_i \neq f(S')_i$. So $f(S) \neq f(S')$.

2. *Uncountable.* Define $f : \{0,1\}^{\mathbb{N}} \to \mathbb{R}$ by

$$f(x) = \sum_i x_i 3^{-i}.$$

   We claim that f is one-to-one. (Thus f is a bijection with a subset of $\mathbb{R}$.) Since $\{0,1\}^{\mathbb{N}}$ is uncountable, this implies that $\mathbb{R}$ is uncountable.

Suppose x and x' are two *distinct* sequences in $\{0, 1\}$. We must show that $f(x) \neq f(x')$. Let n be the smallest index different between x and x'. That is, $x_n \neq x'_n$, but $x_i = x'_i$ for $i < n$. Without loss of generality $x_n = 1$ and $x'_n = 0$. Now

$$\begin{aligned} f(x) - f(x') = &\sum_i (x_i - x'_i)3^{-i} \\ =&3^{-n} + \sum_{i>n}(x_i - x'_i)3^{-i} \\ \geq&3^{-n} - \sum_{i>n} 3^{-i} \\ =&3^{-n} - \frac{3^{-n-1}}{1 - 3^{-1}} \\ =&\frac{3^{-n}}{2} \\ >&0, \end{aligned}$$

as required.

3. *Countably Infinite.* A finite language can be written down. That is, you can just concatenate all the words in the language into one string (with a special separator symbol \$). So there is a one-to-one map from finite languages over $\{a, b\}$ to strings over $\{a, b, \$\}$. There are only countably many strings, so there are countably many finite languages. As finite languages can be arbitrarily large, there are infinitely many.

4. *Finite.* The Oxford English Dictionary (supposedly) lists all english words. There are about 300,000 in total.

5. *Countably Infinite.* This is open to debate. While arbitrarily long sentences may not make much sense, rather than drawing a line in the sand, we accept the possibility of arbitrarily long sentences. Thus there are infinitely many valid English sentences. Since this is a subset of Σ^*, where Σ is the English alphabet plus space and punctuation, it is countable.