CS 3133: Homework 1

Adam Camilli

September 4, 2017

1. **2.23** (page 60) Give a regular expression that represents the described set: The set of strings over $\{a, b, c\}$ that begin with a, contain exactly two b's, and end with cc.

$$a(a, c)^{\star}b(a, c)^{\star}b(a, c)^{\star}cc$$

2. **2.26** (page 60) Give a regular expression that represents the described set: The set of strings over $\{a, b\}$ in which the number of a's is divisible by three.

$$(b^{\star}ab^{\star}ab^{\star}ab^{\star})^{\star}$$

3. **2.39d** (page 61) Use the regular expressions in Table 2.1 to establish the following identity: $(a \cup b)^{\star} = (a^{\star} \cup ba^{\star})^{\star}$:

To get desired result, we can use identity 12. from Table 2.1, defined as

$$\begin{aligned}
(a \cup b)^{\star} &= (a^{\star} \cup b)^{\star} \\
&= a^{\star}(a \cup b)^{\star} \\
&= (a \cup ba^{\star})^{\star} \\
&= (a^{\star}b^{\star})^{\star} \\
&= a^{\star}(ba^{\star})^{\star} \\
&= (a^{\star}b)^{\star}a^{\star}
\end{aligned}$$

Take the intermediate result $(a \cup ba^{\star})^{\star}$, and, using $u = a$ and $v = ba^{\star}$, reapply the first parts of identity 12. to obtain desired identity:

$$\begin{aligned}
(u \cup v)^{\star} &= (u^{\star} \cup v)^{\star} && \text{(Step 1 of Identity 12.)} \\
&= (a^{\star} \cup ba^{\star})^{\star} && \text{(Using } u = a \text{ and } v = ba^{\star}\text{)}
\end{aligned}$$

4. Let Σ be an alphabet, and u, v, w regular expressions over Σ. Are the following regular expression identities true?

   (a) $u \cup (vw) = (u \cup v)(u \cup w)$

   No. The lefthand expression contains v and w only in the combined form (vw), while the righthand expression allows singular v and w. Here is one of many counterexamples that can therefore be formed:

   Let alphabet $\Sigma = \{a, b, c\}$, and regular expressions $u = a, v = b, w = c$:

$$a \cup (bc) \not\supset ac$$

$$(a \cup b)(a \cup c) \supset ac$$

   Therefore, $u \cup (vw) \neq (u \cup v)(u \cup w)$.

   (b) $u^\star(v \cup w) = u^\star v \cup u^\star w$

   Yes. This is merely an application of the distributive regular expression identity

$$u \cup (v \cup w) = uv \cup uw$$

   with $u = u^\star$, which, since u, v, and w can represent any regular expression, must be valid.

5. **1.44** (page 39) Give a recursive definition of the set of ancestors of a node x in a tree.

   Basis: The parent node p of node x is an ancestor of x: $p \in A(x)$

   Recursive Step: An ancestor of p is an ancestor of x: $A(p) \in A(x)$

   Closure: A node a is an ancestor of x if it is obtainable from x with finitely many applications of the recursive step.

6. **3.1** (page 97) Let G be the grammar

$$\begin{aligned} S &\to abSc \mid A \\ A &\to cAd \mid cd. \end{aligned}$$

(a) Give a derivation of $ababccddcc$.

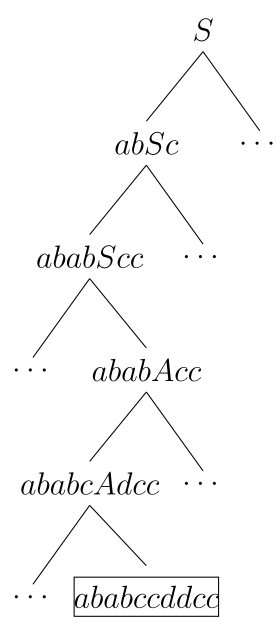
Defining the productions as

1. $S \to abSc$
2. $S \to A$
3. $A \to cAd$
4. $A \to cd$

we can derive it like so:

$$\begin{aligned} S &\Rightarrow abSc && (1.) \\ &\Rightarrow ababScc && (1.) \\ &\Rightarrow ababAcc && (2.) \\ &\Rightarrow ababcAdcc && (3.) \\ &\Rightarrow ababccddcc && (4.) \end{aligned}$$

(b) Build the derivation tree for the derivation in part (a).



(c) Use set notation to define L(G).

$$L(G) = \left\{ \{ab\}^n \{c\}^m cd \{d\}^m \{c\}^n \mid n \geq 1, m \geq 0 \right\}$$

Note that n and m, respectively, are essentially the number of times that recursive productions $S \to abSc$ and $A \to cAd$ are applied.