Discrete Structures *Prelim 2 sample questions*— ***Solutions***
CS2800 *Questions selected for spring 2017*

1 Structural induction, finite automata, regular expressions

1. *We define a set S of functions from $\mathbb{Z}$ to $\mathbb{Z}$ inductively as follows:*

Rule 1. *For any $n \in \mathbb{Z}$, the translation (or offset) function $t_n : x \mapsto x + n$ is in S.*

Rule 2. *For any $k \neq 0 \in \mathbb{Z}$, the scaling function $r_k : x \mapsto kx$ is in S.*

Rule 3. *If f and g are elements of S, then the composition $f \circ g \in S$.*

Rule 4. *If $f \in S$ and f has a right inverse g, then g is also in S.*

In other words, S consists of functions that translate and scale integers, and compositions and right inverses thereof.

Note: *This semester, we made a bigger distinction between the elements of an inductively defined set and the meaning of an inductively defined set. We probably would have phrased this question as follows: Let S be given by*

$$s \in S ::= t_n \mid r_k \mid s_1 \circ s_2 \mid rinv\ s$$

and inductively, let the function defined by s (written $F_s : \mathbb{Z}$) be given by the rules $F_{t_n}(x) ::= x + n$, $F_{r_k}(x) ::= ks$, $F_{s_1 \circ s_2}(x) ::= F_{s_1} \circ F_{s_2}$ and let $F_{rinv\ s} ::= g$ where g is a right inverse of F_s.

(a) *[1 point] Show that the function $f : x \mapsto 3x + 17$ is in S.*

Solution By rule 1, the function $t_{17} : x \mapsto x + 17$ is in S, and by rule 2, $r_3 : x \mapsto 3x$ is in S. By rule 3, therefore, $t_{17} \circ r_3 : x \mapsto 3x + 17$ is in S.

(b) *Use structural induction to prove that for all $f \in S$, f is injective. You may use without proof the fact that the composition of injective functions is injective.*

Solution We must show that all functions formed with each of the rules are injective. Let $P(s)$ be the statement s is injective.

$P(t_k)$ holds, because t_k has a two sided inverse t_{-k}, and is therefore injective.

$P(r_k)$ holds, because we required that $k \neq 0$. Therefore, if $kx_1 = kx_2$, we can cancel k to find $x_1 = x_2$.

$P(f \circ g)$ holds, assuming $P(f)$ and $P(g)$, because the composition of injections is an injection.

If g is the right inverse of f, then $P(g)$ holds, because g has a left-inverse (namely f) and is therefore injective.

(c) *Give a surjection ϕ from S to $\mathbb{Z}$ (proof of surjectivity not necessary). Remember that this surjection must map a* function *to an* integer*, and for every integer there must be a function that maps to it.*

Solution Let $\phi(s) ::= s(0)$. This is a surjection, because $t_n(0) = 0 + n = n$, so for any n there exists $s \in S$ (namely t_n) with $\phi(s) = n$.

2. *Seeking to avoid the limitations of regular expressions, a student designs a set S of "super expressions". Just as with regular expressions, each super expression $s \in S$ matches a set of strings, denoted by $L(s)$. The set S and the function L are defined inductively as follows:*

   - *A "character class" is denoted $[a_1 a_2 \dots a_n]$ (where the a_i are single characters, i.e. elements of Σ). It matches a single character which is any of the a_i. For example, $[012]$ matches "0", "1", and "2", but does not match "12".*

     Formally, $[a_1 a_2 \dots a_n] \in S$ and $L([a_1 a_2 \dots a_n]) = \{a_1, a_2, \dots, a_n\}$.

   - *If $s \in S$, a "positive repetition" of s is denoted $s+$. It matches one or more strings, each of which is matched by s. For example, $[45]+$ matches "4", "54", and "555" but does not match ϵ.*

     Formally, $s+ \in S$ and $L(s+) = \{x_1 x_2 \cdots x_n \mid n \geq 1 \text{ and } x_i \in L(s)\}$.

   - *If $s_1 \in S$ and $s_2 \in S$ then the "difference" of s_1 and s_2 is denoted by $s_1 \setminus s_2$. It matches all strings that s_1 matches but s_2 does not. For example, $([123]+) \setminus [123]$ matches "1322" and "11" but not "1", "2", or "3".*

     Formally, $s_1 \setminus s_2 \in S$ and $L(s_1 \setminus s_2) = L(s_1) \setminus L(s_2)$.

   - *If $s_1 \in S$ and $s_2 \in S$ then the "concatenation" of s_1 and s_2 is denoted by $s_1 s_2$. It matches any string that is formed by concatenating a string matched by s_1 with a string matched by s_2. For example $[012][34]$ matches "03" and "24" but not "0" or "32".*

     Formally, $s_1 s_2 \in S$ and $L(s_1 s_2) = \{xy \mid x \in L(s_1) \text{ and } y \in L(s_2)\}$.

   (a) *Let $\Sigma = \{0, 1, 2\}$. Let s be the super expression "$([01][12])+ \setminus ([012][012])$". Write a (normal) regular expression r such that $L(r) = L(s)$.*

   Solution To explain the answer, I'll refer to a string matching $[01][12]$ as a unit. We must match strings containing two or more units. A unit is matched by the (normal) regular expression $(0 + 1)(1 + 2)$. Therefore, two or more units is matched by $(0+1)(1+2)(0+1)(1+2)((0+1)(1+2))^*$.

   (b) *Prove by structural induction that every super expression $s \in S$ has an equivalent regular expression. You may use without proof the fact that the union, intersection, and complement of regular languages are regular.*

   Solution Let $P(s)$ be the statement "there exists $r \in RE$ such that $L(r) = L(s)$."

   We must show $P([a_1 \dots a_n])$, $P(s^+)$ (assuming $P(s)$), $P(s_1 \setminus s_2)$ (assuming $P(s_1)$ and $P(s_2)$), and $P(s_1 s_2)$ (again, assuming $P(s_1)$ ad $P(s_2)$)

   $P([a_1 \dots a_n])$ holds, because $L([a_1 \dots a_n]) = L(a_1 + a_2 + \cdots + a_n)$.

   $P(s_1 \setminus s_2)$ holds; we assume inductively that s_1 and s_2 are regular. We know that $L(s_1 \setminus s_2) = L(s_1) \setminus L(s_2) = L(s_1) \cap \overline{L(s_2)}$, which is regular, because the intersection and complement of regular languages is regular.

   Finally, we must show that $L(s_1 s_2)$ is regular. We inductively assume that s_1 and s_2 are regular. Therefore, there exist r_1 and r_2 with $L(r_1) = L(s_1)$ and $L(r_2) = L(s_2)$. By definition, this shows that $L(r_1 r_2) = L(s_1 s_2)$, so $s_1 s_2$ is regular.

3. *Given DFAs $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$, we can construct a machine M_{12} with $L(M_{12}) = L(M_1) \cap L(M_2)$ as follows:*

   - *Let $Q = Q_1 \times Q_2 =$ the set of all ordered pairs (q_1, q_2), where $q_1 \in Q_1$ and $q_2 \in Q_2$.*
   - *Let $q_0 \in Q = (q_{01}, q_{02})$.*

- *Let* $F = F_1 \times F_2 = \{(q_1, q_2) \mid q_1 \in F_1 \text{ and } q_2 \in F_2\}$.
- *Let* $\delta_{12}((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$.
- *Let* $M_{12} = (Q, \Sigma, \delta_{12}, q_0, F)$.

Use structural induction to prove that for all $x \in \Sigma^*$, $\widehat{\delta}_{12}((q_1, q_2), x) = \left(\widehat{\delta}_1(q_1, x), \widehat{\delta}_2(q_2, x)\right)$.

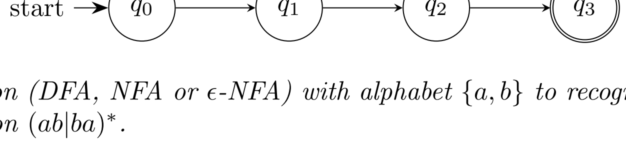
Solution The main challenge here is wading through the notational jungle to understand what the problem actually says. Once you've done this, the proof is short and straightforward. Here it is in all its glory:

> We will prove the result by structural induction on x, as suggested. Both the set of strings Σ^* and the extended transition function $\widehat{\delta}$ are defined recursively (see the definitions at the end). The base case for x is the empty string ϵ. We can simply read off the corresponding line in the definition of $\widehat{\delta}$, which tells us that
>
> - $\widehat{\delta}_1(q_1, \epsilon) = q_1$,
> - $\widehat{\delta}_2(q_2, \epsilon) = q_2$, and
> - $\widehat{\delta}_{12}((q_1, q_2), \epsilon) = (q_1, q_2)$.
>
> Hence $\widehat{\delta}_{12}((q_1, q_2), \epsilon) = (q_1, q_2) = \left(\widehat{\delta}_1(q_1, \epsilon), \widehat{\delta}_2(q_2, \epsilon)\right)$, so the statement is true in the base case.
>
> Now assume the statement is true for some string x, and consider the "next larger" string xa.
>
> Again reading off the appropriate line in the definition of $\widehat{\delta}$, we know that
>
> - $\widehat{\delta}_1(q_1, xa) = \delta_1(\widehat{\delta}_1(q_1, x), a)$, and
> - $\widehat{\delta}_2(q_2, xa) = \delta_2(\widehat{\delta}_2(q_2, x), a)$
>
> What is $\widehat{\delta}_{12}((q_1, q_2), xa)$? Well, we also have
>
> $$\begin{aligned}\widehat{\delta}_{12}((q_1, q_2), xa) &= \delta_{12}(\widehat{\delta}_{12}((q_1, q_2), x), a) && \text{(definition of } \widehat{\delta}_{12}\text{)}\\ &= \delta_{12}\left(\left(\widehat{\delta}_1(q_1, x), \widehat{\delta}_2(q_2, x)\right), a\right) && \text{(inductive hypothesis)}\\ &= \left(\delta_1\left(\widehat{\delta}_1(q_1, x), a\right), \delta_2\left(\widehat{\delta}_2(q_2, x), a\right)\right) && \text{(definition of } \delta_{12}\text{)}\\ &= \left(\widehat{\delta}_1(q_1, xa), \widehat{\delta}_2(q_2, xa)\right) && \text{(definition of } \widehat{\delta}_1, \widehat{\delta}_2\text{)}\end{aligned}$$
>
> This proves the statement for all strings $x \in \Sigma^*$ by (structural) induction.

4. (a) *Draw a finite automaton (DFA, NFA or* ϵ*-NFA) with alphabet* $\{0, 1\}$ *to recognize the language*

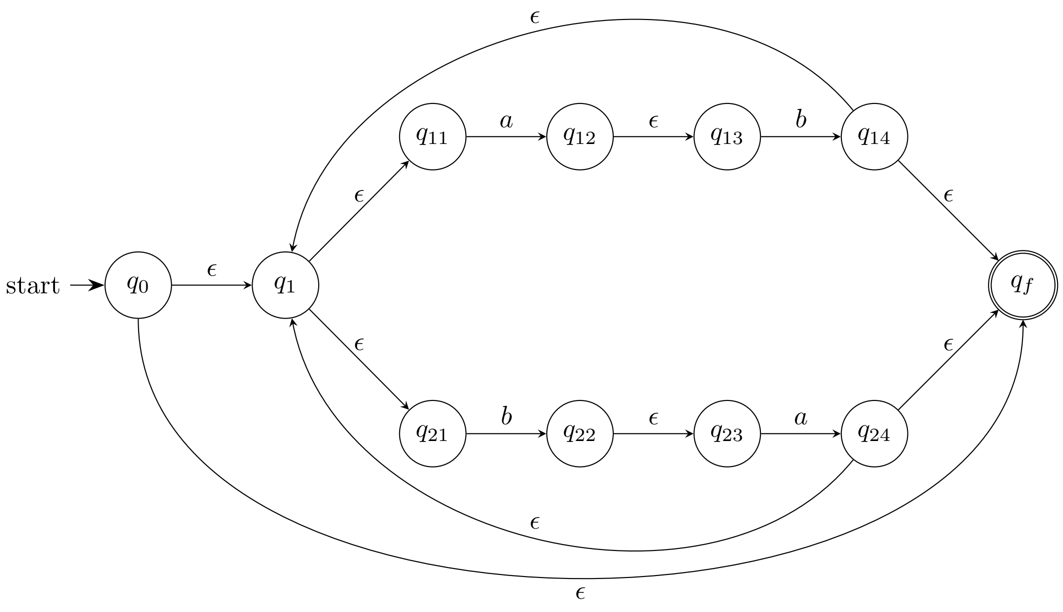
$$\{x \in \{0, 1\}^* \mid x \textit{ contains the substring } 010\}$$

Solution An NFA is probably the easiest to construct.

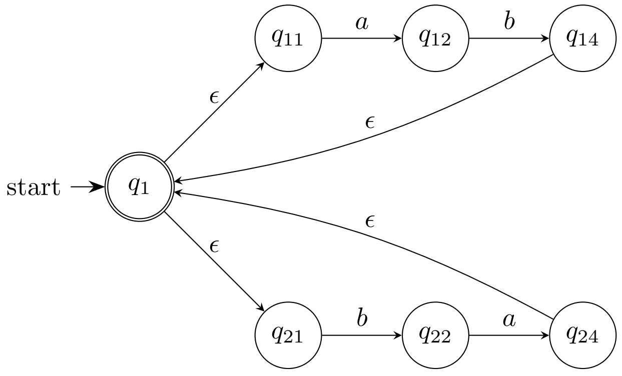


(b) *Draw a finite automaton (DFA, NFA or* ϵ*-NFA) with alphabet* $\{a, b\}$ *to recognize the same language as the regular expression* $(ab|ba)^*$.

Solution We can blindly apply the regex $\to$ ϵ-NFA construction from the proof of Kleene's Theorem to get something like this:



Of course this is a bit messy, a more compressed version might look like this:



5. *Prove that $L = \{0^n 1 0^n \mid n \in \mathbb{N}\}$ is not regular.*

Solution Suppose that this language is accepted by some deterministic finite automaton with N states. Consider the string $x = 0^n 1 0^n$. Since x is in the language and $|x| \geq N$, by the Pumping Lemma, there exist strings u, v, and w such that $x = uvw$, $|v| \geq 1$, $|uv| \leq N$, and M accepts $uv^i w$ for all $i > 0$. Since $|uv| \leq N$, it must be the case that uv is a string of 0's, and that w contains the 1 in $0^N 1 0^N$. Thus, if $i > 1$, $uv^i w$ has more than N 0s to the left of the 1 and only N 0s to the right of the 1, and thus is not in the language. This contradicts the assumption that the language is accepted by M (since M accepts a string not in the language).

6. *Let r be a regular expression. Show that there exists a regular expression r' with $L(r') = \overline{L(r)}$ (the complement of $L(r)$). If your proof involves the construction of a regular expression or automaton, you must prove that the language of the regular expression/automaton is what you claim it is (using the*

definitions).

Solution First, note that $L(r)$ is a language, which means that it's a set of strings. So its complement consists of all the strings in $L(r)$ that are not in Σ^*. (Many of you took "complement" to mean the result of switching the 0s and 1s in the strings in $L(r)$. This is not what complement means in this context; moreover, this approach fails if $\Sigma \neq \{0, 1\}$. Unfortunately, you typically got 0, 1, or 2 out of 6 if you did this, depending on what else you did.)

By Kleene's theorem, there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ with $L(M) = L(R)$. We can form a new automaton M' by flipping the accept and reject states of M: $M' = (Q, \Sigma, \delta, q_0, Q \setminus F)$.

I claim $L(M') = \overline{L(M)}$. Indeed,

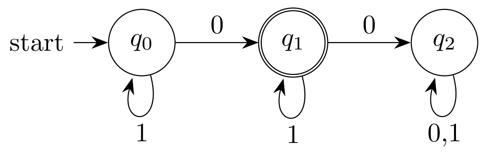
$$\begin{aligned} x \in L(M') &\iff \hat{\delta}(q_0, x) \in Q \setminus F \\ &\iff \hat{\delta}(q_0, x) \notin F \\ &\iff x \notin L(M) \end{aligned}$$

(You typically lost 1 point if you didn't give a proof.)

Applying Kleene's theorem again tells us that since there is a DFA with $L(M') = \overline{L(r)}$ there must be a regular expression r' with $L(r') = \overline{L(r)}$.

7. *Build a deterministic finite automaton that recognizes the set of strings of 0's and 1's, that only contain a single 0 (and any number of 1's). Describe the set of strings that lead to each state.*

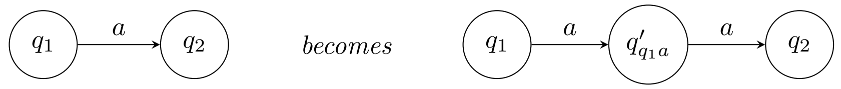
Solution



The strings leading to q_i contain i 0's.

8. *Given a string x, we can define the "character doubling" of x to be x with every character doubled: for example $cd(abc) = aabbcc$. Formally, $cd(\epsilon) = \epsilon$, and $cd(xa) = cd(x)aa$. We can then define the "character doubling" of a language L to be the set of all strings formed by doubling the characters of strings in L; formally $cd(L) = \{cd(x) \mid x \in L\}$.*

   Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$, we can construct a new DFA M_{cd} that recognizes $cd(L(M))$ by adding a new state q'_{qa} to the middle of every transition from q on character a:



   (a) *Formally describe the components $(Q_{cd}, \Sigma_{cd}, \delta_{cd}, q_{0cd}, F_{cd})$ of M_{cd} in terms of the components of M. Be sure to describe δ_{cd} on all inputs (you may need to add one or more additional states).*

   Solution $Q_{cd} = Q \cup \{q'_{qa} \mid q \in Q, a \in A\} \cup \{X\}$

   $\delta_{cd} : (q, a) \mapsto q'_{qa}$; $\delta_{cd} : (q'_{qa}, a) \mapsto \delta(q, a)$; $\delta_{cd} : (q'_{qa}, b) \mapsto X$ if $a \neq b$, and $\delta_{cd} : (X, a) \mapsto X$.

   The remaining components are unchanged: $\Sigma_{cd} = \Sigma$, $q_{0cd} = q_0$, and $F_{cd} = F$.

   (b) *Use structural induction on x to prove that for all x, $\widehat{\delta}(q_0, x) = \widehat{\delta}_{cd}(q_{0cd}, cd(x))$.*

Solution Let $P(x)$ be the statement that $\widehat{\delta}(q_0, x) = \widehat{\delta}_{cd}(q_0, cd(x))$. I will prove $\forall x, P(x)$ by structural induction.

To show $P(\epsilon)$, note that $\widehat{\delta}(q_0, \epsilon) = q_0$. Moreover, $cd(\epsilon) = \epsilon$, so $\widehat{\delta}_{cd}(q_0, cd(\epsilon)) = \widehat{\delta}_{cd}(q_0, \epsilon) = q_0 = \widehat{\delta}(q_0, \epsilon)$, as required.

To show $P(xa)$, we assume the inductive hypothesis $P(x)$. we compute:

$$\begin{aligned}
\widehat{\delta}_{cd}(q_0, cd(xa)) &= \widehat{\delta}_{cd}(q_0, cd(x)aa) && \text{by definition of } cd \\
&= \delta_{cd}(\delta_{cd}(\widehat{\delta}_{cd}(q_0, cd(x)), a), a) && \text{by definition of } \widehat{\delta}_{cd} \\
&= \delta_{cd}(\delta_{cd}(\widehat{\delta}(q_0, x), a), a) && \text{by definition of } \widehat{\delta}_{cd} \\
&= \delta_{cd}(q_{(\widehat{\delta}(q_0,x))a}, a) && \text{by definition of } \delta_{cd} \\
&= \delta(\widehat{\delta}(q_0, x), a) && \text{by definition of } \delta_{cd} \\
&= \widehat{\delta}(q_0, xa) && \text{by definition of } \widehat{\delta}
\end{aligned}$$

9. *We can also define the "string doubling" of x to be xx. For example, $sd(abc) = abcabc$. Show that the set of regular languages is* not *closed under string doubling. In other words, give a regular language L and prove that $sd(L) = \{sd(x) \mid x \in L\}$ is not regular.*

   You can use any theorem proved in class to help prove this result.

Solution Let $L = 0^*1$. Clearly L is regular. Moreover, $sd(L) = \{0^n 1 0^n 1 \mid n \in \mathbb{N}\}$.

This language is not regular. To see this, assume for the sake of contradiction that it is. Then there exists some natural number m as in the pumping lemma. Let $x = 0^m 1 0^m 1$. Clearly $x \in sd(L)$, and $|x| \geq m$, so we can split x into u, v, and w, as in the pumping lemma. We know that $|uv| \leq n$, so v can only contain 0's. Then $x' = uv^2w$ contains more 0's before the first 1 than after, and thus $x' \notin sd(L)$. But the pumping lemma says that $x' \in sd(L)$; this is a contradiction, and thus $sd(L)$ is not regular.

10. *Happy Cat has been shown the following proof, and has promptly turned into Grumpy Cat. Briefly but clearly identify the error which has induced grumpiness.*

    To prove: *The language of the regular expression 0^*1^* is, in fact, not DFA-recognizable.*

    Proof. *Let L be the language of 0^*1^*. Assume there is some DFA M with n states that recognizes L. Let $x = 0^{n-1}11$. Clearly, $x \in L$ and $|x| \geq n$. Therefore according to the Pumping Lemma, we can split x into three parts u, v and w, such that $|uv| \leq n$, $|v| \geq 1$, and $uv^iw \in L$ for all natural numbers i. Let $|v| = n$. Since $|uv| \leq n$, it must be the case that $u = \epsilon$, and $v = 0^{n-1}1$. Then $uv^2w = 0^{n-1}10^{n-1}11$, which is clearly not in L. This contradicts our assumption that there is a DFA which recognizes the language.*

Solution The pumping lemma says there exists some v, but we have chosen a specific v. It may be that the pumping lemma gives some other v (such as 0).

2 Probability problems

1. (a) *Give the definition of variance in terms of expectation.*

Solution

$$\mathrm{Var}(X) = \mathrm{E}((X - \mathrm{E}(X))^2)$$

I would also accept $\mathrm{Var}(X) = \mathrm{E}(X^2) - \mathrm{E}^2(X)$.

(b) *Let X and Y be random variables with* $\mathrm{E}(X) = \mathrm{E}(Y) = 0$. *Prove that* $\mathrm{Var}(X+Y) = \mathrm{Var}(X)+\mathrm{Var}(Y)$. *Make (and clearly state) additional assumptions if necessary.*

Solution We must require that X and Y are independent. Therefore $\mathrm{E}(XY) = 0$. Then we have

$$\begin{aligned}
\mathrm{Var}(X+Y) &= \mathrm{E}\left((X+Y-\mathrm{E}(X+Y))^2\right) && \text{by definition}\\
&= \mathrm{E}\left((X+Y)^2\right) && \text{since } \mathrm{E}(X+Y) = \mathrm{E}(X)+\mathrm{E}(Y) = 0\\
&= \mathrm{E}(X^2+2XY+Y^2) && \text{arithmetic}\\
&= \mathrm{E}(X^2)+2\mathrm{E}(XY)+\mathrm{E}(Y^2) && \text{expectation of the sum is sum of expectations}\\
&= \mathrm{E}(X^2)+\mathrm{E}(Y^2) && \text{since } \mathrm{E}(XY) = 0 \text{ (independence)}\\
&= \mathrm{Var}(X)+\mathrm{Var}(Y) && \text{by definition and fact that } E(X) = E(Y) = 0
\end{aligned}$$

2. (a) *The average human height is 5 feet and 4 inches, and the variance is 2 squared inches. How large a sample must I take so that my estimate of the average will (with 90% probability) be correct to within a half inch?*

Solution Let H denote the "height" random variable, and let E be the estimated average.

The law of large numbers states that $\mathcal{P}(|E - E(H)| \geq \epsilon) \leq \sigma^2(H)/n\epsilon^2$. Plugging in a half inch for epsilon and solving for n, we see that if $n \geq 2/(0.25 \cdot 0.1)$ that the probability of being incorrect is no larger than 0.1.

(b) *A certain high school is divided into two teams: 35% of the students are "beliebers", and the remaining 65% are "directioners".*

90% of the songs on a belieber's playlist will be Justin Bieber songs, while the other 10% will be by One Direction. Directioners are a bit more broad-minded: 80% of their songs will be One Direction songs, while the remaining 20% will be Justin Bieber songs.

A student is selected at random, and a random song is selected from their playlist. It turns out to be "Baby" by Justin Bieber. What is the probability that the student was a directioner?

Solution Let D be the event that the student is a directioner, and let B be the event that the student is a belieber. Let S_D be the event that the selected song was a One Direction song, and S_B be the event that the selected song was a Justin Bieber song. We are given

- $P(B) = 35\%$
- $P(D) = 65\%$
- $P(S_D|D) = 80\%$
- $P(S_D|B) = 10\%$
- $P(S_B|D) = 20\%$
- $P(S_B|B) = 90\%$

We wish to find $P(D|S_B)$. The correct answer is given by Bayes' rule:

$$P(D|S_B) = \frac{P(S_B|D)P(D)}{P(S_B)} = \frac{P(S_B|D)P(D)}{P(S_B|D)P(D) + P(S_B|B)P(B)} = \frac{.2 \cdot .65}{.2 \cdot .65 + .9 \cdot .35}$$

3. *Suppose that a coin has probability .6 of landing heads. You flip it 100 times. The coin flips are all mutually independent.*

   (a) *What is the expected number of heads?*

   Solution Let X_i be the outcome of the ith coin toss; $X_i = 1$ if the ith coin toss lands heads and 0 otherwise. The total number of heads is $Y = X_1 + \cdots + X_{100}$. We are interested in $E(Y)$. By linearity, $E(Y) = E(X_1) + \cdots + E(X_{100}) = 100(.6) = 60$.

   (b) *What upper bound does Markov's Theorem give for the probability that the number of heads is at least 80?*

   Solution Markov's Theorem says that $\Pr(Y \geq 80) \leq E(Y)/80 = 60/80 = 3/4$.

   (c) *What is the variance of the number of heads for a* single *toss? Calculate the variance using either of the equivalent definitions of variance.*

   Solution Let X_i be an indicator variable that is 1 is the ith toss is heads, and 0 otherwise. We are given $P(X_i = 1) = 0.6$. The expectation of X_i is $0.6 \cdot 1 + 0.4 \cdot 0 = 0.6$. Note that $X_i^2 = X_i$, so $E(X_i^2) = 0.6$ as well. Therefore $Var(X_i) = E(X_i)^2 - E(X_i)^2 = 0.6 - 0.36 = 0.24$.

   (d) *What is the variance of the number of heads for 100 tosses? You may use the fact that if $X_1, \ldots, X_n$ are mutually independent, then $\mathrm{Var}(\sum X_i) = \sum \mathrm{Var}(X_i)$; you don't need to prove this.*

   Solution $Var(Y) = Var(X_1) + \cdots + Var(X_{100})$. Since $Var(X_i) = .24$ for $i = 1, \ldots, 100$, $Var(Y) = 100 \times .24 = 24$.

   (e) *What upper bound does Chebyshev's Theorem give for the probability that the number of heads is either less than 40 or greater than 80?*

   Solution By Chebyshev's Theorem. $\Pr(|Y - E(Y)| \geq 20) \leq \mathrm{Var}(Y)/400$. Since $E(Y) = 60$, $\Pr(Y \geq 80 \cup Y \leq 40) \leq 24/400 = .06$.