COVER TIMES FOR MARKOV-GENERATED BINARY SEQUENCES OF LENGTH TWO: APPENDICES

Appendix A. Expected cover times for the cases $k = 3$ and $k = 4$. (fair coin)

A.1. **The case $k = 3$.** We let T_{abc} equal the time to observe pattern abc where $a, b, c \in \{0, 1\}$ and define $e_1 = ET_{000}, e_2 = ET_{111}, e_3 = ET_{001}, e_4 = ET_{110}, e_5 = ET_{010}$,
$e_6 = ET_{101}, e_7 = ET_{100}$, and $e_8 = ET_{011}$. Then applying Lemma 2.4 in ([4]) or Corollary 3.1 in [2], we obtain

$$e_1 = e_2 = 14, e_3 = e_4 = e_7 = e_8 = 8, e_5 = e_6 = 10.$$

If $a \in \mathbb{R}$ and $b \in \mathbb{R}$, then $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$. We calculate the expected cover time using the following expression:

$$\begin{aligned} E(T_1 \vee T_2 \vee \cdots \vee T_8) &= \sum_{i=1}^{8} ET_i - \sum_{i_1 < i_2} E(T_{i_1} \wedge T_{i_2}) \\ &\quad + (-1)^{k-1} \sum_{k=3}^{8} \sum_{i_1 < \cdots < i_k} E(T_{i_1} \wedge \cdots \wedge T_{i_k}) \end{aligned} \tag{A.1}$$

The following matrix, m, is defined by $m(i, i) = 0$, $m(1, j) = 1 = m(j, 1)$ for $i, j \in \{2, \ldots, 9\}$, and $m(i, j) = EN(A_j, A_i)$ for $i, j \in \{2, \ldots, 9\}$ and where A_i equals the pattern defined by e_i. In this notation, $E(N(A_j, A_i)$ can be computed using the results of Li and Fisher/Cui noted above.

$$m = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 14 & 14 & 12 & 12 & 14 & 8 & 14 \\ 1 & 14 & 0 & 12 & 14 & 14 & 12 & 14 & 8 \\ 1 & 2 & 8 & 0 & 6 & 6 & 8 & 2 & 8 \\ 1 & 8 & 2 & 6 & 0 & 8 & 6 & 8 & 2 \\ 1 & 8 & 10 & 6 & 8 & 0 & 6 & 8 & 10 \\ 1 & 10 & 8 & 8 & 6 & 6 & 0 & 10 & 8 \\ 1 & 8 & 6 & 6 & 4 & 4 & 6 & 0 & 6 \\ 1 & 6 & 8 & 4 & 6 & 6 & 4 & 6 & 0 \end{pmatrix}.$$

The vector c in the former paper equals $c = (1, 14, 14, 8, 8, 10, 10, 8, 8)$ where c_i equals the expected time for the pattern corresponding to e_i to occur for $i \in \{2, 3, \ldots, 8\}$.

The *Mathematica* code that follows is based on Theorem 5.1 in [2] or, equivalently on pages 102–103 in [3].

These lines of code enter the matrix m and the vector c.

In[∘]:= m = $\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 14 & 14 & 12 & 12 & 14 & 8 & 14 \\ 1 & 14 & 0 & 12 & 14 & 14 & 12 & 14 & 8 \\ 1 & 2 & 8 & 0 & 6 & 6 & 8 & 2 & 8 \\ 1 & 8 & 2 & 6 & 0 & 8 & 6 & 8 & 2 \\ 1 & 8 & 10 & 6 & 8 & 0 & 6 & 8 & 10 \\ 1 & 10 & 8 & 8 & 6 & 6 & 0 & 10 & 8 \\ 1 & 8 & 6 & 6 & 4 & 4 & 6 & 0 & 6 \\ 1 & 6 & 8 & 4 & 6 & 6 & 4 & 6 & 0 \end{pmatrix}$;

```
In[∘]:= c = {1, 14, 14, 8, 8, 10, 10, 8, 8};
```

We now implement the results cited above in *Mathematica* and obtain the exact (and approximate) expected cover time for the case $k = 3$.

```
In[•]:= f[x_] := Inverse[m[[x, x]]].c[[x]]

In[•]:= g[n_] := (s = Subsets[Range[2, 9], {n}];
          t = Map[Prepend[#, 1] &, s];
          Total[Map[f, t][[All, 1]]])

In[•]:= 80 + Sum[(-1) ^ (n + 1) g[n], {n, 2, 8}]

Out[•]= 89259/3640

In[•]:= 80 + Sum[(-1) ^ (n + 1) g[n], {n, 2, 8}] // N

Out[•]= 24.5217
```

Here is a *Mathematica* simulation of 100,000 random binary sequences that computes the mean of the 100,000 corresponding cover times. This demonstrates the empirical consistency with the theoretical result.

```
In[•]:= Mean[
   Table[Length[NestWhile[Join[#, {RandomChoice[{1/2, 1/2} → {0, 1}]}] &,
      {RandomChoice[{1/2, 1/2} → {0, 1}]},
      Nand[MatchQ[#, {___, PatternSequence[0, 0, 0], ___}],
        MatchQ[#, {___, PatternSequence[0, 0, 1], ___}],
        MatchQ[#, {___, PatternSequence[0, 1, 0], ___}],
        MatchQ[#, {___, PatternSequence[1, 0, 0], ___}],
        MatchQ[#, {___, PatternSequence[0, 1, 1], ___}],
        MatchQ[#, {___, PatternSequence[1, 0, 1], ___}],
        MatchQ[#, {___, PatternSequence[1, 1, 0], ___}],
        MatchQ[#, {___, PatternSequence[1, 1, 1], ___}]] &]], {100000}]] // N

Out[•]= 24.5508
```

A.2. **The case $k = 4$.** We first use *Mathematica* to identify the patterns of length 4.

```
In[•]:= a = Tuples[{0, 1}, 4](*in lexicographical order*)

Out[•]= {{0, 0, 0, 0}, {0, 0, 0, 1}, {0, 0, 1, 0}, {0, 0, 1, 1}, {0, 1, 0, 0},
   {0, 1, 0, 1}, {0, 1, 1, 0}, {0, 1, 1, 1}, {1, 0, 0, 0}, {1, 0, 0, 1},
   {1, 0, 1, 0}, {1, 0, 1, 1}, {1, 1, 0, 0}, {1, 1, 0, 1}, {1, 1, 1, 0}, {1, 1, 1, 1}}
```

Next, we determine the positions of overlaps to find ET_i where T_i equals the number of observations required to obtain the pattern a_i for $i \in \{1, 2, \ldots, 16\}$.

```
In[•]:= v =
  Table[
   Flatten[Position[Table[a[[k]][[n ;; 4]] == a[[k]][[1 ;; 5 - n]], {n, 4, 1, -1}] // Boole,
     1]], {k, 1, Length[a]}]

Out[•]= {{1, 2, 3, 4}, {4}, {1, 4}, {4}, {1, 4}, {2, 4}, {1, 4},
   {4}, {4}, {1, 4}, {2, 4}, {1, 4}, {4}, {1, 4}, {4}, {1, 2, 3, 4}}
```

Here, we apply Corollary 3.1 in [2] to find $c_{i+1} = ET_i$ for $i \in \{1, 2, \ldots, 16\}$ and as before set $c_1 = 1$.

```
In[•]:= c = Prepend[Table[Total[2 ^ # &[v[[n]]]], {n, 1, Length[a]}], 1]

Out[•]= {1, 30, 16, 18, 16, 18, 20, 18, 16, 16, 18, 20, 18, 16, 18, 16, 30}
```

We construct the matrix m as defined earlier.

```
In[∘]:= e[i_, j_] :=
          Flatten[Position[Table[a[[j - 1]][[n ;; 4]] == a[[i - 1]][[1 ;; 5 - n]], {n, 4, 1, -1}] // Boole,
            1]]

In[∘]:= m[i_, j_] := c[[i]] - Total[2 ^ # &[e[i, j]]]

In[∘]:= m[i_, i_] := 0; m[i_, 1] := 1; m[1, j_] := 1

In[∘]:= m = Table[m[i, j], {i, 1, 17}, {j, 1, 17}];

In[∘]:= MatrixForm[m]

Out[∘]//MatrixForm=
```

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	30	28	30	24	30	28	30	16	30	28	30	24	30	28	30
1	2	0	14	16	10	16	14	16	2	16	14	16	10	16	14	16
1	12	10	0	18	12	18	16	18	12	10	16	18	12	18	16	18
1	10	8	14	0	10	16	14	16	10	8	14	16	10	16	14	16
1	16	14	8	18	0	14	16	18	16	14	8	18	16	14	16	18
1	18	16	10	20	18	0	18	20	18	16	10	20	18	16	18	20
1	16	14	16	10	16	14	0	18	16	14	16	10	16	14	16	18
1	14	12	14	8	14	12	14	0	14	12	14	8	14	12	14	16
1	16	14	12	14	8	14	12	14	0	14	12	14	8	14	12	14
1	18	16	14	16	10	16	14	16	18	0	14	16	10	16	14	16
1	20	18	16	18	20	10	16	18	20	18	0	18	20	10	16	18
1	18	16	14	16	18	8	14	16	18	16	14	0	18	8	14	16
1	16	14	16	10	16	14	8	10	16	14	16	10	0	14	8	10
1	18	16	18	12	18	16	10	12	18	16	18	12	18	0	10	12
1	16	14	16	10	16	14	16	2	16	14	16	10	16	14	0	2
1	30	28	30	24	30	28	30	16	30	28	30	24	30	28	30	0

As for the case $k = 3$, we repeat the calculations for the case $k = 4$ and obtain the exact and approximate expected cover time.

```
In[∘]:= f[x_] := Inverse[m[[x, x]]].c[[x]];

In[∘]:= g[n_] := (s = Subsets[Range[2, 17], {n}];
          t = Map[Prepend[#, 1] &, s];
          Total[Map[f, t][[All, 1]]])

In[∘]:= Total[c] - 1 + Sum[(-1) ^ (n + 1) g[n], {n, 2, 17}]

Out[∘]= 15 196 470 103 027 446 764 838 236 318 296 131 920 851 968 094 230 950 060 807 620 630 943 693 /
          259 180 013 898 712 074 394 595 904 741 652 282 392 543 237 486 671 525 526 056 835 614 400

In[∘]:= N[%]

Out[∘]= 58.6329
```

We perform 100,000 simulations and find the empirical expected cover time.

```
In[◦]:= Mean[
   Table[Length[NestWhile[Join[#, {RandomChoice[{1/2, 1/2} → {0, 1}]}] &,
      {RandomChoice[{1/2, 1/2} → {0, 1}]},
      Nand[MatchQ[#, {___, PatternSequence[0, 0, 0, 0], ___}],
        MatchQ[#, {___, PatternSequence[0, 0, 0, 1], ___}],
        MatchQ[#, {___, PatternSequence[0, 0, 1, 0], ___}],
        MatchQ[#, {___, PatternSequence[0, 0, 1, 1], ___}],
        MatchQ[#, {___, PatternSequence[0, 1, 0, 0], ___}],
        MatchQ[#, {___, PatternSequence[0, 1, 0, 1], ___}],
        MatchQ[#, {___, PatternSequence[0, 1, 1, 0], ___}],
        MatchQ[#, {___, PatternSequence[0, 1, 1, 1], ___}],
        MatchQ[#, {___, PatternSequence[1, 0, 0, 0], ___}],
        MatchQ[#, {___, PatternSequence[1, 0, 0, 1], ___}],
        MatchQ[#, {___, PatternSequence[1, 0, 1, 0], ___}],
        MatchQ[#, {___, PatternSequence[1, 0, 1, 1], ___}],
        MatchQ[#, {___, PatternSequence[1, 1, 0, 0], ___}],
        MatchQ[#, {___, PatternSequence[1, 1, 0, 1], ___}],
        MatchQ[#, {___, PatternSequence[1, 1, 1, 0], ___}],
        MatchQ[#, {___, PatternSequence[1, 1, 1, 1], ___}]] &]], {100000}]] // N

Out[◦]= 58.6463
```

Appendix B. Application of *Mathematica* and R to Theorem 2.1 and Theorem 3.1

We first use *Mathematica* to calculate and simulate the result of Theorem **??** for a Markov chain having transition probability matrix

$$P = \begin{pmatrix} 3/4 & 1/4 \\ 3/5 & 2/5 \end{pmatrix}$$

and initial distribution μ where $\mu(0) = \mu(1) = 1/2$. The following *Mathematica* code and output calculates the exact value of $E(t_{\text{cov}}(2))$.

```
In[•]:= a = 1 / 2; b = 1 / 2; α = 1 / 4; β = 3 / 5;

In[•]:= a α + b β + ((1 + α) - b β (1 - β))/(α (1 - β)) + ((1 + β) - a α (1 - α))/(β (1 - α)) - (1 + α)/(1 - α β)

Out[•]= 41621/3060

In[•]:= N[%]

Out[•]= 13.6016
```

Next, we use *Mathematica* to simulate the Markov chain 100,000 times and compute the mean cover time.

```
In[•]:= t = Table[Length[NestWhile[Join[#, {(a = RandomReal[];
              Which[Last[#] == 0 && 0 < a ≤ 3 / 4, 0, Last[#] == 0 && 3 / 4 < a ≤ 1, 1,
               Last[#] == 1 && 0 < a ≤ 3 / 5, 0, Last[#] == 1 && 3 / 5 < a ≤ 1, 1])}] &,
          {RandomChoice[{1 / 2, 1 / 2} → {0, 1}]},
          Nand[MatchQ[#, {___, PatternSequence[0, 1], ___}],
            MatchQ[#, {___, PatternSequence[0, 0], ___}],
            MatchQ[#, {___, PatternSequence[1, 0], ___}],
            MatchQ[#, {___, PatternSequence[1, 1], ___}]] &]], {100000}];

In[•]:= Mean[t] // N

Out[•]= 13.4036
```

Next, we apply *Mathematica* to demonstrate Theorem 3.1 for this same Markov chain and obtain the exact cover time probabilities $P(t_{\text{cov}}(2) = n)$ for $n \in \{5, 6, \ldots, 15\}$.

```
In[]:= f[a_, α_, β_] :=
        α (1 - β)
         (a Sum[Binomial[k, n - 3 - k] (1 - α) ^ (2 k + 3 - n) (α β) ^ (n - 3 - k), {k, Ceiling[(n - 3) / 2], n - 3}] +
          β (1 - a) Sum[Binomial[k, n - 4 - k] (1 - α) ^ (2 k + 4 - n) (α β) ^ (n - 4 - k),
            {k, Ceiling[(n - 4) / 2], n - 4}])

In[]:= g[a_, α_, β_] := a α (1 - β) (1 - α) ^ (n - 3);

In[]:= cover[n_] := f[a, α, β] + f[1 - a, β, α] - (g[a, α, β] + g[1 - a, β, α]) +
         α β (1 - α) (1 - β) / (α - β) ((1 - β) ^ (n - 4) - (1 - α) ^ (n - 4)) -
         (a (α - β) + β (1 - α)) (α β) ^ ((n - 3) / 2) * Boole[OddQ[n]] -
         (1 - β - a (α - β)) (α β) ^ ((n - 2) / 2) * Boole[EvenQ[n]];

In[]:= Table[cover[n], {n, 5, 15}]
```

$$\text{Out}[\,]= \left\{\frac{9}{100}, \frac{927}{8000}, \frac{837}{8000}, \frac{287\,577}{3\,200\,000}, \frac{2\,384\,721}{32\,000\,000}, \frac{79\,459\,839}{1\,280\,000\,000}, \frac{332\,788\,149}{6\,400\,000\,000}, \frac{4\,516\,220\,529}{102\,400\,000\,000}, \frac{193\,690\,821\,483}{5\,120\,000\,000\,000}, \frac{6\,720\,055\,830\,171}{204\,800\,000\,000\,000}, \frac{14\,709\,685\,518\,909}{512\,000\,000\,000\,000}\right\}$$

```
In[]:= Table[cover[n], {n, 5, 15}] // N

Out[]= {0.09, 0.115875, 0.104625, 0.0898678, 0.0745225,
        0.062078, 0.0519981, 0.0441037, 0.0378302, 0.0328128, 0.0287299}
```

We compare these theoretical probabilities with empirical ones based on the prior 100,000 simulations.

```
In[]:= counts = Counts[t] // KeySort;

In[]:= covertimedistribution = counts / 100 000 // N

Out[]= <|5 → 0.09, 6 → 0.11656, 7 → 0.10423, 8 → 0.09092, 9 → 0.07456, 10 → 0.06161, 11 → 0.05045,
        12 → 0.04388, 13 → 0.03768, 14 → 0.03251, 15 → 0.02959, 16 → 0.02608, 17 → 0.02323,
        18 → 0.01974, 19 → 0.01854, 20 → 0.01577, 21 → 0.01535, 22 → 0.01298, 23 → 0.01183,
        24 → 0.01094, 25 → 0.00911, 26 → 0.00838, 27 → 0.00888, 28 → 0.00747, 29 → 0.00662,
        30 → 0.00597, 31 → 0.00605, 32 → 0.00529, 33 → 0.00495, 34 → 0.00409, 35 → 0.0041,
        36 → 0.00351, 37 → 0.00316, 38 → 0.00326, 39 → 0.00311, 40 → 0.00257, 41 → 0.00226,
        42 → 0.00216, 43 → 0.00192, 44 → 0.00172, 45 → 0.00148, 46 → 0.00143, 47 → 0.00124,
        48 → 0.00146, 49 → 0.00124, 50 → 0.00097, 51 → 0.00105, 52 → 0.00082, 53 → 0.00075,
        54 → 0.00065, 55 → 0.00065, 56 → 0.0005, 57 → 0.00063, 58 → 0.0004, 59 → 0.00048, 60 → 0.00031,
        61 → 0.0004, 62 → 0.00044, 63 → 0.00037, 64 → 0.00028, 65 → 0.00029, 66 → 0.00028,
        67 → 0.00021, 68 → 0.00021, 69 → 0.0002, 70 → 0.00015, 71 → 0.0002, 72 → 0.00012,
        73 → 0.00017, 74 → 0.00017, 75 → 0.00021, 76 → 0.00009, 77 → 0.00013, 78 → 0.00009,
        79 → 0.0001, 80 → 0.00009, 81 → 0.00009, 82 → 0.00004, 83 → 0.00009, 84 → 0.00003,
        85 → 0.00002, 86 → 0.00007, 87 → 0.00003, 88 → 0.00002, 89 → 0.00003, 90 → 0.00001,
        91 → 0.00001, 92 → 0.00002, 93 → 0.00001, 94 → 0.00001, 95 → 0.00003, 96 → 0.00002,
        99 → 0.00004, 100 → 0.00002, 101 → 0.00002, 102 → 0.00002, 103 → 0.00001, 104 → 0.00001,
        108 → 0.00001, 110 → 0.00001, 111 → 0.00001, 112 → 0.00001, 119 → 0.00001, 126 → 0.00001|>
```

Here we use R to generate simulations and empirical probabilities.

```
library(stringr) # loads the package
trials <- 100000 # number of trials
P <- matrix(c(3/4,1/4,3/5,2/5),nrow=2,ncol=2,byrow=T)
simlist <- numeric(trials) # sets up vector whose components will
#          equal the simulated cover times
for (i in 1:trials){
  pattern1 <- "00"
  pattern2 <- "01"
  pattern3 <- "11"
  pattern4 <- "10"
  sequence <- sample(c(0,1),1,prob=c(1/2,1/2),replace=T) # random coin tosses
  sequence1 <- as.character(sequence) # converts the elements of the sequence to a list
#        of strings
  sequence2 <- paste(sequence1,collapse="") # converts the sequence to one string
  while(!(str_detect(sequence2,pattern1)&str_detect(sequence2,pattern2)&
          str_detect(sequence2,pattern3)&str_detect(sequence2,pattern4)))
  {flip <- sample(1:2,1,prob=P[tail(sequence,1)+1,])-1
  sequence <- append(sequence,flip)
  sequence1 <- as.character(sequence)
  sequence2 <- paste(sequence1,collapse="")}
  simlist[i] <- str_length(sequence2)}
mean(simlist) #Expected cover time
```

```
## [1] 13.35225
```

```
tabulate(simlist)/100000 #Empirical distribution for cover time length based on 100000
                     simulations
```

```
##   [1] 0.00000 0.00000 0.00000 0.00000 0.09086 0.11694 0.10438 0.08963
##   [9] 0.07498 0.06256 0.05246 0.04333 0.03755 0.03249 0.02882 0.02546
##  [17] 0.02260 0.01982 0.01809 0.01584 0.01445 0.01330 0.01238 0.01132
##  [25] 0.00973 0.00910 0.00813 0.00747 0.00699 0.00597 0.00616 0.00482
##  [33] 0.00478 0.00404 0.00374 0.00344 0.00323 0.00320 0.00275 0.00236
##  [41] 0.00226 0.00213 0.00211 0.00174 0.00157 0.00146 0.00126 0.00111
##  [49] 0.00118 0.00111 0.00087 0.00086 0.00072 0.00073 0.00058 0.00057
##  [57] 0.00059 0.00054 0.00050 0.00042 0.00027 0.00036 0.00041 0.00029
##  [65] 0.00025 0.00027 0.00017 0.00027 0.00014 0.00025 0.00015 0.00015
##  [73] 0.00007 0.00014 0.00009 0.00012 0.00011 0.00010 0.00010 0.00007
##  [81] 0.00009 0.00002 0.00007 0.00009 0.00007 0.00002 0.00009 0.00003
##  [89] 0.00001 0.00004 0.00002 0.00002 0.00000 0.00001 0.00000 0.00000
##  [97] 0.00002 0.00000 0.00002 0.00001 0.00000 0.00001 0.00001 0.00000
## [105] 0.00002 0.00001 0.00000 0.00000 0.00001 0.00000 0.00000 0.00000
## [113] 0.00001 0.00000 0.00002 0.00000 0.00001 0.00001
```

Acknowledgement

The second and third authors were supported by Lafayette College EXCEL grants.

References

[1] Abbas M. Alhakin, *A simple combinatorial algorithm for de Bruijn sequences* Amer. Math. Monthly. **117** (2010), 728–732.

[2] Evan Fisher and Shiliang Cui, *Patterns Generated by m^{th} order Markov chains.* Stat. & Prob. Letters. **80** (2010) 1157–1166.

[3] H.U. Gerber and S.Y.R. Li, *The occurrence of sequence patterns in repeated experiments and hitting times in a Markov chain*, Stochastic Process. Appl. **11** (1981), 101–108.

[4] Shoo-Yen Robert Li, *A martingale approach to the study of occurrence of sequence patterns in repeated experiments*, Ann. Probab. **8** (1980), 1171–1176.

[5] David Levin and Yuval Peres, *Markov Chains and Mixing Times*, 2nd ed., American Mathematical Society, Providence, R.I., 2017.

[6] Terry R. McConnell, *The Expected Time to Find a String in a Random Binary Sequence*, `http://barnyard.syr.edu/cover.pdf`, 2001.

[7] Sidney I. Resnick, *Adventures in Stochastic Processes*, Birkhäuser, Boston, 1992.

[8] Richard P. Stanley, *Enumerative Combinatorics, Volume 2*, Cambridge University Press, Cambridge UK, 1999.