

```

▽APPEND[□]
[0] APPEND X
[1] AUT←AUT,[1]X
[2] ▽
    ▽BAD[□]▽
[0] Z←N BAD V;A;B
[1] A←K STRINGS N
[2] B←ϕ"A
[3] Z←v/V[1+K⊥"A]^V[1+K⊥"B]
    ▽CLOSE[□]▽
[0] CLOSE
[1] □NUNTIE -1
    ▽COMMA[□]▽
[0] Z←COMMA V
[1] Z←⌈V
[2] Z[(Z=' ')/⊥ρZ]←','
    ▽CREATE[□]▽
[0] CREATE F
[1] A CREATE FILENAME AND TIE WITH 1
[2] ('/Users/shallit/Desktop/',F)□NCREATE -1
    ▽DELDEAD[□]▽
[0] Z←DELDEAD G;A;B;I;C;D
[1] A G is a matrix in Grail format
[2] A result is new Grail automaton with the single nonaccepting state
[3] A deleted in all transitions
[4] A←(v/G='F')/[1]G
[5] B←,A,' '
[6] B←⊥(B∈'0123456789 ')/B
[7] Z←(0,-1⊥ρG)ρ''
[8] I←0
[9] L1:
[10] I←I+1
[11] →(I>1⊥ρG)/0
[12] C←G[I;]
[13] →(v/C∈'SF')/L2
[14] D←⊥C
[15] →(∼^/D[1 3]∈B)/L1
[16] L2:
[17] Z←Z,[1]C
[18] →L1
|

```

Dyalog APL code for
 "Words avoiding reversed
 subwords, revisited",
 paper by Fleischer and
 Shallit, 2019.

See README.txt for
 documentation.

```

▽GETDATA[[]]▽
[0] GETDATA F
[1] OPEN F
[2] A←READ
[3] CLOSE
[4] B← $\sim 1$  0+[]AV[3]MATRIFY A
[5] C←DELDEAD B
[6] D←GRAILTOGV C
[7] E←GRAILTOMAT A
[8] G←GRAILTOMAPLE A
[9] H←GRAILTOAUT A
[10] L←F;'. '
[11] CREATE(L↑F), 'gv'
[12] WRITE D
[13] CLOSE
[14] CREATE((L-1)↑F), 'maple'
[15] WRITE G
[16] CLOSE
▽GRAILTOAUT[[]]▽
[0] Z←GRAILTOAUT V;A;B;C;D;N;E;F;G;H;L;M;P;Q;R;S;T;U
[1] A converts vector with grail representation to automaton
[2] A←[]AV[3]MATRIFY V
[3] B←( $\sim \wedge$ /A=' ')/[1]A
[4] C←( $\sim \vee$ /B='(')/[1]B
[5] D← $\pm$ , C, ' '
[6] N←( $\imath 0$ ) $\rho$ ( $\rho D$ )÷3
[7] D←(N,3) $\rho D$ 
[8] E←1+[/,D[:1 3]
[9] G←REFORM D
[10] H←( $\vee$ /B='S')/[1]B
[11] L←,H, ' '
[12] M←(L $\in$ '0123456789 ')/L
[13] P← $\pm$ M
[14] Q←E $\rho$ 0
[15] Q[P+1]←1
[16] R←( $\vee$ /B='F')/[1]B
[17] S←,R, ' '
[18] T←(S $\in$ '0123456789 ')/S
[19] U← $\pm$ T
[20] V←E $\rho$ 0
[21] V[U+1]←1
[22] Z←(Q)(G)(V)

```

```

VGRAILTOGV
[0] Z←GRAILTOGV M;Z1;Z2;Z3;Z4;I;J;A;B;C;D;E;F;S;ST;N
[1] A M is a matrix with each line a row of a grail spec of an automaton
[2] Z1←'digraph G {' ,CR
[3] Z1+Z1,'rankdir = LR;' ,CR
[4] Z5+Z3+Z4+' '
[5] N←(ρM)[1]
[6] S←ST+10
[7] TR←(¬v/Me'SF')/[1]M
[8] TR2+±,TR,' '
[9] TR3←(((ρTR2)÷3),3)ρTR2
[10] TR4+uc[2]TR3[;1 3]
[11] I←0
[12] L0:
[13] I←I+1
[14] →(I>ρTR4)/L7
[15] A←>TR4[I]
[16] B←(TR3[;1 3]∧.=A)/[1]TR3
[17] Z5+Z5,(⌘A[1]),' -> ',(⌘A[2]),'[ label = "',(COMMA B[;2]),'"';',CR
[18] →L0
[19] L7:
[20] I←0
[21] L1:
[22] I←I+1
[23] →(I>N)/L4
[24] A←RTB M[I;]
[25] →((10↑A)∧.= '(START) |-')/L2
[26] →((¬10↑A)∧.= '-| (FINAL)')/L3
[27] →L1
[28] L3:
[29] F←±¬10↑A
[30] S←S,F
[31] Z3+Z3,'node [shape = doublecircle, label="',(⌘F),' "fontsize=12]',(⌘F),';',CR
[32] →L1
[33] L2:
[34] Z4+Z4,'node [shape = point ]; qi',CR
[35] Z4+Z4,'qi ->0;',CR
[36] →L1
[37] L4:
[38] ST←vST
[39] ST←(¬ST∈S)/ST
[40] J←0
[41] L5:
[42] J←J+1
[43] →(J>ρST)/L6
[44] Z3+Z3,'node [shape = circle, label="',(⌘ST[J]),'", fontsize=12]',(⌘ST[J]),';',CR
[45] →L5
[46] L6:
[47] Z←Z1,Z3,Z4,Z5,'}' ,CR

```

```

VGRAILTOMAPLE
[0] ZZ+GRAILTOMAPLE V;A;B;C;D;N;E;F;G;H;L;M;P;Q;R;S;T;U
[1] A converts vector with grail representation to matrix rep
[2] A+[]AV[3]MATRIFY V
[3] B+(-^/A=' ')/[1]A
[4] C+(-v/B=' ')/[1]B
[5] D+±,C,' '
[6] N+(10)ρ(ρD)÷3
[7] D+(N,3)ρD
[8] E+u,D[;1 3]
[9] ZZ+'with(LinearAlgebra);',CR
[10] ZZ+ZZ,'Digits := 50;',CR
[11] ZZ+ZZ,'m := Matrix(1..',(⌘ρE),',1..',(⌘ρE),',fill=0);',CR
[12] F+(2ρρE)ρ0
[13] F SET2c[2]E1D[;1 3]
[14] H+(v/B='S')/[1]B
[15] L+,H,' '
[16] M+(Lε'0123456789 ')/L
[17] P+±M
[18] Q+(ρE)ρ0
[19] Q[E1P]+1
[20] R+(v/B='F')/[1]B
[21] S+,R,' '
[22] T+(Sε'0123456789 ')/S
[23] U+±T
[24] V+(ρE)ρ0
[25] V[E1U]+1
[26] ZZ+ZZ,'u := Vector[row](',(⌘ρE),',fill=0);',CR
[27] ZZ+ZZ,'u[1] := 1;',CR
[28] ZZ+ZZ,'v := Vector[column](',(⌘ρE),',fill=0);',CR
[29] I+0
[30] L1:
[31] I+I+1
[32] →(I>ρU)/L2
[33] ZZ+ZZ,'v['',(⌘E1U[I]),'] := 1;',CR
[34] →L1
[35] L2:
[36] ZZ+ZZ,'p := MinimalPolynomial(m,x);',CR
[37] ZZ+ZZ,'factor(p);',CR
[38]

```

```

VINIT[0]V
0] INIT
1] AUT←0 3p0
  VMAKE6[0]V
0] Z←K MAKE6 N;A;I;C;D;J
1] A makes Grail script constructing automaton for K letter alphabet accepting those words
2] A such that if W is a factor, and |W| ≥ N, then W reversed is not a factor
3] A using formula due to Lukas Fleischer
4] A←'(', (1+ε'+', '(c1 0)⌘"-1+1K),')*'
5] XX←'# making automaton for K = ', (⌘K),'; N = ', (⌘N),CR
6] XX+XX,'echo "', (Np'0'),',"',CR
7] XX+XX,'echo "', A, (Np'0'), A, '"' | ./retofm | ./fmdeterm | ./fmmin | ./fmcment > d0',CR
8] XX+XX,'./fmstats d0',CR
9] I←J+0
10] L1:
11] I←I+1
12] →(I=K*N)/L2
13] C←1 0⌘D←(NpK)⌘I
14] →((K⌘D)>K⌘ϕD)/L1 A avoid duplicate work
15] →(Λ/C=ϕC)/L3 A special case for palindromes
16] J←J+1
17] XX+XX,'echo "', C, "'",CR
18] XX+XX,'echo "', A, C, A, '"' | ./retofm | ./fmdeterm | ./fmmin | ./fmcment > a', (⌘J),CR
19] XX+XX,'echo "', A, (ϕC), A, '"' | ./retofm | ./fmdeterm | ./fmmin | ./fmcment > b', (⌘J),CR
20] XX+XX,'./fmunion a', (⌘J), ' b', (⌘J), ' | ./fmdeterm | ./fmmin > c', (⌘J),CR
21] XX+XX,'./fmcross d', (⌘J-1), ' c', (⌘J), ' | ./fmdeterm | ./fmmin > d', (⌘J),CR
22] XX+XX,'./fmstats d', (⌘J),CR
23] →L1
24] L3:
25] J←J+1
26] XX+XX,'echo "', C, "'",CR
27] XX+XX,'echo "', A, C, A, '"' | ./retofm | ./fmdeterm | ./fmmin | ./fmcment > c', (⌘J),CR
28] XX+XX,'./fmcross d', (⌘J-1), ' c', (⌘J), ' | ./fmdeterm | ./fmmin > d', (⌘J),CR
29] XX+XX,'./fmstats d', (⌘J),CR
30] →L1
31] L2:
32] XX+XX,'cp d', (⌘J), ' aut', (⌘K), (⌘N), '.txt'

```

```

[0] K MAKEAUTQ N;DD;E
[1] CREATE 'autq', (K), (N), '.txt'
[2] DD←'(START) |- 1', CR
[3] K WARQUEUE N
[4] DD←DD, (AUT), CR
[5] E←v, AUT[;1 3]
[6] DD←DD, ((pE), 1)pE, (((pE), 11)p' -| (FINAL)'), CR
[7] WRITE DD
[8] CLOSE
VMMAKESCRIPT[ ]v
[0] K MAKESCRIPT N
[1] # makes Grail script constructing automaton for K-letter alphabet avoiding
[2] # N-reversed factors, and stores it in file called autKN.txt
[3] CREATE 'script', (K), (N), '.txt'
[4] K MAKE6 N
[5] WRITE XX
[6] CLOSE
VMATRIFY[ ]v
[0] Z←C MATRIFY X;I;D
[1] I←(X=C)/1pX
[2] D←(I, 1+pX)-1+0, I
[3] Z←((pD), [ /D)p(, D°.≥1[ /D)\, (X≠C)/X
VOPEN[ ]v
[0] OPEN F
[1] # OPEN FILENAME F
[2] ('/Users/shallit/Desktop/', F)⌈NTIE ~1
VREAD[ ]v
[0] Z←READ
[1] Z←NREAD ~1 82 ~1
VREFORM[ ]v
[0] Z←REFORM M;N;K;I
[1] K←1+[ /M[;2] # alphabet size
[2] N←1+[ /,M[;1 3] # a number of states
[3] Z←(N, K)pN
[4] I←0
[5] L1:
[6] I←I+1
[7] →(I>1+pM)/0
[8] Z[M[I;1]+1;M[I;2]+1]+M[I;3]
[9] →L1
|

```

```

VRLB[[]]V
[0] Z←RLB V
[1] Z←(¬1+(V=' ')10)÷V
VRTB[[]]V
[0] Z←RTB V
[1] Z←φRLBφV
VSET[[]]V
[0] Z←M SET L;I
[1] I←0
[2] Z←M
[3] L1:
[4] I←I+1
[5] →(I>1↑ρL)/0
[6] A←L[I]
[7] Z[A[1];A[2]]+Z[A[1];A[2]]+1
[8] →L1
VSET2[[]]V
[0] M SET2 L;I;A
[1] I←0
[2] L1:
[3] I←I+1
[4] →(I>1↑ρL)/0
[5] A←L[I]
[6] ZZ+ZZ,'m[',(⊘A[1]),',',',',(⊘A[2]),'] := ', 'm[',(⊘A[1]),',',',',(⊘A[2]),'] + 1;',CR
[7] →L1
VSTRINGS[[]]V
[0] Z←K STRINGS N
[1] →(0≠NC'K')/L1
[2] K←2
[3] L1:
[4] Z←c[1](NρK)τ-1+1K*N
VUPDATE[[]]V
[0] Z←A UPDATE C;B;D;H
[1] →((ρA)=(K*N)+N-1)/L1
[2] Z←A,C
[3] →0
[4] L1:
[5] B←(K*N)↑A
[6] D←(1-N)↑A
[7] H←1+K1D,C
[8] B[H]←1
[9] Z←B,1↑D,C

```

```

[25] L←0,1,0,0
      VWARQUEUE[0]V
[0] K WARQUEUE N
[1] A makes automaton accepting all finite words over alphabet of size K
[2] A avoiding reversed subwords for length ≥ N
[3] INIT
[4] Q←X←,c(K*N)ρ0
[5] L1:
[6] →(0=ρQ)/L2
[7] ±(0∈1000|(ρQ),ρX)/'□←''queue size ''',(ρρQ),' 'total size ''',ρρX'
[8] A→Q[1]
[9] Q←1↓Q
[10] C←-1
[11] L4:
[12] C←C+1
[13] →(C=K)/L1
[14] B←A UPDATE C
[15] →((cB)∈X)/L5
[16] →(N BAD(K*N)↑B)/L4
[17] Q←Q,cB
[18] X←X,cB
[19] APPEND(X↑cA),C,ρX
[20] →L4
[21] L5:
[22] APPEND(X↑cA),C,X↑cB
[23] →L4
[24] L2:
[25] 'all done'
[26] 'size is ',ρρX
[27] 'total is ',ρρν(K*N)↑''X
      VWRITE[0]V
[0] Z←WRITE X
[1] Z←X □NAPPEND -1

```