Appendix A

Random Number Generator

Each of the Source-code listing below represents an extract from the C++ implementation of various random number generator algorithms from [PTVF02] . We have chosen in our experiments the implementation suggested by L'Ecuyer [PTVF02] with Bays-Durham shuffle and added safeguard because it has the ability of providing fairly randomized sequence of numbers in the order of hundreds.

Source-code 9. *// C++ extract for random number generator of Park and Miller*

```
#include ''nr.h''
DP ran1(int &idum){
   const int IA = 16807, IM = 214783647, IQ = 127773, IR = 2836, NTAB = 32;
   const int NDIV = (1+(IM-1)/NTAB);
   const DP EPS = 3.0e-16, AM = 1.0/IM, RNM= (1.0-EPS);
   static vect_INT iv(NTAB);
   int j, k;
   DP temp;
   if (idum <= 0|| !iy){            //Initialize.
       if (-idum <1) idum = 1;      //Be sure to prevent idum = 0.
```

```
        else idum = -idum;
        for (j = NTAB+7; j>=0; j--){ //Load shuffle table (after 8 warm-ups).
            k = idum/IQ;
            idum = IA*(idum-k*IQ)-IR*k;
            if (idum < 0) idum +=IM;
            if (j < NTAB) iv[j] = idum;
        }
        iy = iv[0];
    }
    k = idum/IQ;                      //Start here when not initializing.
    idum = IA*(idum-k*IQ)-IR*k;       //Compute idum = (IA*idum)%IM without
    if (idum < 0) idum += IM;         //overflow by schrage's method
    j = iy/NDIV;                      //Will be in the range 0..NTAB-1.
    iy = iv[j];                       //Output previously stored value and
    iv[j] = idum;                     //refill the shuffle  table.
    if ((temp = AM*iy)>RNMX)
        return RNMX;              //Because user don't expect endpoint values
    else return temp;
}
```

Source-code 10. *// C++ extract for random number generator of L'Ecuyer with // Bays-Durham shuffle and added safeguard*

```
#include ''nr.h''
DP ran2(int &idum){
    const int IM1 = 2147483563, IM2 = 2147483399;
    const int IA1 = 40014, IA2 = 40692, IQ1 = 53668, IQ2 = 52774;
    const int IR1 = 12211, IR2 = 3791, NTAB = 32, IMM1=IM1-1;
```

```
const int NDIV = 1+IMM1/NTAB;
const DP EPS = 3.0e-16, RNMX = 1.0-EPS, AM = 1.0/DP(IM1);
static int idnum2 = 123456789, iy=0;
static Vect_INT iv(NTAB);
int j, k;
DP temp;
if (idum <= 0){                          //Initialize.
    idum = (idum ==0 ? 1: -idum); //Be sure to prevent idum = 0.
    idum2 = idum;
    for (j = NTAB+7; j>=0; j--){ //Load shuffle table (after 8 warm-ups).
        k = idum/IQ1;
        idum = IA1*(idum-k*IQ1)-k*IR1;
        if (idum < 0) idum +=IM1;
        if (j < NTAB) iv[j] = idum;
    }
    iy = iv[0];
}
k = idum/IQ1                        //Start here when not initializing.
//Compute idum = (IA1*idum)% IM1 without
idum = IA1*(idum-k*IQ1)-k*IR1;
if (idum < 0) idum += IM1;
k = idum2/IQ2;
//Compute idum2 = (IA2*idum)\% IM2 likewise.
idum2 = IA2*(idum2-k*IQ2)-k*IR2;
if (idum2 < 0) idum2 += IM2;
j = iy/NDIV;                        //Will be in the range 0..NTAB-1.
/*Below idum is shuffled, idum and idum2 are
```

```
    combined to generate output.*/
    iy = iv[j]-idum2
    iv[j] = idum;
    if (iy < 1) iy += IMM1;
    if ((temp = AM*iy) > RNMX)
     return RNMX;       //Because user don't expect endpoint values.
    else return temp;
}
```

Source-code 11. *// C++ extract for random number generator using //Knuth's suggestions*

```
#include <cstdlib>
#include ''nr.h''
using namespace std;
DP ran3(int &idum){
   statc int inext, inextp;
   static int iff=0;
   const int MBIG = 10000000000, MSEED=161803398,MZ=0;
   const DP  FAC = (1.0/MBIG);
   /* The value 56 (range ma[1..55])is special and
      should not be modified; see Knuth.*/
   static Vect_INT ma(56);
   int i,ii,k,mj,mk;
   if (idum < 0 || iff == 0){    //Initialization.
       iff=1;
       /*Initialize ma[55] using the seed idum and the
         large number MSEED.*/
```

```
    mj=labs(MSEED-labs(idum));
    mj \%=MBIG;
    ma[55]=mj;
    mk=1;
    for (i=1;i<=54;i++){
        /* Now initialize the rest of the table,
           in a slightly random order,
           with numbers that are not especially random.
        */
        ii-(21*i)%55;
        ma[ii]=mk;
        mk=mj-mk;
        if (mk < int(MZ)) mk += MBIG;
        mj=ma[ii];
    }
    for (k = 0; k<4,k++)
        //We randomize them by ''warming up generator.''
    for (i=1;i<= 54; i++){
        ma[i] -= ma[1+(i+30)%55];
        if (ma[i] < int(MZ)) ma[i] += MBIG;
    }
    //Prepare indices for our first generated number.
    inext=0;
    //The constant 31 is special; see Knuth.
    inextp=31;
    idum=1;
}
```

```
    // Here is where we start, except on initialization.
    if (++inext == 56)
          //Increment inext and inextp, wrapping around
          inext=1;
    if (++inextp == 56) inextp=1;     56 to 1.
    // Generate a new random number subtractively.
    mj=ma[inext]-ma[inextp];
    // Be sure that it is in range.
    if (mj < int int(MZ)) mj +=MBIG
    ma[inext]=mj;                     //Store it,
    //and output the derived uniform deviate.
    return mj*FAC;
}
```

Appendix B

Data Collected

The tables below depict data collected for various experiments. In the case of two-alphabet symbols and the string recognition experiments (described in Chapter 7), we only provide sample data for automata of up to 500 states. The remaining part of the table may be obtain upon request to the author.

For the Single symbol experiment described in Chapter 5, tables B.1 to B.6 depict data collected for each of the technique used namely, table-driven, switch and nested conditional statements, jump table, linear search and direct jump. A few number of values reproduced were considered as outliers since their order of magnitude appeared to be out of range compared to the large majority of data. Rows containing outliers values were simply deleted from the table. Table B.7 represents the averaged data of each of the last nine columns of the tables B.1, B.2, B.3, B.4, B.5 and B.6 without taking into account the problem size (number of accepting symbols - first column).

For the two-alphabet symbols experiments (Chapter 7), the data was collected based on both table-driven and hardcoded implementation. The hardcoded data relied only on the jump table implementation since it was considered to be the best of the other hardcoded methods investigated. Table B.8 depicts the sample data (up to 500 states) collected for both table-driven and hardcoded implementation.

Our String recognition experiment (Chapter 7) was based on an alphabet size of 10 symbols. 4000 different data were items collected. Table B.9 depicts an extract of the first 50 rows of data collected for both table-driven and hardcoded experiments using searching and direct indexing.

# accepting symbols	Min valid trans	Max valid trans	Avrg valid trans	Min no trans	Max no trans	Avrg no trans	Min both cases	Max both cases	Avrg both cases
93	88	88	88	88	88	88	88	88	88
94.6	88	88	88	88	88	88	88	88	88
96.3	88	88	88	88	88	88	88	88	88
98.6	88	88	88	88	88	88	88	88	88
100.9	88	88	88	88	88	88	88	88	88
105.2	88	88	88	88	88	88	88	88	88
105.7	88	88	88	88	88	88	88	88	88
109.5	88	88	88	88	88	88	88	88	88
109.8	88	88	88	88	88	88	88	88	88
110.5	88	88	88	88	88	88	88	88	88
110.56	88	88	88	88	88	88	88	88	88
113.2	88	88	88	88	117.2	102.6	88	117.2	88.02
116.5	88	88	88	88	88	88	88	88	88
117.5	88	88	88	88	120.8	104.4	120.8	88	88.01
119	88	88	88	88	88	88	88	88	88
119.2	88	88	88	88	88	88	88	88	88
120	88	88	88	88	88	88	88	88	88
120.2	88	88	88	88	88	88	88	88	88
121.2	88	88	88	88	88	88	88	88	88
130.2	88	123.2	105.6	88	88	88	88	123.2	88.02
131.3	88	88	88	88	88	88	88	88	88
136.6	88	88	88	88	88	88	88	88	88
136.9	88	88	88	88	88	88	88	88	88
139.1	88	88	88	88	93.2	90.6	88	93.2	88.00
140.2	88	88	88	88	88	88	88	88	88
146.1	88	88	88	88	88	88	88	88	88
148.6	88	88	88	88	88	88	88	88	88
149.8	88	88	88	88	88	88	88	88	88
152.5	88	88	88	88	88	88	88	88	88
153.3	88	88	88	88	88	88	88	88	88
160.5	88	88	88	88	88	88	88	88	88
161.8	88	88	88	88	94.4	91.2	88	94.4	88.00
162.3	88	88	88	88	88	88	88	88	88
183	88	88	88	88	88	88	88	88	88
186.7	88	108.4	98.2	88	150.8	119.4	88	150.8	88.05

Table B.1: The Table-driven Experiment Data

# accepting symbols	Min valid trans	Max valid trans	Avrg valid trans	Min no trans	Max no trans	Avrg no trans	Min both cases	Max both cases	Avrg both cases
85	94.4	147.2	100.25	88	140	93.58	88	147.2	95.75
90.9	93.6	122	99.62	88.8	110.4	93.81	88.4	124.4	96.34
91.8	94.8	117.2	98.79	88	116.4	94.15	88	120	96.20
92.7	96	154.4	100.18	88	139.6	93.54	88	154.4	96.20
102.9	95.2	121.6	99.45	88.4	119.2	93.55	88.4	121.6	95.77
107.3	94.8	118	98.90	88.8	115.6	93.32	88.4	118	95.06
107.9	94.4	123.2	99.06	88.4	114.4	93.95	88	123.2	96.35
108.7	94	117.2	98.73	88.8	114	93.78	88.4	117.2	95.63
109.6	93.6	121.6	97.98	88.8	118.4	92.79	88.4	124	94.25
110.9	93.6	118	98.80	88.4	112.4	94.05	88	118	96.17
111	92	124.4	98.98	88.8	123.6	92.42	88.4	130.8	94.02
116	95.2	121.6	100.41	88	118	93.60	88	121.6	96.03
116.8	93.2	116.8	100.15	88.4	118	94.35	88.4	119.6	96.60
117	94.4	122.4	99.21	88.4	121.2	94.06	88	126.8	96.06
117.8	94.8	122.4	100.44	88	140.4	95.30	88	144.8	97.69
123.1	91.6	142.8	97.71	88.8	134.4	92.51	88	144	94.19
126	95.2	144	99.74	88.4	120.8	93.67	88.4	146.4	96.30
127.4	94.4	120.4	99.82	88.4	119.2	93.692	88	122.4	95.87
128.1	92.4	121.2	98.85	88	110	92.86	88	121.2	94.60
128.3	94.4	126.4	100.95	88	122.4	93.59	88	126.4	96.04
129.5	91.2	122	99.22	88.4	118	93.49	88.4	125.2	95.44
134.3	91.6	116	99.80	88.8	120	92.02	88.8	122	93.70
134.5	94.8	126.8	99.86	88	116.8	93.25	88	126.8	95.69
143.6	93.6	119.6	98.24	88	116.8	92.87	88	120.4	94.82
143.6	93.2	113.2	98.74	88.8	112.4	93.85	88.8	118.8	95.56
145.2	92.8	123.2	99.10	88.8	114	94.09	88.4	123.6	96.04
147.4	96.4	146	100.32	88	116.4	94.41	88	146	97.01
148.5	92.8	122	100.17	88	120	93.75	88	124.8	96.01
149	95.6	131.6	100.56	88	117.6	94.72	88	131.6	97.32
150.7	94.4	116	98.27	88.4	114.4	92.82	88.4	116.8	94.55
160.56	94.67	124	100.89	88	120.89	94.09	88	126.67	96.16
163.6	94	123.6	99.72	88.4	118.8	92.99	88.4	126.8	95.27
179.1	92	116.8	98.21	88.8	110.4	92.60	88.8	118.8	94.28
180.2	92	121.2	99.30	88	113.2	93.39	88	121.2	95.37
196.1	94	119.6	98.47	88	112.4	92.83	88	119.6	94.73

Table B.2: The Switch Statements Data

# accepting symbols	Min valid trans	Max valid trans	Avrg valid trans	Min no trans	Max no trans	Avrg no trans	Min both cases	Max both cases	Avrg both cases
85.00	93.60	163.60	118.73	119.60	328.40	125.37	93.60	334.80	122.32
92.70	94.40	176.00	117.80	123.20	222.00	126.80	94.40	224.80	122.99
98.10	94.00	172.00	118.18	126.80	209.60	128.77	94.00	211.20	123.34
98.80	92.40	131.20	107.27	110.00	159.60	110.94	92.40	161.60	109.88
103.40	93.60	166.00	120.96	124.00	197.20	129.61	93.60	200.00	125.63
107.90	92.40	214.40	132.43	140.40	256.00	151.47	92.40	264.40	141.91
108.10	93.60	199.20	129.22	136.40	251.60	144.05	93.60	253.20	136.71
108.60	89.60	221.60	129.89	146.80	271.60	150.93	89.60	272.80	137.86
108.70	91.60	201.20	128.79	137.60	298.00	143.44	91.60	298.00	136.73
109.80	92.00	164.80	122.40	124.40	212.40	132.43	92.00	212.80	127.53
110.60	92.00	159.20	118.89	125.60	214.00	129.53	92.00	214.00	125.06
110.90	91.20	202.40	130.86	140.00	240.00	149.15	91.20	248.00	139.83
111.90	93.20	177.60	125.34	131.60	236.40	139.19	93.20	237.60	132.01
115.30	92.00	216.80	129.31	145.20	258.00	147.00	92.00	260.00	138.34
116.00	94.40	180.00	125.51	136.00	250.40	138.93	94.40	250.40	131.95
117.00	94.40	206.40	130.83	140.40	254.80	147.81	94.40	261.20	138.94
119.50	93.20	192.80	125.64	133.60	248.80	139.66	93.20	251.20	133.59
122.50	92.80	208.80	135.52	148.80	276.80	157.60	92.80	277.20	145.25
123.10	94.80	207.60	121.54	134.40	239.60	137.08	94.80	258.80	130.33
125.30	95.60	172.40	119.92	126.40	238.00	129.98	95.60	238.00	124.88
130.80	94.40	174.00	117.82	130.40	223.60	131.97	94.40	223.60	125.66
137.50	93.60	217.20	132.75	152.00	275.20	153.86	93.60	275.20	142.89
138.70	91.20	210.40	126.36	135.60	240.80	140.88	91.20	240.80	133.65
140.00	94.40	210.80	136.24	144.80	264.40	155.73	94.40	265.60	146.73
140.90	92.00	202.40	130.20	140.80	264.80	147.35	92.00	265.60	139.91
141.90	93.20	198.00	129.58	144.00	268.80	146.82	93.20	268.80	138.86
143.10	93.20	195.20	129.86	138.80	245.20	145.38	93.20	246.80	136.80
145.70	96.40	230.80	136.67	155.20	297.60	161.04	96.40	298.40	148.86
146.80	90.40	215.20	137.02	148.40	283.20	158.56	90.40	283.20	148.06
150.80	92.80	215.20	132.24	140.80	266.80	150.38	92.80	268.00	140.85
165.50	95.60	245.20	135.39	159.20	331.60	162.86	95.60	332.40	148.65
165.70	95.20	205.20	135.48	150.00	278.80	156.97	95.20	278.80	146.74
179.60	96.00	235.60	142.47	155.20	306.80	167.10	96.00	309.20	156.38
183.00	100.00	148.00	112.13	112.00	184.00	112.39	100.00	184.00	112.35
181.60	95.36	220.76	139.25	154.36	314.44	162.40	95.36	315.08	150.65

Table B.3: the Nested Conditional Statements Data

# accepting symbols	Min valid trans	Max valid trans	Avrg valid trans	Min no trans	Max no trans	Avrg no trans	Min both cases	Max both cases	Avrg both cases
3.44	6.67	216.89	13.62	4.00	216.89	16.18	4.00	216.89	14.74
10.70	7.20	177.60	15.99	4.00	165.60	8.64	4.00	181.60	11.56
15.40	11.20	235.60	18.12	4.00	233.20	8.87	4.00	235.60	11.49
22.30	8.80	296.80	15.60	4.00	289.60	9.55	4.00	296.80	12.28
31.00	6.40	218.00	16.27	4.00	200.40	8.85	4.00	220.00	11.85
39.70	9.60	280.40	17.26	4.00	281.60	8.46	4.00	290.00	10.84
49.20	8.80	268.00	15.70	4.00	267.20	8.67	4.00	294.80	10.91
54.70	5.60	298.00	17.56	4.00	273.20	9.58	4.00	298.00	12.33
60.50	8.80	299.60	16.92	4.00	326.00	9.59	4.00	326.00	12.48
68.90	7.20	318.00	17.79	4.00	257.20	8.62	4.00	318.00	11.61
77.70	8.80	352.80	17.36	4.00	338.40	9.67	4.00	364.80	12.75
83.30	10.40	300.40	18.44	4.00	296.40	9.65	4.00	316.80	12.72
92.70	8.00	304.00	18.09	4.00	256.40	8.50	4.00	304.00	11.83
99.20	10.40	241.60	18.47	4.00	291.60	9.50	4.00	322.40	12.52
111.50	9.60	279.60	18.70	4.00	308.00	9.41	4.00	317.60	12.29
119.60	7.20	393.60	19.41	4.00	424.80	10.03	4.00	434.00	13.73
126.60	9.60	377.20	20.55	3.60	358.80	9.43	3.60	383.20	13.66
134.10	9.60	339.60	19.28	3.60	355.20	9.11	3.60	371.20	12.03
144.60	7.20	4063.20	23.50	4.00	400.80	10.66	4.00	4101.60	16.95
151.80	7.20	329.60	20.61	4.00	404.00	10.29	4.00	405.20	14.07
156.60	8.00	369.60	21.34	4.00	340.40	9.78	4.00	406.80	13.41
163.30	9.60	391.60	22.16	4.00	380.40	11.35	4.00	391.60	15.94
168.30	7.20	389.20	21.58	4.00	4018.80	10.43	4.00	4037.20	14.36
174.90	8.80	355.20	22.89	4.00	337.60	9.78	4.00	366.80	14.43
185.30	8.80	443.60	22.85	4.00	446.00	10.54	4.00	447.20	15.97
194.40	9.60	312.80	22.75	4.00	324.80	9.23	4.00	340.00	12.76
201.00	8.80	4218.80	25.57	4.00	359.20	10.98	4.00	4224.80	17.48
208.50	6.40	8906.00	28.69	4.00	430.80	10.39	4.00	8923.60	17.81
214.40	8.00	416.80	23.39	4.00	365.20	10.05	4.00	425.20	14.59
221.80	6.40	338.80	22.63	4.00	356.00	10.56	4.00	371.20	15.31
228.90	8.80	398.00	23.58	4.00	383.20	10.88	4.00	399.60	15.95
236.30	8.00	371.20	25.82	4.00	431.60	11.18	4.00	444.80	16.47
245.90	8.80	365.20	23.44	4.00	364.40	10.28	4.00	375.20	14.58
253.17	8.00	374.00	21.74	4.00	370.67	9.28	4.00	379.33	12.63

Table B.4: The Jump Table Data

# accepting symbols	Min valid trans	Max valid trans	Avrg valid trans	Min no trans	Max no trans	Avrg no trans	Min both cases	Max both cases	Avrg both cases
3.44	2.22	52.89	8.87	2.67	24.89	4.77	2.22	52.89	7.12
10.90	6.40	65.60	14.02	6.80	45.60	7.99	5.20	67.20	10.43
16.40	7.60	111.60	16.04	8.40	105.60	9.81	6.00	112.80	12.24
25.30	8.40	248.40	22.87	15.60	238.00	18.40	6.80	248.40	20.19
33.40	10.80	256.00	21.86	16.00	252.00	19.12	10.00	256.00	19.68
42.80	6.40	198.40	23.68	18.40	196.80	21.45	6.40	199.60	21.09
51.10	9.60	404.00	27.96	21.60	405.60	28.15	9.60	405.60	27.21
55.31	7.36	297.60	29.44	24.56	296.96	30.50	6.96	299.76	29.40
61.80	12.40	341.20	31.27	26.40	324.00	34.22	11.20	345.60	32.53
71.00	8.80	388.80	30.51	26.40	392.00	32.36	8.40	392.40	31.03
79.50	13.20	391.60	37.70	37.20	439.60	44.41	11.60	439.60	40.25
85.90	7.20	3192.80	36.73	34.40	3491.60	41.60	7.20	6354.40	38.83
93.80	9.60	351.60	34.37	30.40	356.80	39.59	9.60	357.20	36.42
100.30	14.00	478.40	39.17	38.40	478.40	47.97	14.00	478.40	43.97
113.00	12.40	541.60	43.74	46.00	601.20	53.69	12.00	644.00	47.92
120.90	10.40	784.00	50.21	57.20	766.80	64.57	10.40	794.40	57.63
127.50	12.80	686.40	46.64	52.40	746.80	59.70	12.40	762.40	53.52
134.80	12.80	558.80	43.23	45.60	535.60	53.12	12.80	560.40	47.45
144.60	8.80	853.20	60.98	76.00	816.80	83.73	8.80	926.40	70.97
151.26	9.68	856.92	54.76	64.80	735.28	72.94	8.48	908.24	61.69
156.60	10.00	3860.00	50.31	55.60	659.20	62.73	10.00	3860.00	56.46
163.30	12.80	1076.00	64.22	78.00	950.00	88.43	12.80	1076.00	76.17
168.30	7.20	602.00	51.89	59.60	673.20	67.24	6.40	750.40	59.12
175.80	11.20	15007.60	70.93	69.60	708.80	77.96	11.20	15007.60	73.46
185.30	10.40	3969.20	78.12	86.00	880.00	112.23	10.40	4100.80	94.09
194.30	8.80	682.80	47.97	52.40	712.40	60.87	8.00	727.60	55.94
201.70	12.80	1012.80	75.27	98.00	1105.60	107.92	12.80	1136.00	91.04
209.30	7.60	775.20	64.55	79.60	528.00	88.78	7.60	840.40	77.07
216.70	10.40	1088.40	71.49	89.20	951.20	99.42	10.40	1099.60	86.70
224.60	8.80	945.60	78.76	99.60	1016.40	114.83	8.80	1132.80	96.84
231.20	11.60	1054.00	77.52	101.60	913.60	111.15	11.60	1148.00	94.04
239.30	12.40	1080.40	80.81	108.00	3980.00	120.15	12.00	4022.40	99.84
247.30	10.00	1046.40	64.88	80.40	949.20	89.36	10.00	1046.40	79.33
253.75	13.50	1018.00	66.29	84.00	21755.50	99.54	13.50	21755.50	87.06

Table B.5: The Linear Search Data

# accepting symbols	Min valid trans	Max valid trans	Avrg valid trans	Min no trans	Max no trans	Avrg no trans	Min both cases	Max both cases	Avrg both cases
3.70	12.00	39.20	20.31	12.00	35.20	16.86	12.00	44.80	18.91
11.50	13.60	56.00	21.01	12.00	56.40	20.39	12.00	58.00	20.63
15.15	15.36	84.00	21.03	13.60	85.24	21.14	12.00	85.40	21.04
22.30	14.40	138.80	21.06	14.40	138.80	20.89	12.00	138.80	20.96
31.00	16.00	193.60	21.90	14.40	194.00	21.88	12.00	194.00	21.87
39.70	16.00	214.40	21.97	14.00	214.40	21.99	12.00	214.80	22.01
49.20	17.60	249.60	22.25	13.20	249.60	22.28	12.00	249.60	22.29
54.70	13.60	186.80	22.90	16.40	187.20	22.91	12.00	187.20	22.89
60.50	14.40	189.20	23.63	13.20	282.40	23.89	12.00	282.40	23.77
68.90	12.80	130.40	23.18	12.80	132.00	23.16	12.00	132.40	23.16
77.70	14.00	219.20	22.69	12.80	220.00	22.97	12.00	220.00	22.93
83.30	14.00	156.00	22.27	12.80	157.20	22.36	12.00	157.20	22.33
91.00	15.20	180.40	24.46	13.20	257.60	24.70	12.00	257.60	24.67
96.90	18.00	238.80	24.72	12.80	238.00	24.83	12.00	242.00	24.80
108.00	16.00	254.40	21.81	13.20	280.00	21.89	12.00	280.00	21.89
117.50	15.60	276.80	24.72	14.80	23851.20	36.40	12.00	23854.00	34.71
124.30	15.20	271.60	25.42	14.80	270.40	25.40	12.00	277.20	25.45
130.30	18.40	27695.60	48.15	12.80	267.20	25.47	12.00	27697.20	36.19
138.20	14.00	200.00	24.64	12.00	226.80	24.70	12.00	251.20	24.73
148.30	13.60	253.60	25.55	15.60	27638.00	38.18	12.80	27639.20	34.84
153.80	15.60	303.60	25.05	14.00	295.60	25.25	12.00	318.00	25.13
158.80	15.20	304.40	26.06	13.20	306.00	26.23	12.00	306.00	26.21
165.50	14.40	278.40	26.29	12.80	284.80	26.30	12.00	285.60	26.27
171.40	13.60	284.00	26.65	13.20	285.60	26.41	12.00	285.60	26.45
179.14	14.56	268.80	26.82	12.92	266.16	26.80	12.00	270.56	26.81
189.20	14.40	273.60	27.71	14.80	300.80	27.78	12.00	302.00	27.74
197.00	16.40	282.00	27.74	12.00	296.00	27.81	12.00	298.00	27.80
204.60	13.60	275.20	28.24	14.00	276.00	28.15	12.00	279.60	28.23
211.50	14.00	272.80	27.70	12.40	303.20	27.58	12.00	305.20	27.64
218.90	14.40	29771.60	55.24	12.80	8078.40	33.74	12.00	29774.00	39.39
227.60	16.00	329.20	26.87	13.60	305.60	26.91	12.00	331.20	26.87
235.70	15.20	281.20	29.29	12.80	278.00	29.26	12.00	284.40	29.23
246.40	15.20	332.00	27.99	12.80	325.60	27.78	12.80	337.20	27.79
253.14	15.43	4751.43	31.91	13.14	275.43	29.89	12.00	4754.86	30.76

Table B.6: The Direct Jump Data

	Valid Transition			No transition			Both cases		
	Min	**Max**	**Avrg**	**Min**	**Max**	**Avrg**	**Min**	**Max**	**Avrg**
Table-driven	88.00	89.59	88.79	88.00	91.90	89.95	88.00	92.90	88.00
Switch Statements	93.86	124.70	99.40	88.34	119.16	93.54	88.21	126.89	95.63
Nested Conditional	93.56	195.94	127.50	137.51	254.55	143.24	93.56	256.73	135.49
Jump Table	8.46	322.14	19.61	3.97	321.49	9.85	3.97	341.49	13.26
Linear Search	9.69	545.84	41.53	44.45	516.78	51.22	9.22	574.60	46.12
Direct Jump	14.84	223.71	24.54	13.31	232.71	24.43	12.03	237.10	24.50

Table B.7: Averaged Data collected independently to the problem size

	Table-driven			Hardcode		
# of states	Avrg min time	Avrg max time	Avrg avrg time	Avrg min time	Avrg max time	Avrg avrg time
10	424	649	436	57	235	69
20	840	987	847	112	721	150
30	1247	1396	1254	169	2034	275
40	1642	1864	1655	224	3670	413
50	2032	2249	2057	279	4033	488
60	2465	2717	2479	335	4842	585
70	2865	3127	2888	390	7089	754
80	3293	3499	3307	460	5842	772
90	3699	4016	3717	536	6291	887
100	4030	4309	4111	613	8663	1093
110	4512	4619	4525	667	7440	1096
120	4929	5201	4944	760	9703	1314
130	5324	5535	5338	855	10072	1437
140	5753	5809	5758	929	9405	1464
150	6139	6377	6155	1015	13007	1733
160	6490	6793	6575	1251	11886	1897
170	6978	7267	6994	1432	13043	2135
180	7418	7538	7424	1616	15137	2438
190	7658	8143	7797	1824	17380	2775
200	8038	8478	8211	2096	13985	2847
210	8485	185347	17475	2253	15786	3125
220	8975	9324	9064	2374	17676	3339
230	9397	9651	9417	2647	17863	3618
240	9678	9934	9699	3063	19773	4036
250	10238	10587	10261	3335	20361	4337
260	10444	10685	10659	3641	20673	4648
270	10938	11045	10953	3896	20474	4897
280	11319	11760	11361	4313	22638	5387
290	11706	11999	11886	4706	25121	5890
300	12089	12667	12322	5090	25033	6264
310	12740	12986	12754	5430	124328	12658
320	13124	13268	13150	5839	26174	6990
330	13559	13888	13580	6234	26281	7369
340	13998	14215	14009	6706	202403	17768
350	14246	14610	14279	6974	28786	8161
360	14554	14816	14787	7386	30275	8611
370	15114	15210	15142	7771	31341	9313
380	15330	15601	15369	8059	29948	9192
390	15766	16198	16009	8319	31951	9534
400	16104	16635	16159	8549	34266	9863
410	16843	17104	16858	8781	36855	10207
420	16857	17140	17106	9025	36610	10428
430	17299	17671	17623	9274	21243	9898
440	17778	18080	18060	9505	21164	10111
450	18315	18548	18336	9733	23876	10456
460	18890	18972	18896	9981	20511	10528
470	19137	19424	19315	10198	22078	10814
480	19650	19949	19665	10416	22415	11033
490	19837	20254	20030	10632	24994	11367
500	20197	20387	20207	10849	21277	11388

Table B.8: Sample Data for the two-alphabet symbols Experiments

	String Recognition using Linear Search						String Recognition using Direct Index					
	Table-driven			Hardcode			Table-driven			Hardcode		
# of states	Avrg min time	Avrg max time	Avrg avrg time	Avrg min time	Avrg max time	Avrg avrg time	Avrg min time	Avrg max time	Avrg avrg time	Avrg min time	Avrg max time	Avrg avrg time
10	1785	2720	1805	576	3419	636	340	659	348	91	2470	141
20	3797	4506	3811	1283	5800	1383	698	1151	709	184	6165	308
30	5717	6553	5734	1908	8260	2056	1049	1403	1059	282	9089	468
40	7639	8554	7659	2539	10899	2723	1400	1856	1411	372	11895	615
50	9580	10304	9594	3203	15339	3474	1753	2232	1764	468	13011	740
60	11558	12340	11574	4214	18978	4532	2109	2580	2119	643	16480	994
70	13652	14429	13667	4741	23830	5154	2460	2772	2468	781	21418	1226
80	15597	16394	15613	5550	22115	5917	2818	3291	2829	857	21008	1296
90	17461	18228	17476	5936	23944	6348	3169	3618	3180	931	23370	1420
100	19519	20274	19534	6771	30223	7312	3532	3948	3541	1173	26865	1748
110	21699	22426	21714	7776	35837	8393	3889	4201	3897	1320	31447	1970
120	23670	24419	23685	8941	36895	9548	4229	4464	4237	1526	35666	2699
130	25641	26398	25656	10161	39970	10805	4484	5000	4581	2038	38148	2840
140	27678	28262	27690	10882	1807381	47007	4934	5240	4943	2591	39015	3391
150	29718	30468	29733	12408	48226	13188	5289	5740	5300	2786	47801	3746
160	31680	32378	31694	12920	49600	13730	5652	6031	5660	3067	45179	3973
170	33660	34405	33675	13911	59202	14874	5962	6327	6025	3670	51786	5174
180	35802	36615	35818	15496	67729	16597	6311	6852	6369	4141	58319	5281
190	37712	38539	37730	16583	67413	17636	6704	7084	6714	4669	63788	5903
200	39614	40471	39632	17870	73731	19130	7070	7399	7079	5274	67080	6581
210	41519	59819	41903	19053	75763	20264	7408	7749	7417	5659	73443	7080
220	43418	64129	43850	20158	80724	21973	7681	8101	7797	5979	77634	7481
230	45332	46227	45350	20932	85096	22260	8116	8521	8144	6430	79685	7993
240	47432	48286	47449	21774	143616	25614	8441	8918	8451	6913	79821	8433
250	49297	50064	49312	23176	89264	24576	8843	9293	8855	7474	84791	9091
260	51243	52036	51259	23627	97556	25169	9136	9658	9149	7820	91989	9573
270	53221	586024	63945	24741	169972	29372	9564	10136	9583	8413	100716	10296
280	55065	56194	55088	26060	112703	27851	9845	10270	9856	8967	103267	10899
290	57087	57924	57105	26393	111594	28186	10186	10790	10203	9306	105460	11794
300	59172	59963	59188	29386	112494	31140	10520	11153	10543	9792	106645	11790
310	61114	1828776	96498	30386	120012	32225	11042	11411	11052	10139	113321	12259
320	63040	63860	63057	31493	117679	33253	11402	11902	11434	10400	118619	12612
330	65059	87578	65531	30909	127564	32900	11811	12295	11869	10965	121293	13216
340	66949	84500	67321	31462	127632	33436	12116	12796	12183	11374	126034	13754
350	69011	70112	69034	32470	127860	34418	12546	13162	12561	11808	126155	14180
360	70904	201363	73550	33958	146674	36286	12934	13639	12992	12101	160247	15124
370	72907	91467	73302	35109	148358	37434	13274	13892	13292	12685	141602	15753
380	75197	749073	88713	37630	153435	40075	13689	14572	13730	13167	142012	15803
390	77031	207068	79680	38628	154922	41030	14090	14795	14120	13513	159357	19333
400	79178	95878	79546	39642	158204	420600	14561	15084	14591	14096	145721	16779
410	81149	82636	81180	39143	164597	41719	14898	15786	14935	14666	149139	17433
420	83096	754076	97026	39278	165208	41850	15257	15984	15279	14836	156112	17717
430	85338	86679	85366	40309	185388	46014	15758	16570	15775	15335	161386	18311
440	87146	104880	87534	43800	166546	46333	16273	16782	16344	15673	164625	18699
450	89338	90906	89388	42103	180878	44920	16533	17174	16579	16321	173763	19550
460	91136	142873	92217	45832	191300	48810	17176	17558	17220	16755	186600	20252
470	93113	94907	93149	46588	193856	49592	17512	226983	21746	17098	165980	20126
480	95192	114015	95597	44737	195712	47820	17959	57126	18743	17354	178416	23529
490	97083	98513	97111	45025	193362	48062	18224	18533	18232	17947	397911	26038
500	99238	121324	99711	46985	198716	52620	18716	38069	19130	18249	191298	21780

Table B.9: Sample Data for the String Recognition Experiment with Searching and Direct Indexing

Bibliography

[AhSU86] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools. Addison Wesley 1986

[AhUl72] Alfred V. Aho, and Jeffrey D. Ullman. The Theory of Parsing,Translation, and Compiling Volume 1: Parsing. Prentice Hall, 1972

[Andersen et al 99] Bjarne S. Andersen, Alexander Karaivanov, Jerzy Wasniewski, Fred Gustavson, and Plamen Y. Yalamov. Linear Algebra With Recursive Algorithms. http://lawra.uni-c.dk/lawra/abstracts/KazDolny99/KazDolny99.html

[BhPr95] Achyutram Bhamidipaty, and Todd A. Proebsting. Very fast YACC-Compatible Parsers (for Very Little Effort), Department of Computer Science university of Arizona, 1995 .

[Cleo03] L. G. W. A. Cleophas. A New Taxonomy and Toolkit of Keyword Pattern Matching Algorithms. MSC Thesis, Faculty of Computing Science, Eindhoven University of Tehnology, The Netherlands, September 2003.

[CroHa97] Maxime Crochemre, and Christophe Hancart. Automata for Matching Patterns. Hadbook of Formal Language. Springler Verlang 1997.

[Douglas et al 00] Craig C. Douglas, Jonathan Hu, Mohamed Iskandarani, Markus Kowarschik, Ulrich Rüde, and Christian Weiß. Maximizing Cache Memory Usage for Multigrid Algorithms. Proceedings of the International Workshop held at Beijing, China, August 2-6, 1999, Lecture Notes in Physics, pp. 124ff. Springer, August 2000.

[DoGK84] J. J. Dongarra, F. G. Gustafson, and A. Karp. Implementing linear algebra algorithms for dense matrices on a vector pipeline machine. SIAM Review, 26 (1984), pp. 91-112

[ELI02] Syntactic Analysis - Context-Free Grammars and Parsing. http://www.cs.colorado.edu/ eliuser/elionline4.3/syntax_1.html.

[FrHe91] Christopher W. Fraser, and Robert R. Henry. Hard-Coding Bottom-up code generation tables to save time and space. Software-Practice & Experiences 21, 1(Jan. 1991), 1-12.

[Ger02] Richard Gerber. The Software Optimization Cookbook: High-performance Recipes for the Intel Architecture. Intel Corporation, 2002.

[GRJA91] Dick Grune, Ceriel J.H. Jacobs, Parsing Techniques : A Practical Guide, Prentice Hall,1991

[Herr03] Jack Herrington, Code Generation in Action. Manning Publiction 2003

[HoWh88] R. Niegel Horspool, and M. Whitney. Even faster LR parsing. Software Practice and Experiences, 20(6):515-535, June 1988.

[Intel] Intel Corporation. The Intel Architecture Optimization Reference Manual. http://www.intel.com/design/pentiumiii/manuals/

[John75] Stephen C Johnson. YACC - Yet Another Compiler-Compiler. Bell Labs 1975.

[KeWK03] E.Ketcha Ngassam, Bruce W.Watson, Derrick G. Kourie. Preliminary Experiments on Hardcoding Finite Automata. 8th International Conference on

Implementation and Application of Automata, CIAA, 2003. LNCS 2759, pages 299-300. Springer, 2003.

[KIM02] Paul Kimmel. The Visual Basic .Net Developer's Book. Addison-Wesley 2003.

[Krus88] F.E.J Kruseman Arez. On a Recursive Ascent Parser. Information Processing Letters, Volume 29, No 4, p. 201-206, November 1988

[KWK03] E.Ketcha Ngassam, Bruce W.Watson, Derrick G. Kourie. Hardcoding Finite Automata Processing. Annual Conference of the South African Institute of Computer Scientists and Information Technologists, SAICSIT 2003. pageff 111. ACM, 2003.

[LePa81] Harry R. Lewis, and Christo H. Papadimitrou. Elemenets of The Theory of Computation. Prentice Hall, 1981

[LeSc75] M. E. Lesk and E. Schmidt. Lex – A Lexical Analyzer Generator. Computer Science Technical Report 39, AT&T Bell Laboratorie, Murray Hill, NJ,USA, Oct. 1975

[McN82] Robert McNaughton. Elementary Computability, Formal Languages and Automata. Prentice Hall, 1982

[Penn86] Thomas J. Pennello. Very fast LR parsing. In Proceedings of the SIGPLAN '86 Symposium on Compiler Construction, pages 145-151, 1989.

[Pfah90] Peter Pfahler. Optimizing directly executable LR parsers. In Compiler Compilers: Third International Workshop CC'90, pages 179-192, October 1990

[PTVF02] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical recipes in C++: the art of scientific computing. Cambridge, UK; New York: Cambridge University Press, 2002.

[Wat02] Bruce W. Watson. Directly Constructing Minimal DFAs: Combining Two Algorithms by Brzozowski. SART/SACJ, No 29, 2002, 17-23

[Wat95] Bruce W. Watson. Taxonomies and Toolkits of Regular Language Algorithms. PhD Thesis, Faculty of Computing Science, Eindhoven University of Technology, The Netherlands, September 1995.