Tutorial on Formal Verification of Programs (Part I)

Wenhui Zhang

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences

P.O.Box 8718, Beijing 100190, China

2012-12-19

This part argues that model checking is the right choice for certifying the correctness of certain types of programs.

1. Introduction

Program correctness is an important issue.

Suppose that we are given the following program, where the functions in(), isr(), and isk() are specified in Appendix A.

```
#include <stdio.h>
/**************************************************/
int in();
int isr(int x,int k);
int isk(int n,int k);
/**************************************************/
int main()
{
    int n=0;
    int m=0;
    int k=1;

    printf("INFO:   system is now active\n");
    while (1) {
        n=in();
        m=isr(n,k);
        k=isk(n,k);
        printf("RESULT: %i\n\n",m);
    }
}
/**************************************************/
```

Let “isr” be the name of the executable file of the program. A sample execution of the program is as follows.

```
./isr
```

```
INFO:   system is now active
N:      2
RESULT: 1

N:      10
RESULT: 3

N:      20
RESULT: 4

N:      30
INFO:   the input number must be in {0,...,20}

N:      5
RESULT: 2

N:
```

Suppose that the correctness claim of the program is as follows.

Whenever the input is a number between 0 and 20, the program prints the integer part of the square root of the number.

How do we know that the program is correct with respect to the claim?

The methods for assuring that a program is correct include testing and formal verification.

1.1. On Testing

In order to have evidence on whether the program is correct, we may compile and run the program.

In this case, the program is apparently correct with a limited number of tests. However, we do not know whether the program is correct. As said by Edsger W. Dijkstra, program testing can be used to show the presence of bugs, but never to show their absence [Notes on structured programming, 1972]. In fact, the program does not behave correctly at all times. However, the error only occurs rarely in some specific test sequences, such that testing is not useful for detecting such errors in practice.

1.2. On Formal Verification

For the formal verification of the program, we need the correctness claims specified formally, for instance as follow.

```
(at line 18): ((m*m)<=n)&&((m*m)+2*m+1>n)
```

There are two types of methods for formal verification. One is deduction, and the other is model checking.

Deduction is not suitable in this case, since the program is not correct and deduction is not suitable for detecting errors.

Model checking is the appropriate choice in this case, since it is well suited for both proving correctness and detecting errors of programs.

The weakness of model checking is that it may not be able to handle large programs. For this purpose, model checking and deduction may be combined to form a more powerful verification approach.

2. Model Checking

Model checking is a kind of approaches that use algorithmic methods to determine whether a model satisfies a property. In this case, the model is the program or a similar representation of the potential executions of the program.

For the application of model checking, we need to have a model checking tool. In this case, we use *verds*, a tool for verification of hierarchical discrete systems by symbolic methods.

2.1. Experimental Result

Let isr.c be the file that contains the program (including the functions specified in Appendix A) and isr.sp be the file that contains the correctness claim as specified above.

The command lines for checking the program is as follows.

```
./c2vvm isr.c isr.sp
./verds isr.vvm
```

Then a selected part of the output (of which the full one is presented in Appendix B) of the second command is as follows.

```
VERSION:     verds 1.42 - SEP 2012
FILE:        isr.vvm
PROPERTY:    A G (! (pc = 4 )| (((m * m ){ n )& (((m * m )+ ((2 * m )+ 1 ))> n )))
bound =  0   time = 855
----------   time = 855
bound =  1   time = 855
----------   time = 855
.
.
bound =104   time = 1368
----------   time = 1368
The property is false, preparing files ...
CONCLUSION: FALSE (time=4130)
```

2.2. Error Trace

A selected part of the file “logcex.tmp” that contains the error trace (of which the full one is presented in Appendix C) is as follows.

```
--- STATE 0 ---
n        =0
m        =0
k        =1
pc       =0
--- TRANS 1 ---
--- STATE 1 ---
n        =0
m        =0
k        =1
pc       =1
--- TRANS 2 ---
--- STATE 2 ---
n        =1
m        =0
k        =1
pc       =2
--- TRANS 3 ---
--- STATE 3 ---
n        =1
m        =1
k        =1
pc       =3
--- TRANS 4 ---
```

```
--- STATE 4 ---
n        =1
m        =1
k        =1
pc       =4
.
.
--- STATE 99 ---
n        =18
m        =4
k        =20
pc       =4
--- TRANS 5 ---
--- STATE 100 ---
n        =18
m        =4
k        =20
pc       =0
--- TRANS 1 ---
--- STATE 101 ---
n        =18
m        =4
k        =20
pc       =1
--- TRANS 2 ---
--- STATE 102 ---
n        =4
m        =4
k        =20
pc       =2
--- TRANS 3 ---
--- STATE 103 ---
n        =4
m        =1
k        =20
pc       =3
--- TRANS 4 ---
--- STATE 104 ---
n        =4
m        =1
k        =21
pc       =4
```

By analyzing the content of the file, we discover that the problem occurs when the following sequence is used as input.

```
1 3 5 7 9 11 13 15 17 19 0 2 4 6 8 10 12 14 16 18 4
```

In fact, the possibility for hitting an error is 3 of $(21)^{21}$ with random test cases with length at least 21.

2.3. Correction of the Program

One simple way to correct the error is to modify the main program as follows.

```
#include <stdio.h>
/**************************************************/
int in();
int isr(int x,int k);
int isk(int n,int k);
/**************************************************/
int main()
{
    int n=0;
    int m=0;
    int k=1;

    printf("INFO:   system is now active\n");
    while (1) {
        n=in();
        k=isk(n,k);
        m=isr(n,k);
        printf("RESULT: %i\n\n",m);
    }
}
/**************************************************/
```

This modification swaps the following two statements

```
        m=isr(n,k);
        k=isk(n,k);
```

Suppose that the file containing this modified version of the program is “isr2.c”. Then we verify the program as follows.

```
./c2vvm isr2.c isr.sp
./verds isr2.vvm
```

Then a selected part of the output (of which the full one is presented in Appendix B) of the second command is as follows.

```
VERSION:    verds 1.42 - SEP 2012
FILE:       isr2.vvm
PROPERTY:   A G (! (pc = 4 )| (((m * m ){ n )& (((m * m )+ ((2 * m )+ 1 ))> n )))
bound =  0  time = 732
----------  time = 732
bound =  1  time = 732
----------  time = 732
.
.
bound =104  time = 1233
----------  time = 1233
CONCLUSION: TRUE (time=1233)
```

This verifies the program with respect to the claim.

A. Appendix

The following are the three functions involved in the main program.

```
int isr(int x,int k)
{
        int y1=0;
        int y2=0;
        int y3=0;

        y1=0;
        y2=1;
        y3=1;
        if (x==2||(x>2&&k==20)) x=x-1;
        while (y3<=x) {
                y1=y1+1;
                y2=y2+2;
                y3=y3+y2;
        }
        return y1;
}
/*****************************************************/
int isk(int n,int k)
```

```
{
        if (k!=20) {
                if (k!=n) k=21; else if (k==19) k=0; else k=k+2;
        } else {
                k=21;
        }
        return k;
}
/**************************************************/
int in()
{
        char c=0;
        int k=0;

        while (1) {
                k=0;
                putc('N',stdout); putc(':',stdout); putc(9,stdout);
                c=getc(stdin);
                if (c=='\n') {
                        printf("INFO:   the input must be 1 or 2 digits\n\n");
                        continue;
                }
                if (c<'0'||c>'9') {
                        while (1) {
                                c=getc(stdin);
                                if (c=='\n') break;
                        }
                        printf("INFO:   the input must be 1 or 2 digits\n\n");
                        continue;
                }
                k=c-'0';
                c=getc(stdin);
                if (c=='\n') { return k; }
                if (c<'0'||c>'9') {
                        while (1) {
                                c=getc(stdin);
                                if (c=='\n') break;
                        }
                        printf("INFO:   the input must be 1 or 2 digits\n\n");
                        continue;
                }
                if (k<2) k=k*10+(c-'0');
                else if (k==2&&c=='0') k=20;
                else {
```

```
                        while (1) {
                                c=getc(stdin);
                                if (c=='\n') break;
                    }
                    printf("INFO:   the input number must be in {0,...,20}\n\n ");
                    continue;
                }
                c=getc(stdin);
                if (c!='\n') {
                        while (1) {
                                c=getc(stdin);
                                if (c=='\n') break;
                        }
                        printf("INFO:   the input must be 1 or 2 digits\n\n");
                        continue;
                }
                return k;
        }
}
```

B. Appendix

The full output of the command "./verds isr.vvm" is as follows.

```
VERSION:    verds 1.42 - SEP 2012
FILE:       isr.vvm
PROPERTY:   A G (! (pc = 4 )| (((m * m ){ n )& (((m * m )+ ((2 * m )+ 1 ))> n )))
bound =  0  time = 855
----------  time = 855
bound =  1  time = 855
----------  time = 855
bound =  2  time = 855
----------  time = 855
bound =  3  time = 861
----------  time = 861
bound =  4  time = 863
----------  time = 863
bound =  5  time = 865
----------  time = 865
bound =  6  time = 865
----------  time = 865
bound =  7  time = 865
----------  time = 865
```

```
bound =  8  time = 873
----------  time = 873
bound =  9  time = 876
----------  time = 876
bound = 10  time = 878
----------  time = 878
bound = 11  time = 878
----------  time = 878
bound = 12  time = 878
----------  time = 878
bound = 13  time = 886
----------  time = 886
bound = 14  time = 889
----------  time = 889
bound = 15  time = 891
----------  time = 891
bound = 16  time = 891
----------  time = 891
bound = 17  time = 891
----------  time = 891
bound = 18  time = 901
----------  time = 901
bound = 19  time = 904
----------  time = 904
bound = 20  time = 907
----------  time = 907
bound = 21  time = 907
----------  time = 907
bound = 22  time = 907
----------  time = 907
bound = 23  time = 917
----------  time = 917
bound = 24  time = 920
----------  time = 920
bound = 25  time = 922
----------  time = 922
bound = 26  time = 922
----------  time = 922
bound = 27  time = 922
----------  time = 922
bound = 28  time = 934
----------  time = 934
bound = 29  time = 937
----------  time = 937
```

```
bound = 30  time = 940
----------  time = 940
bound = 31  time = 940
----------  time = 940
bound = 32  time = 940
----------  time = 940
bound = 33  time = 952
----------  time = 952
bound = 34  time = 956
----------  time = 956
bound = 35  time = 959
----------  time = 959
bound = 36  time = 959
----------  time = 959
bound = 37  time = 959
----------  time = 959
bound = 38  time = 972
----------  time = 972
bound = 39  time = 976
----------  time = 976
bound = 40  time = 979
----------  time = 979
bound = 41  time = 979
----------  time = 979
bound = 42  time = 979
----------  time = 979
bound = 43  time = 993
----------  time = 993
bound = 44  time = 997
----------  time = 997
bound = 45  time = 1000
----------  time = 1000
bound = 46  time = 1000
----------  time = 1000
bound = 47  time = 1000
----------  time = 1000
bound = 48  time = 1016
----------  time = 1016
bound = 49  time = 1022
----------  time = 1022
bound = 50  time = 1026
----------  time = 1026
bound = 51  time = 1026
----------  time = 1026
```

```
bound = 52  time = 1026
----------  time = 1026
bound = 53  time = 1042
----------  time = 1042
bound = 54  time = 1046
----------  time = 1046
bound = 55  time = 1050
----------  time = 1050
bound = 56  time = 1050
----------  time = 1050
bound = 57  time = 1050
----------  time = 1050
bound = 58  time = 1067
----------  time = 1067
bound = 59  time = 1073
----------  time = 1073
bound = 60  time = 1077
----------  time = 1077
bound = 61  time = 1077
----------  time = 1077
bound = 62  time = 1077
----------  time = 1077
bound = 63  time = 1095
----------  time = 1095
bound = 64  time = 1100
----------  time = 1100
bound = 65  time = 1104
----------  time = 1104
bound = 66  time = 1104
----------  time = 1104
bound = 67  time = 1104
----------  time = 1104
bound = 68  time = 1123
----------  time = 1123
bound = 69  time = 1129
----------  time = 1129
bound = 70  time = 1133
----------  time = 1133
bound = 71  time = 1133
----------  time = 1133
bound = 72  time = 1133
----------  time = 1133
bound = 73  time = 1153
----------  time = 1153
```

```
bound = 74  time = 1159
----------  time = 1159
bound = 75  time = 1163
----------  time = 1163
bound = 76  time = 1163
----------  time = 1163
bound = 77  time = 1163
----------  time = 1163
bound = 78  time = 1184
----------  time = 1184
bound = 79  time = 1190
----------  time = 1190
bound = 80  time = 1195
----------  time = 1195
bound = 81  time = 1195
----------  time = 1195
bound = 82  time = 1195
----------  time = 1195
bound = 83  time = 1217
----------  time = 1217
bound = 84  time = 1223
----------  time = 1223
bound = 85  time = 1228
----------  time = 1228
bound = 86  time = 1228
----------  time = 1228
bound = 87  time = 1228
----------  time = 1228
bound = 88  time = 1251
----------  time = 1251
bound = 89  time = 1257
----------  time = 1257
bound = 90  time = 1262
----------  time = 1262
bound = 91  time = 1262
----------  time = 1262
bound = 92  time = 1262
----------  time = 1262
bound = 93  time = 1286
----------  time = 1286
bound = 94  time = 1293
----------  time = 1293
bound = 95  time = 1298
----------  time = 1298
```

```
bound = 96  time = 1298
----------  time = 1298
bound = 97  time = 1298
----------  time = 1298
bound = 98  time = 1323
----------  time = 1323
bound = 99  time = 1330
----------  time = 1330
bound =100  time = 1335
----------  time = 1335
bound =101  time = 1335
----------  time = 1335
bound =102  time = 1335
----------  time = 1335
bound =103  time = 1361
----------  time = 1361
bound =104  time = 1368
----------  time = 1368
The property is false, preparing files ...
CONCLUSION: FALSE (time=4130)
```

C. Appendix

The content of the file “logcex.tmp” is as follows.

```
--- STATE 0 ---
n       =0
m       =0
k       =1
pc      =0
--- TRANS 1 ---
--- STATE 1 ---
n       =0
m       =0
k       =1
pc      =1
--- TRANS 2 ---
--- STATE 2 ---
n       =1
m       =0
k       =1
pc      =2
--- TRANS 3 ---
```

```
--- STATE 3 ---
n        =1
m        =1
k        =1
pc       =3
--- TRANS 4 ---
--- STATE 4 ---
n        =1
m        =1
k        =3
pc       =4
--- TRANS 5 ---
--- STATE 5 ---
n        =1
m        =1
k        =3
pc       =0
--- TRANS 1 ---
--- STATE 6 ---
n        =1
m        =1
k        =3
pc       =1
--- TRANS 2 ---
--- STATE 7 ---
n        =3
m        =1
k        =3
pc       =2
--- TRANS 3 ---
--- STATE 8 ---
n        =3
m        =1
k        =3
pc       =3
--- TRANS 4 ---
--- STATE 9 ---
n        =3
m        =1
k        =5
pc       =4
--- TRANS 5 ---
--- STATE 10 ---
n        =3
```

```
m        =1
k        =5
pc       =0
--- TRANS 1 ---
--- STATE 11 ---
n        =3
m        =1
k        =5
pc       =1
--- TRANS 2 ---
--- STATE 12 ---
n        =5
m        =1
k        =5
pc       =2
--- TRANS 3 ---
--- STATE 13 ---
n        =5
m        =2
k        =5
pc       =3
--- TRANS 4 ---
--- STATE 14 ---
n        =5
m        =2
k        =7
pc       =4
--- TRANS 5 ---
--- STATE 15 ---
n        =5
m        =2
k        =7
pc       =0
--- TRANS 1 ---
--- STATE 16 ---
n        =5
m        =2
k        =7
pc       =1
--- TRANS 2 ---
--- STATE 17 ---
n        =7
m        =2
k        =7
```

```
pc        =2
--- TRANS 3 ---
--- STATE 18 ---
n         =7
m         =2
k         =7
pc        =3
--- TRANS 4 ---
--- STATE 19 ---
n         =7
m         =2
k         =9
pc        =4
--- TRANS 5 ---
--- STATE 20 ---
n         =7
m         =2
k         =9
pc        =0
--- TRANS 1 ---
--- STATE 21 ---
n         =7
m         =2
k         =9
pc        =1
--- TRANS 2 ---
--- STATE 22 ---
n         =9
m         =2
k         =9
pc        =2
--- TRANS 3 ---
--- STATE 23 ---
n         =9
m         =3
k         =9
pc        =3
--- TRANS 4 ---
--- STATE 24 ---
n         =9
m         =3
k         =11
pc        =4
--- TRANS 5 ---
```

```
--- STATE 25 ---
n        =9
m        =3
k        =11
pc       =0
--- TRANS 1 ---
--- STATE 26 ---
n        =9
m        =3
k        =11
pc       =1
--- TRANS 2 ---
--- STATE 27 ---
n        =11
m        =3
k        =11
pc       =2
--- TRANS 3 ---
--- STATE 28 ---
n        =11
m        =3
k        =11
pc       =3
--- TRANS 4 ---
--- STATE 29 ---
n        =11
m        =3
k        =13
pc       =4
--- TRANS 5 ---
--- STATE 30 ---
n        =11
m        =3
k        =13
pc       =0
--- TRANS 1 ---
--- STATE 31 ---
n        =11
m        =3
k        =13
pc       =1
--- TRANS 2 ---
--- STATE 32 ---
n        =13
```

```
m        =3
k        =13
pc       =2
--- TRANS 3 ---
--- STATE 33 ---
n        =13
m        =3
k        =13
pc       =3
--- TRANS 4 ---
--- STATE 34 ---
n        =13
m        =3
k        =15
pc       =4
--- TRANS 5 ---
--- STATE 35 ---
n        =13
m        =3
k        =15
pc       =0
--- TRANS 1 ---
--- STATE 36 ---
n        =13
m        =3
k        =15
pc       =1
--- TRANS 2 ---
--- STATE 37 ---
n        =15
m        =3
k        =15
pc       =2
--- TRANS 3 ---
--- STATE 38 ---
n        =15
m        =3
k        =15
pc       =3
--- TRANS 4 ---
--- STATE 39 ---
n        =15
m        =3
k        =17
```

```
pc        =4
--- TRANS 5 ---
--- STATE 40 ---
n         =15
m         =3
k         =17
pc        =0
--- TRANS 1 ---
--- STATE 41 ---
n         =15
m         =3
k         =17
pc        =1
--- TRANS 2 ---
--- STATE 42 ---
n         =17
m         =3
k         =17
pc        =2
--- TRANS 3 ---
--- STATE 43 ---
n         =17
m         =4
k         =17
pc        =3
--- TRANS 4 ---
--- STATE 44 ---
n         =17
m         =4
k         =19
pc        =4
--- TRANS 5 ---
--- STATE 45 ---
n         =17
m         =4
k         =19
pc        =0
--- TRANS 1 ---
--- STATE 46 ---
n         =17
m         =4
k         =19
pc        =1
--- TRANS 2 ---
```

```
--- STATE 47 ---
n        =19
m        =4
k        =19
pc       =2
--- TRANS 3 ---
--- STATE 48 ---
n        =19
m        =4
k        =19
pc       =3
--- TRANS 4 ---
--- STATE 49 ---
n        =19
m        =4
k        =0
pc       =4
--- TRANS 5 ---
--- STATE 50 ---
n        =19
m        =4
k        =0
pc       =0
--- TRANS 1 ---
--- STATE 51 ---
n        =19
m        =4
k        =0
pc       =1
--- TRANS 2 ---
--- STATE 52 ---
n        =0
m        =4
k        =0
pc       =2
--- TRANS 3 ---
--- STATE 53 ---
n        =0
m        =0
k        =0
pc       =3
--- TRANS 4 ---
--- STATE 54 ---
n        =0
```

```
m        =0
k        =2
pc       =4
--- TRANS 5 ---
--- STATE 55 ---
n        =0
m        =0
k        =2
pc       =0
--- TRANS 1 ---
--- STATE 56 ---
n        =0
m        =0
k        =2
pc       =1
--- TRANS 2 ---
--- STATE 57 ---
n        =2
m        =0
k        =2
pc       =2
--- TRANS 3 ---
--- STATE 58 ---
n        =2
m        =1
k        =2
pc       =3
--- TRANS 4 ---
--- STATE 59 ---
n        =2
m        =1
k        =4
pc       =4
--- TRANS 5 ---
--- STATE 60 ---
n        =2
m        =1
k        =4
pc       =0
--- TRANS 1 ---
--- STATE 61 ---
n        =2
m        =1
k        =4
```

```
pc       =1
--- TRANS 2 ---
--- STATE 62 ---
n        =4
m        =1
k        =4
pc       =2
--- TRANS 3 ---
--- STATE 63 ---
n        =4
m        =2
k        =4
pc       =3
--- TRANS 4 ---
--- STATE 64 ---
n        =4
m        =2
k        =6
pc       =4
--- TRANS 5 ---
--- STATE 65 ---
n        =4
m        =2
k        =6
pc       =0
--- TRANS 1 ---
--- STATE 66 ---
n        =4
m        =2
k        =6
pc       =1
--- TRANS 2 ---
--- STATE 67 ---
n        =6
m        =2
k        =6
pc       =2
--- TRANS 3 ---
--- STATE 68 ---
n        =6
m        =2
k        =6
pc       =3
--- TRANS 4 ---
```

```
--- STATE 69 ---
n        =6
m        =2
k        =8
pc       =4
--- TRANS 5 ---
--- STATE 70 ---
n        =6
m        =2
k        =8
pc       =0
--- TRANS 1 ---
--- STATE 71 ---
n        =6
m        =2
k        =8
pc       =1
--- TRANS 2 ---
--- STATE 72 ---
n        =8
m        =2
k        =8
pc       =2
--- TRANS 3 ---
--- STATE 73 ---
n        =8
m        =2
k        =8
pc       =3
--- TRANS 4 ---
--- STATE 74 ---
n        =8
m        =2
k        =10
pc       =4
--- TRANS 5 ---
--- STATE 75 ---
n        =8
m        =2
k        =10
pc       =0
--- TRANS 1 ---
--- STATE 76 ---
n        =8
```

```
m        =2
k        =10
pc       =1
--- TRANS 2 ---
--- STATE 77 ---
n        =10
m        =2
k        =10
pc       =2
--- TRANS 3 ---
--- STATE 78 ---
n        =10
m        =3
k        =10
pc       =3
--- TRANS 4 ---
--- STATE 79 ---
n        =10
m        =3
k        =12
pc       =4
--- TRANS 5 ---
--- STATE 80 ---
n        =10
m        =3
k        =12
pc       =0
--- TRANS 1 ---
--- STATE 81 ---
n        =10
m        =3
k        =12
pc       =1
--- TRANS 2 ---
--- STATE 82 ---
n        =12
m        =3
k        =12
pc       =2
--- TRANS 3 ---
--- STATE 83 ---
n        =12
m        =3
k        =12
```

```
pc        =3
--- TRANS 4 ---
--- STATE 84 ---
n         =12
m         =3
k         =14
pc        =4
--- TRANS 5 ---
--- STATE 85 ---
n         =12
m         =3
k         =14
pc        =0
--- TRANS 1 ---
--- STATE 86 ---
n         =12
m         =3
k         =14
pc        =1
--- TRANS 2 ---
--- STATE 87 ---
n         =14
m         =3
k         =14
pc        =2
--- TRANS 3 ---
--- STATE 88 ---
n         =14
m         =3
k         =14
pc        =3
--- TRANS 4 ---
--- STATE 89 ---
n         =14
m         =3
k         =16
pc        =4
--- TRANS 5 ---
--- STATE 90 ---
n         =14
m         =3
k         =16
pc        =0
--- TRANS 1 ---
```

```
--- STATE 91 ---
n        =14
m        =3
k        =16
pc       =1
--- TRANS 2 ---
--- STATE 92 ---
n        =16
m        =3
k        =16
pc       =2
--- TRANS 3 ---
--- STATE 93 ---
n        =16
m        =4
k        =16
pc       =3
--- TRANS 4 ---
--- STATE 94 ---
n        =16
m        =4
k        =18
pc       =4
--- TRANS 5 ---
--- STATE 95 ---
n        =16
m        =4
k        =18
pc       =0
--- TRANS 1 ---
--- STATE 96 ---
n        =16
m        =4
k        =18
pc       =1
--- TRANS 2 ---
--- STATE 97 ---
n        =18
m        =4
k        =18
pc       =2
--- TRANS 3 ---
--- STATE 98 ---
n        =18
```

```
m        =4
k        =18
pc       =3
--- TRANS 4 ---
--- STATE 99 ---
n        =18
m        =4
k        =20
pc       =4
--- TRANS 5 ---
--- STATE 100 ---
n        =18
m        =4
k        =20
pc       =0
--- TRANS 1 ---
--- STATE 101 ---
n        =18
m        =4
k        =20
pc       =1
--- TRANS 2 ---
--- STATE 102 ---
n        =4
m        =4
k        =20
pc       =2
--- TRANS 3 ---
--- STATE 103 ---
n        =4
m        =1
k        =20
pc       =3
--- TRANS 4 ---
--- STATE 104 ---
n        =4
m        =1
k        =21
pc       =4
```

D. Appendix

The full output of the command “./verds isr2.vvm” is as follows.

```
VERSION:    verds 1.42 - SEP 2012
FILE:       isr2.vvm
PROPERTY:   A G (! (pc = 4 )| (((m * m ){ n )& (((m * m )+ ((2 * m )+ 1 ))> n )))
bound =  0  time = 732
----------  time = 732
bound =  1  time = 732
----------  time = 732
bound =  2  time = 732
----------  time = 732
bound =  3  time = 738
----------  time = 738
bound =  4  time = 740
----------  time = 740
bound =  5  time = 742
----------  time = 742
bound =  6  time = 742
----------  time = 742
bound =  7  time = 742
----------  time = 742
bound =  8  time = 751
----------  time = 751
bound =  9  time = 752
----------  time = 752
bound = 10  time = 755
----------  time = 755
bound = 11  time = 755
----------  time = 755
bound = 12  time = 755
----------  time = 755
bound = 13  time = 763
----------  time = 763
bound = 14  time = 765
----------  time = 765
bound = 15  time = 767
----------  time = 767
bound = 16  time = 767
----------  time = 767
bound = 17  time = 767
----------  time = 767
bound = 18  time = 777
----------  time = 777
bound = 19  time = 779
----------  time = 779
bound = 20  time = 781
```

```
----------  time = 781
bound = 21  time = 781
----------  time = 781
bound = 22  time = 781
----------  time = 781
bound = 23  time = 792
----------  time = 792
bound = 24  time = 794
----------  time = 794
bound = 25  time = 797
----------  time = 797
bound = 26  time = 797
----------  time = 797
bound = 27  time = 797
----------  time = 797
bound = 28  time = 808
----------  time = 808
bound = 29  time = 811
----------  time = 811
bound = 30  time = 814
----------  time = 814
bound = 31  time = 814
----------  time = 814
bound = 32  time = 814
----------  time = 814
bound = 33  time = 826
----------  time = 826
bound = 34  time = 829
----------  time = 829
bound = 35  time = 832
----------  time = 832
bound = 36  time = 832
----------  time = 832
bound = 37  time = 832
----------  time = 832
bound = 38  time = 846
----------  time = 846
bound = 39  time = 849
----------  time = 849
bound = 40  time = 852
----------  time = 852
bound = 41  time = 852
----------  time = 852
bound = 42  time = 852
```

```
----------  time = 852
bound = 43  time = 867
----------  time = 867
bound = 44  time = 870
----------  time = 870
bound = 45  time = 873
----------  time = 873
bound = 46  time = 873
----------  time = 873
bound = 47  time = 873
----------  time = 873
bound = 48  time = 889
----------  time = 889
bound = 49  time = 893
----------  time = 893
bound = 50  time = 897
----------  time = 897
bound = 51  time = 897
----------  time = 897
bound = 52  time = 897
----------  time = 897
bound = 53  time = 914
----------  time = 914
bound = 54  time = 917
----------  time = 917
bound = 55  time = 922
----------  time = 922
bound = 56  time = 922
----------  time = 922
bound = 57  time = 922
----------  time = 922
bound = 58  time = 939
----------  time = 939
bound = 59  time = 943
----------  time = 943
bound = 60  time = 947
----------  time = 947
bound = 61  time = 947
----------  time = 947
bound = 62  time = 947
----------  time = 947
bound = 63  time = 965
----------  time = 965
bound = 64  time = 969
```

```
----------  time = 969
bound = 65  time = 974
----------  time = 974
bound = 66  time = 974
----------  time = 974
bound = 67  time = 974
----------  time = 974
bound = 68  time = 993
----------  time = 993
bound = 69  time = 997
----------  time = 997
bound = 70  time = 1003
----------  time = 1003
bound = 71  time = 1003
----------  time = 1003
bound = 72  time = 1003
----------  time = 1003
bound = 73  time = 1023
----------  time = 1023
bound = 74  time = 1027
----------  time = 1027
bound = 75  time = 1032
----------  time = 1032
bound = 76  time = 1032
----------  time = 1032
bound = 77  time = 1032
----------  time = 1032
bound = 78  time = 1053
----------  time = 1053
bound = 79  time = 1058
----------  time = 1058
bound = 80  time = 1063
----------  time = 1063
bound = 81  time = 1063
----------  time = 1063
bound = 82  time = 1063
----------  time = 1063
bound = 83  time = 1086
----------  time = 1086
bound = 84  time = 1090
----------  time = 1090
bound = 85  time = 1096
----------  time = 1096
bound = 86  time = 1096
```

```
-----------  time = 1096
bound = 87  time = 1096
-----------  time = 1096
bound = 88  time = 1119
-----------  time = 1119
bound = 89  time = 1124
-----------  time = 1124
bound = 90  time = 1130
-----------  time = 1130
bound = 91  time = 1130
-----------  time = 1130
bound = 92  time = 1130
-----------  time = 1130
bound = 93  time = 1154
-----------  time = 1154
bound = 94  time = 1159
-----------  time = 1159
bound = 95  time = 1165
-----------  time = 1165
bound = 96  time = 1165
-----------  time = 1165
bound = 97  time = 1165
-----------  time = 1165
bound = 98  time = 1190
-----------  time = 1190
bound = 99  time = 1196
-----------  time = 1196
bound =100  time = 1202
-----------  time = 1202
bound =101  time = 1202
-----------  time = 1202
bound =102  time = 1202
-----------  time = 1202
bound =103  time = 1228
-----------  time = 1228
bound =104  time = 1233
-----------  time = 1233
CONCLUSION: TRUE (time=1233)
```