Trabalho Prático 2

Grupo 22

Alexis Correia - A102495 João Fonseca - A102512

Enunciado

Considere a descrição da cifra A5/1 que consta no documento +Lógica Computacional: a Cifra A5/1 . Informação complementar pode ser obtida no artigo da Wikipedia.

Pretende-se

1. Definir e codificar, em Z3 e usando o tipo BitVec para modelar a informação, uma FSM que descreva o gerador de chaves.
2. Considere as seguintes eventuais propriedades de erro:
   a. ocorrência de um "burst" 0^t (t zeros) que ocorre em 2^t passos ou menos.
   b. ocorrência de um "burst" de tamanho t que repete um "burst" anterior no mesmo output em $2^{t/2}$ passos ou menos.

   Tente codificar estas propriedades e verificar se são acessíveis a partir de um estado inicial aleatoriamente gerado.

Resolução

Formalmente uma FSM é um triplo $\Sigma \equiv \langle Q, I, \delta \rangle$ em que:

1. Q é o conjunto (finito) de estados;
2. I é o conjunto de estados iniciais, logo $I \in Q$;
3. δ é uma relação binária $Q \times Q$ designida **relação de transição**;

Neste caso em concreto, o númeor de estados (a cardinalidade de Q) é 64, pois esse é o número de ciclos no gerador de chaves da cifra A5/1. Então, começaremos com as funções `declare`, `init` e `trans`.

```
from pysmt.shortcuts import *
from pysmt.typing import BVType
import random

# Constantes
## Tamanho de cada LFSR
size0 = 19
size1 = 22
size2 = 23

## Posições de bits de controle de cada LSFR
controlBit0 = 8
```

```
controlBit1 = 10
controlBit2 = 10

## Constantes de transição
s0 = BV("11100100000000000000", size0)
s1 = BV("11000000000000000000000", size1)
s2 = BV("11100000000000010000000", size2)

def declare(i):
    s = {}
    s['lfsr0'] = Symbol('lfsr0_e'+str(i), BVType(size0))
    s['lfsr1'] = Symbol('lfsr1_e'+str(i), BVType(size1))
    s['lfsr2'] = Symbol('lfsr2_e'+str(i), BVType(size2))
    return s

def init(state): # Chave da cifra (aleatório)
    r0 = random.getrandbits(size0)
    A = Equals(state['lfsr0'],BV(r0, size0))

    r1 = random.getrandbits(size1)
    B = Equals(state['lfsr1'],BV(r1, size1))

    r2 = random.getrandbits(size2)
    C = Equals(state['lfsr2'],BV(r2, size2))
    return And(A,B,C)

def cBit(state):
    c0 = BVExtract(state['lfsr0'], controlBit0, controlBit0)
    c1 = BVExtract(state['lfsr1'], controlBit1, controlBit1)
    c2 = BVExtract(state['lfsr2'], controlBit2, controlBit2)
    if ((c0 & c1) | (c1 & c2) | (c0 & c2)):
        r = BV(1,1)
    else:
        r = BV(0,1)
    return r

def trans(curr,prox):
    c = cBit(curr)
    t0 = And(Equals(BVExtract(curr['lfsr0'],controlBit0,
controlBit0),c),
             Equals(prox['lfsr0'],
BVXor(BVLShl(curr['lfsr0'],1),BVXor(curr['lfsr0'],s0))))
    t1 = And(Equals(BVExtract(curr['lfsr1'],controlBit1,
controlBit1),c),
             Equals(prox['lfsr1'],
BVXor(BVLShl(curr['lfsr1'],1),BVXor(curr['lfsr1'],s1))))
    t2 = And(Equals(BVExtract(curr['lfsr2'],controlBit2,
controlBit2),c),
             Equals(prox['lfsr2'],
BVXor(BVLShl(curr['lfsr2'],1),BVXor(curr['lfsr2'],s2))))
```

```
    return Or(And(t0,t1), And(t0,t2), And(t1,t2), And(t0,t1,t2))
#Or(t0, t1, t2)#
```

Para a geração de um estado inicial aleatório (conforme o enunciado requisita), utilizamos a biblioteca `random` e a função `random.getrandbits(n)` em que n é o número de bits em cada **LFSR**. Além disso, criamos uma função auxiliar à `trans` denominada `cBit` que calcula o bit majoritario dentre os três bits de controlo. Essa função permite selecionar quais **LFSR** que mudarão entre o dois estados (`curr` e `prox`).

Agora podemos partir para a função `genTrace` que vai de facto criar os 64 estados e escrever(imprimir) os valores de cada **LFSR** em cada estado.

```
def genTrace(declare,init,trans): # k = 64
    states = [declare(i) for i in range(64)]
    solver = Solver(name = "z3")
    solver.add_assertion(init(states[0]))
    for i in range(63):
        solver.add_assertion(trans(states[i], states[i+1]))
    if solver.solve():
        for i,s in enumerate(states):
            r0 = format(solver.get_value(s['lfsr0']).constant_value(),
f'0{size0}b')
            r1 = format(solver.get_value(s['lfsr0']).constant_value(),
f'0{size1}b')
            r2 = format(solver.get_value(s['lfsr0']).constant_value(),
f'0{size2}b')
            print(f"Estado {i}\n lfsr0:{r0} lfsr1:{r1} lfsr2:{r2}")
        pass
    else:
        print("> Not feasible.")
    #return states

genTrace(declare, init, trans)
```

```
Estado 0
 lfsr0:0010000111101010111 lfsr1:00000100001111010101 11
lfsr2:00000100001111010101 11
Estado 1
 lfsr0:1000011000111111001 lfsr1:00010000110001111111001
lfsr2:00001000011000111111001
Estado 2
 lfsr0:0011101101100101101 lfsr1:00000111011011001011 01
lfsr2:00000111011011001011 01
Estado 3
 lfsr0:1010100110101110111 lfsr1:00010101001101011101 11
lfsr2:00010101001101011101 11
Estado 4
```

```
 lfsr0:0001111011110011001 lfsr1:00000011110111110011001
lfsr2:00000001111011110011001
Estado 5
 lfsr0:1100011100010101011 lfsr1:00011000111000101010111
lfsr2:00001100011100010101011
Estado 6
 lfsr0:1010110100111111101 lfsr1:00010101101001111111101
lfsr2:00001010110100111111101
Estado 7
 lfsr0:0001001101000000111 lfsr1:00000010011010000000111
lfsr2:00000001001101000000111
Estado 8
 lfsr0:0101010000110001000 lfsr1:00001010100001100010000
lfsr2:00000101010000110001000
Estado 9
 lfsr0:0001100001010011000 lfsr1:00000011000010100110000
lfsr2:00000001100001010011000
Estado 10
 lfsr0:0101001111100000000 lfsr1:00001010011111000000000
lfsr2:00000101001111100000000
Estado 11
 lfsr0:0001000000100000000 lfsr1:00000010000001000000000
lfsr2:00000001000000100000000
Estado 12
 lfsr0:1101010001100000000 lfsr1:00011010100011000000000
lfsr2:00001101010001100000000
Estado 13
 lfsr0:1001100010100000000 lfsr1:00010011000101000000000
lfsr2:00001001100010100000000
Estado 14
 lfsr0:0100110111100000000 lfsr1:00001001101111000000000
lfsr2:00000100110111100000000
Estado 15
 lfsr0:0011001000100000000 lfsr1:00000110010001000000000
lfsr2:00000011001000100000000
Estado 16
 lfsr0:1000010001000001010 lfsr1:00010000100010000001010
lfsr2:00001000010001000001010
Estado 17
 lfsr0:1110000111100000000 lfsr1:00011100001111000000000
lfsr2:00001110000111100000000
Estado 18
 lfsr0:1100011000100000000 lfsr1:00011000110001000000000
lfsr2:00001100011000100000000
Estado 19
 lfsr0:1010111001100000000 lfsr1:00010101110011000000000
lfsr2:00001010111001100000000
Estado 20
 lfsr0:0001011010100000000 lfsr1:00000010110101000000000
```

```
lfsr2:000000010110101000000000
Estado 21
 lfsr0:1101111111100000000 lfsr1:0001101111111100000000
lfsr2:000011011111111000000000
Estado 22
 lfsr0:1000010000100000000 lfsr1:0001000010000100000000
lfsr2:000010000100001000000000
Estado 23
 lfsr0:0110100001100000000 lfsr1:0000110100001100000000
lfsr2:000001101000011000000000
Estado 24
 lfsr0:0101110010100000000 lfsr1:0000101110010100000000
lfsr2:000001011100101000000000
Estado 25
 lfsr0:0000000111100000000 lfsr1:0000000000111100000000
lfsr2:000000000000111100000000
Estado 26
 lfsr0:1110011000100000000 lfsr1:0001110011000100000000
lfsr2:000011100110001000000000
Estado 27
 lfsr0:1100111001100000000 lfsr1:0001100111001100000000
lfsr2:000011001110011000000000
Estado 28
 lfsr0:1011011010100000000 lfsr1:0001011011010100000000
lfsr2:000010110110101000000000
Estado 29
 lfsr0:0011111111100000000 lfsr1:0000111111111100000000
lfsr2:000001111111111000000000
Estado 30
 lfsr0:1010010000100000000 lfsr1:0001010010000100000000
lfsr2:000010100100001000000000
Estado 31
 lfsr0:0000100001100000000 lfsr1:0000000100001100000000
lfsr2:000000010000110000000000
Estado 32
 lfsr0:1111110010100000000 lfsr1:0001111110010100000000
lfsr2:000011111100101000000000
Estado 33
 lfsr0:1110000111100000000 lfsr1:0001110000111100000000
lfsr2:000011100001111000000000
Estado 34
 lfsr0:1100011000100000000 lfsr1:0001100011000100000000
lfsr2:000011000110001000000000
Estado 35
 lfsr0:1010111001100000000 lfsr1:0001010111001100000000
lfsr2:000010101110011000000000
Estado 36
 lfsr0:0001011010100000000 lfsr1:0000001011010100000000
lfsr2:000000010110101000000000
```

```
Estado 37
 lfsr0:1101111111100000000 lfsr1:00011011111111100000000
lfsr2:00001101111111100000000
Estado 38
 lfsr0:1000010000100000000 lfsr1:00010000100001000000000
lfsr2:00001000010000100000000
Estado 39
 lfsr0:1100001000100000000 lfsr1:00011000010001000000000
lfsr2:00001100001000100000000
Estado 40
 lfsr0:1010001001100000000 lfsr1:00010100010011000000000
lfsr2:00001010001001100000000
Estado 41
 lfsr0:0000001010100000000 lfsr1:00000000010101000000000
lfsr2:00000000001010100000000
Estado 42
 lfsr0:1110001111100000000 lfsr1:00011100011111000000000
lfsr2:00001110001111100000000
Estado 43
 lfsr0:1100000000100000000 lfsr1:00011000000001000000000
lfsr2:00001100000000100000000
Estado 44
 lfsr0:1010010001100000000 lfsr1:00010100100011000000000
lfsr2:00001010010001100000000
Estado 45
 lfsr0:0000100010100000000 lfsr1:00000001000101000000000
lfsr2:00000000100010100000000
Estado 46
 lfsr0:1111110111100000000 lfsr1:00011111101111000000000
lfsr2:00001111110111100000000
Estado 47
 lfsr0:1110001000100000000 lfsr1:00011100010001000000000
lfsr2:00001110001000100000000
Estado 48
 lfsr0:1100001001100000000 lfsr1:00011000010011000000000
lfsr2:00001100001001100000000
Estado 49
 lfsr0:1010001010100000000 lfsr1:00010100010101000000000
lfsr2:00001010001010100000000
Estado 50
 lfsr0:0000001111100000000 lfsr1:00000000011111000000000
lfsr2:00000000001111100000000
Estado 51
 lfsr0:1110000000100000000 lfsr1:00011100000001000000000
lfsr2:00001110000000100000000
Estado 52
 lfsr0:1100010001100000000 lfsr1:00011000100011000000000
lfsr2:00001100010001100000000
Estado 53
```

```
 lfsr0:1010100010100000000 lfsr1:0001010100010100000000
lfsr2:0000101010001010000000
Estado 54
 lfsr0:0001110111100000000 lfsr1:0000001110111100000000
lfsr2:0000000111011110000000
Estado 55
 lfsr0:1100001000100000000 lfsr1:0001100001000100000000
lfsr2:0000110000100010000000
Estado 56
 lfsr0:1010001001100000000 lfsr1:0001010001001100000000
lfsr2:0000101000100110000000
Estado 57
 lfsr0:0000001010100000000 lfsr1:0000000001010100000000
lfsr2:0000000000101010000000
Estado 58
 lfsr0:1110001111100000000 lfsr1:0001110001111100000000
lfsr2:0000111000111110000000
Estado 59
 lfsr0:1100000000100000000 lfsr1:0001100000000100000000
lfsr2:0000110000000010000000
Estado 60
 lfsr0:1010010001100000000 lfsr1:0001010010001100000000
lfsr2:0000101001000110000000
Estado 61
 lfsr0:0000100010100000000 lfsr1:0000000100010100000000
lfsr2:0000000010001010000000
Estado 62
 lfsr0:1111110111100000000 lfsr1:0001111110111100000000
lfsr2:0000111111011110000000
Estado 63
 lfsr0:1110001000100000000 lfsr1:0001110001000100000000
lfsr2:0000111000100010000000
```