using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

//Ramin Edjlal.CopyRight 2014.AllRightReserved.

namespace LearningMachine

{

[Serializable]

public class Bit

{

public bool[] Bits = new bool[2];

public Bit()

{

Object o = new Object();

lock (o)

{

Bits[0] = false;

Bits[1] = false;

}

}

public bool[] GetBits()

{

Object o = new Object();

lock (o)

{

return Bits;

}

}

public void SetZeroZero()//State 0

{

Object o = new Object();

lock (o)

{

Bits[0] = false;

Bits[1] = false;

}

}

public void SetZeroOne()//State 1

{

Object o = new Object();

lock (o)

{

Bits[0] = true;

Bits[1] = false;

}

}

public void SetOneZero()//State SuperPosition

{

Object o = new Object();

lock (o)

{

Bits[0] = false;

Bits[1] = true;

}

}

public void SetOneOne()

{

Object o = new Object();

lock (o)

{

Bits[0] = true;

Bits[1] = true;

}

}

public bool IsZeroZero()

{

Object o = new Object();

lock (o)

{

if (Bits[0] == false && Bits[1] == false)

return true;

return false;

}

}

public bool IsZeroOne()

{

Object o = new Object();

lock (o)

{

if (Bits[0] == true && Bits[1] == false)

return true;

return false;

}

}

public bool IsOneZero()

{

Object o = new Object();

lock (o)

{

if (Bits[0] == false && Bits[1] == true)

return true;

return false;

}

}

public bool IsOneOne()

{

Object o = new Object();

lock (o)

{

if (Bits[0] == true && Bits[1] == true)

return true;

return false;

}

}

};

[Serializable]

public class QuantumAtamata : LearningKrinskyAtamata

{

List<String> States = new List<String>();

List<Byte> StateByte = new List<Byte>();

int r = 0, m = 0, k = 0;

public Bit[] BitState = new Bit[3];

double[] QuatumProbabilities = new double[3];

public LearningKrinskyAtamata[] ThreeSet = new LearningKrinskyAtamata[3];

public int NumberActiveAtamata = 3;

public double[] FirstProbibility = null;

public double[] SecondProbibility = null;

public double[] ThirdProbibility = null;

int A1 = 0;

int A2 = 0;

int A3 = 0;

public String AA = "";

public String AB = "";

public String AC = "";

public String CurrentState = "";

public QuantumAtamata(int r0, int m0, int k0)

: base(r0, m0, k0)

{

Object o = new Object();

lock (o)

{

for (int i = 0; i < 3; i++)

{

BitState[i] = new Bit();

ThreeSet[i] = new LearningKrinskyAtamata(r0, m0, k0);

}

States.Clear();

r = r0;

m = m0;

k = k0;

FirstProbibility = new double[r];

SecondProbibility = new double[r];

ThirdProbibility = new double[r];

}

}

public void CurrenStateInitialize()

{

Object o = new Object();

lock (o)

{

A1 = FirstAtamataState();

A2 = SecondAtamataState();

A3 = ThirdAtamataState();

AA = A1.ToString();

AB = A2.ToString();

AC = A3.ToString();

if (A1 == 0)

AA = "|0>,";

else

if (A1 == 1)

AA = "|1>,";

else

if (A1 == 2)

AA = "|2>+|3>,";

if (A2 == 0)

AB = "|0>,";

else

if (A2 == 1)

AB = "|1>,";

else

if (A2 == 2)

AB = "|2>+|3>,";

if (A3 == 0)

AC = "|0>,";

else

if (A3 == 1)

AC = "|1>,";

else

if (A3 == 2)

AC = "|2>+|3>,";

CurrentState = AA + AB + AC;

// CurrentStateByte = System.Convert.ToByte(CurrentState, 2);

States.Add(CurrentState);

// StateByte.Add(CurrentStateByte);

if (A1 == 2)

{

if (A2 == 2)

{

if (A3 == 2)

{

NumberActiveAtamata = 1;

for (int i = 0; i < r; i++)

{

FirstProbibility[i] = (ThreeSet[0].alpha[i] + ThreeSet[1].alpha[i] + ThreeSet[2].alpha[i]) / 3.0;

}

}

else

{

NumberActiveAtamata = 2;

for (int i = 0; i < r; i++)

{

FirstProbibility[i] = (ThreeSet[0].alpha[i] + ThreeSet[1].alpha[i]) / 2.0;

SecondProbibility[i] = ThreeSet[2].alpha[i];

}

}

}

else

{

if (A3 == 2)

{

NumberActiveAtamata = 2;

for (int i = 0; i < r; i++)

{

FirstProbibility[i] = (ThreeSet[0].alpha[i] + ThreeSet[2].alpha[i]) / 2.0;

SecondProbibility[i] = ThreeSet[1].alpha[i];

}

}

else

{

NumberActiveAtamata = 3;

for (int i = 0; i < r; i++)

{

FirstProbibility[i] = ThreeSet[0].alpha[i];

SecondProbibility[i] = ThreeSet[1].alpha[i];

ThirdProbibility[i] = ThreeSet[2].alpha[i];

}

}

}

}

else

{

if (A2 == 2)

{

if (A3 == 2)

{

NumberActiveAtamata = 2;

for (int i = 0; i < r; i++)

{

FirstProbibility[i] = (ThreeSet[1].alpha[i] + ThreeSet[2].alpha[i]) / 2.0;

SecondProbibility[i] = ThreeSet[0].alpha[i];

}

}

else

{

NumberActiveAtamata = 3;

for (int i = 0; i < r; i++)

{

FirstProbibility[i] = ThreeSet[1].alpha[i];

SecondProbibility[i] = ThreeSet[0].alpha[i];

ThirdProbibility[i] = ThreeSet[2].alpha[i];

}

}

}

else

if (A3 == 2)

{

NumberActiveAtamata = 3;

for (int i = 0; i < r; i++)

{

FirstProbibility[i] = ThreeSet[2].alpha[i];

SecondProbibility[i] = ThreeSet[0].alpha[i];

ThirdProbibility[i] = ThreeSet[1].alpha[i];

}

}

else

{

NumberActiveAtamata = 3;

for (int i = 0; i < r; i++)

{

FirstProbibility[i] = ThreeSet[0].alpha[i];

SecondProbibility[i] = ThreeSet[2].alpha[i];

ThirdProbibility[i] = ThreeSet[1].alpha[i];

}

}

}

}

}

public int FirstAtamataState()

{ Object o = new Object();

lock (o)

{

if (BitState[0].IsZeroZero())

{

// BitState[0].SetZeroZero();

return 0;//0 State

}

else

if (BitState[0].IsZeroOne())

{

// BitState[0].SetZeroOne();

return 1;//1 State

}

// BitState[0].SetOneZero();

return 2;//SuperPosition State

}

}

public int SecondAtamataState()

{ Object o = new Object();

lock (o)

{

if (BitState[1].IsZeroZero())

{

//BitState[1].SetZeroZero();

return 0;//0 State

}

else

if (BitState[1].IsZeroOne())

{

// BitState[1].SetZeroOne();

return 1;//1 State

}

// BitState[1].SetOneZero();

return 2;//SuperPosition State

}

}

public int ThirdAtamataState()

{ Object o = new Object();

lock (o)

{

if (BitState[2].IsZeroZero())

{

// BitState[2].SetZeroZero();

return 0;//0 State

}

else

if (BitState[2].IsZeroOne())

{

// BitState[2].SetZeroOne();

return 1;//1 State

}

// BitState[2].SetOneZero();

return 2;//SuperPosition State

}

}

}

}