

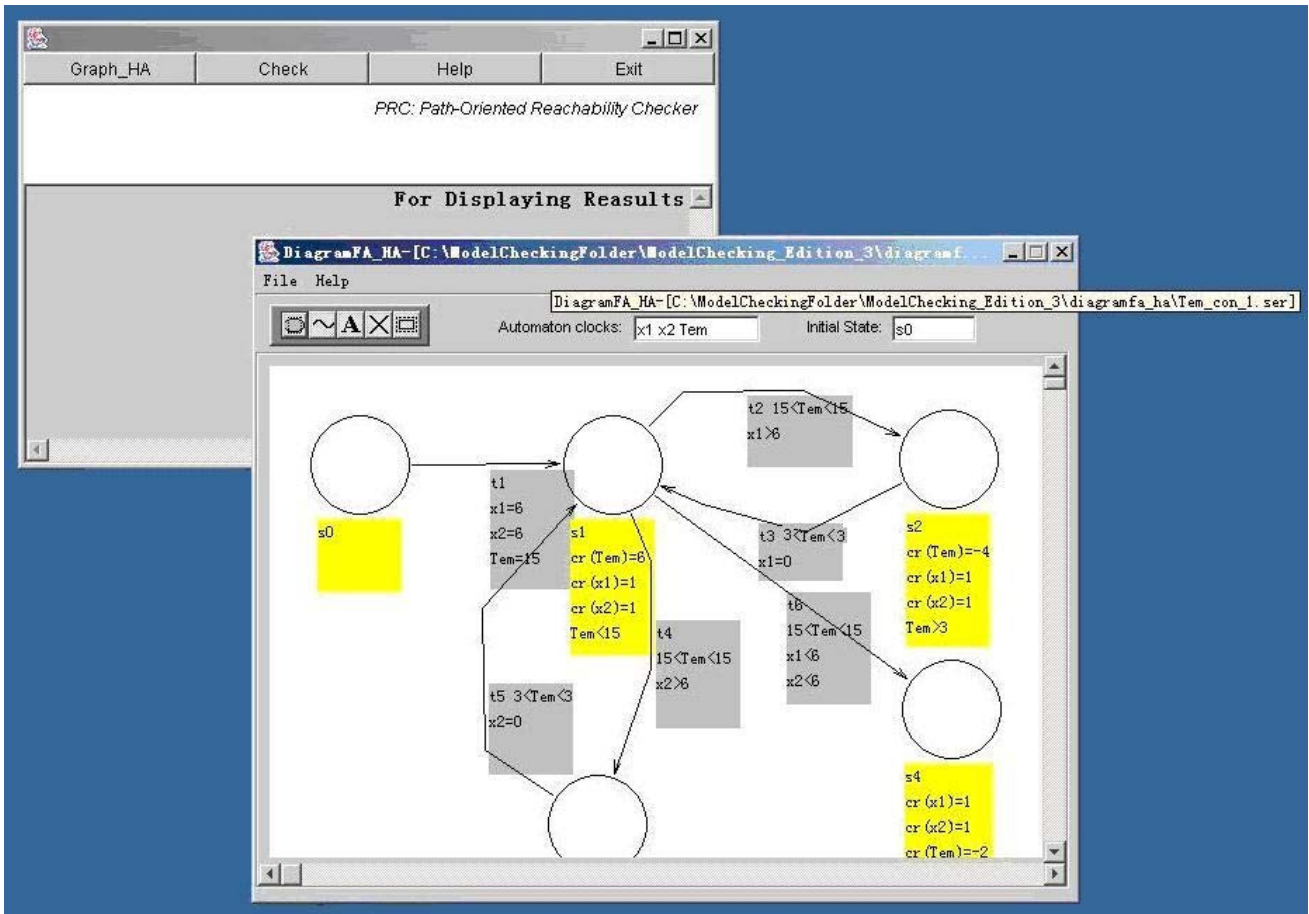
# PRC 1.0 UserManual

PRC 1.0 UserManual.....	1
<b>1</b> Introduction.....	1
<b>2</b> Graph_HA.....	2
<b>2.1</b> Title.....	3
<b>2.2</b> File Menu.....	3
<b>2.3</b> Introduction of the Bar.....	3
<b>2.4</b> Hybrid Automata Clock.....	5
<b>2.5</b> Initial State.....	5
<b>2.6</b> State.....	5
<b>2.7</b> Transition.....	6
<b>3</b> How to create Linear Hybrid Automata.....	6
<b>3.1</b> Example.....	6
<b>3.2</b> Click The Button Graph_HA.....	8
<b>3.3</b> Set the clocks and set the initial state.....	9
<b>3.4</b> Draw the States.....	9
3.4.1 Add the initial State s1.....	10
3.4.2 Add all the states.....	10
<b>3.5</b> Draw the transitions.....	11
3.5.1 Add the first transition t1.....	11
3.5.2 Add all the transition.....	13
<b>3.6</b> Save the Graphical File.....	13
<b>3.7</b> Open The Graphical File.....	14
<b>3.8</b> Make The HA File That Can Be Checked.....	15
<b>4</b> Checker Part.....	16
<b>4.1</b> Select HA File To Check.....	16
<b>4.2</b> Input the path you want to check.....	17
<b>4.3</b> Press OK To Start Check.....	18
<b>5</b> Runtime environment.....	18

## 1 Introduction

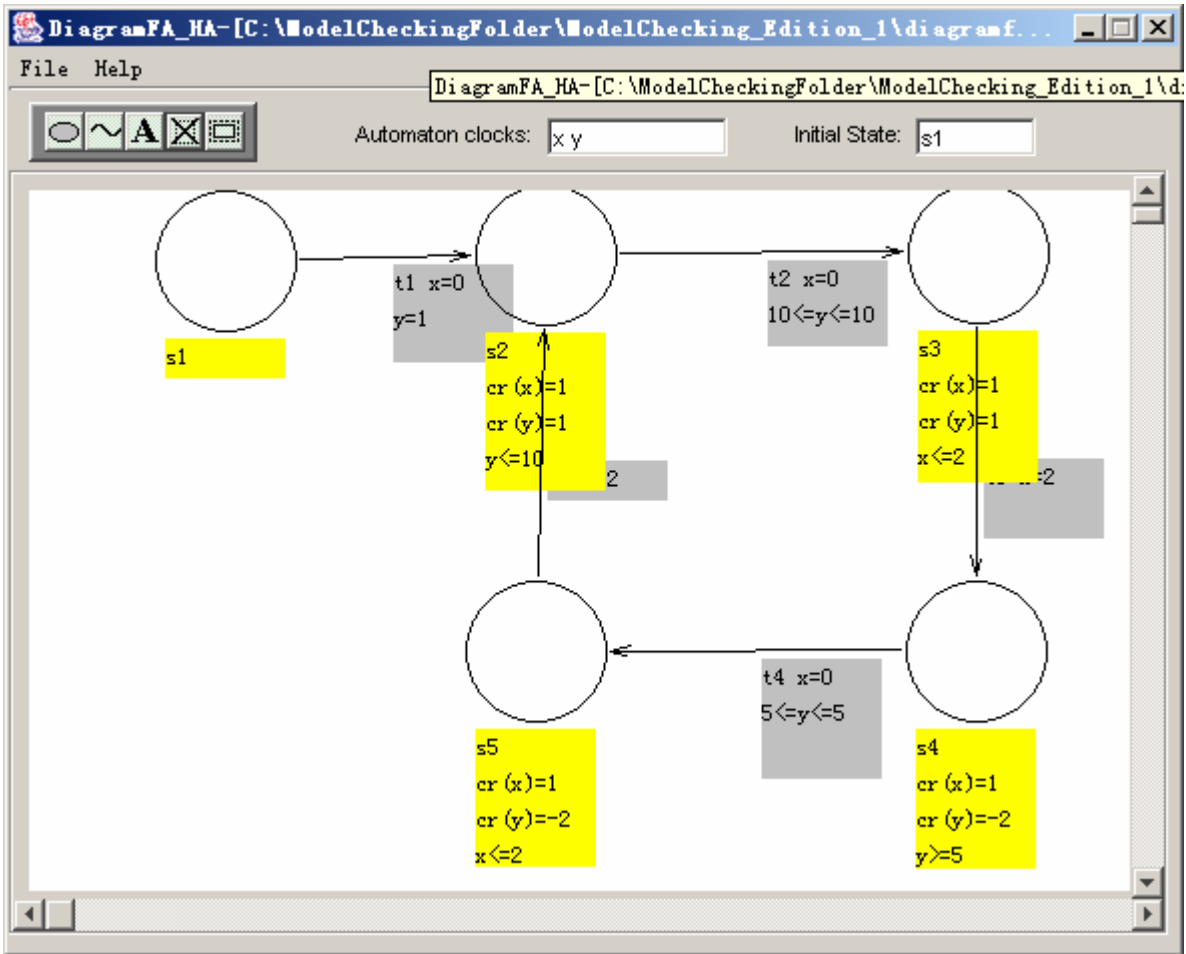
Based on linear programming we develop the techniques towards an efficient path-oriented tool for the bounded reachability analysis of linear hybrid automata. We call this tool PRC stands for Path-Oriented Reachability Checker.

It has two component, one is the Graph\_HA editor and the second is the Checker, just like the figure below



## 2 Graph\_HA

Graph\_ha is the graphical editor for Linear Hybrid Automata



### 2.1 Title



This is the absolute path for this automata

### 2.2 File Menu



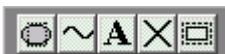
New Make a new Hybrid automata graphical file (HA) .

Open Open and edit a hybrid automata graphical file (HA) .

Save Save the graphical file (HA) with the postfix “.ser”, as “water.ser”

Create HA This tool will do some necessary check of the graphical hybrid automata(as water.ser), then save it in the form of text files with postfix of “.ha”, as water.ha.

### 2.3 Introduction of the Bar



Click on these buttons to do your orders



Draw the states, and adjust the position of the state.

In this tool, circle means the state of the hybrid automata, it has two componet, one is the circle itself, the other is the text attached to this circle.

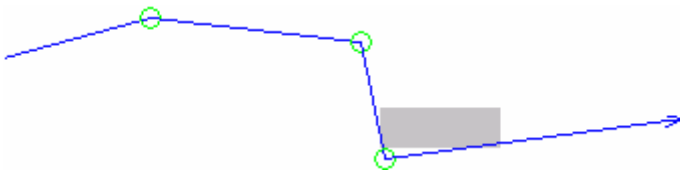
- Click on the blank place to draw a circle.
- Click mouse's left button in the circle to drag the circle.
- You can edit the state info in the yellow text area attached to the circle.



Draw the transitions, and adjust the position of the turning point

In this tool, polyline means the transition of the automata, it has two componet, one is the lineelf, the other is the text attached to this line

- Click mouse's left button in one state to choose the from state, and click on the blank place can set the turning point, you can keep doing this until you click another state(it can be the from state, this is a loop ) this line between two state is a transition.
- In any position of the line click mouse's left button and drag you can make a new turning point
- In one of the line's end(the green circle) click mouse's left button and drag can change this point's position and combine two line to one
- You can edit the state info in the yellow text area attached to the circle.



Adjust the size of text area. Move your mouse pointer to the edge of text area, it will become a horizontal or vertical arrowhead, at this time drag your mouse with left button pressed you can change the size of the text area

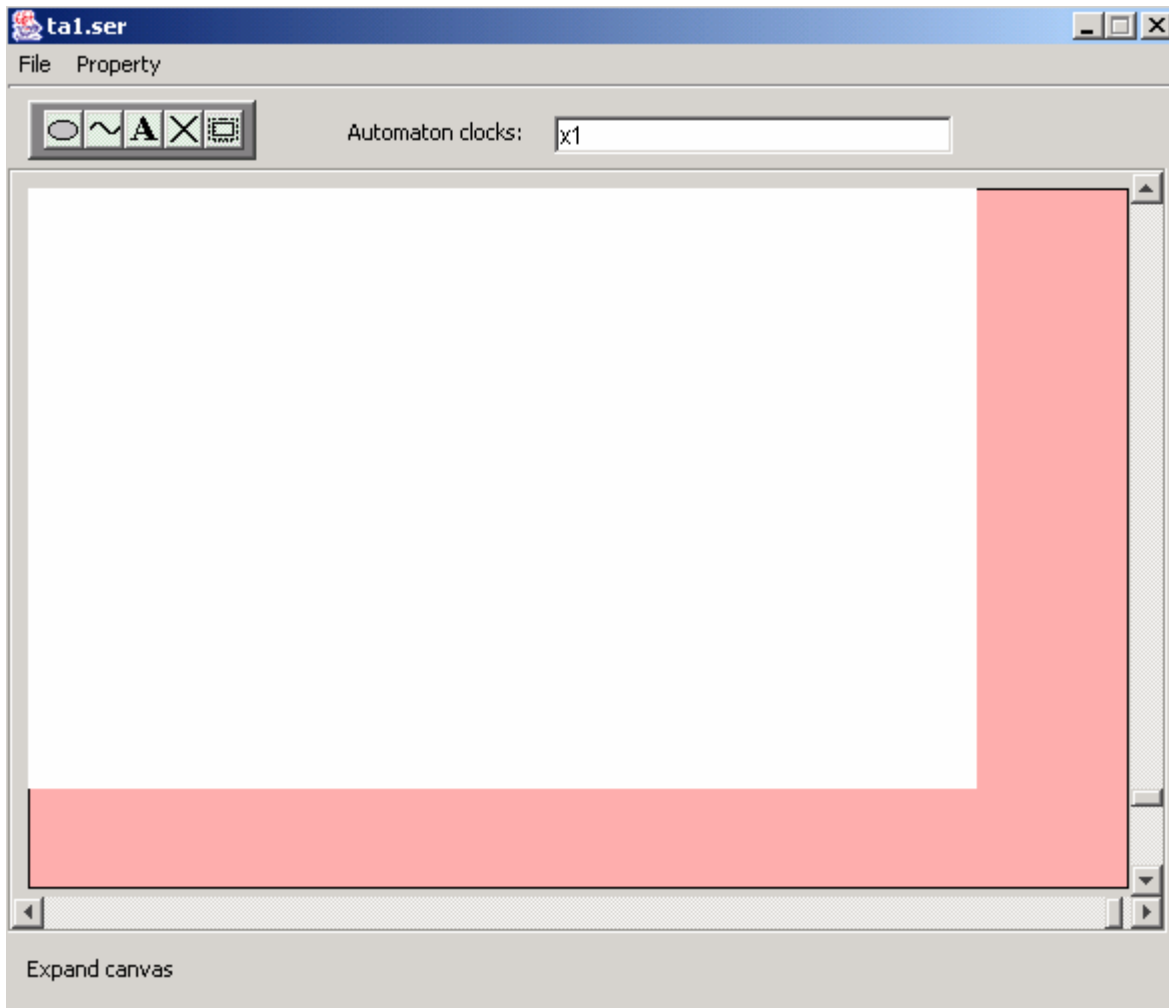


Delete button

Click your mouse's left button on the state or transition you want to delete.



Adjust the size of graphical area(the white area in the pink ones), similar to the button of **A**



#### 2.4 Hybrid Automata Clock

Automaton clocks:

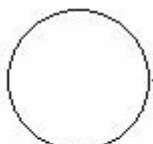
Set your clock here , use ',' to compart them

#### 2.5 Initial State

Initial State:

Set your initial state here. It should be unique.

#### 2.6 State



s5  
 cr (x)=1  
 cr (y)=-2  
 x<=2

State is consist of state name, change rate of every clock and several state constraint. We have to use space bar to compart all the syntax

elements.

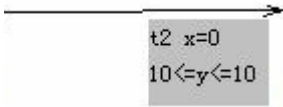
State name eg: s1

Constraint eg :  $x \leq 2$

Change rate of clocks eg :  $cr(x)=1$ , cr means change rate here.

Note: Except the initial state, each state should have a change rate for every clock.

## 2.7 Transition



Transition is consist of transition name, reset of clock and several constraint. We have to use space bar to compart all the syntax elements:

Transition name eg : t2

Reset of clock eg :  $x=0$

Constraint eg :  $10 \leq y \leq 10$

(The equal operator "=" and assignment operator ":" means the same in this tool , so if you want to add a constraint of  $y=10$ , you should write it as  $10 \leq y \leq 10$ )

Note: Each clock should be initialized in the first transition and no transition should aim to the initial state

## 3 How to create Linear Hybrid Automata

### 3.1 Example

This section is the ha files we create by this tool from the sample in the first chapter:

ModelChecking

[VariableInformation]

TotalNumber=2

x:notShared

y:notShared

[StateInformation]

TotalNumber=5

s1:

isInitial=true

isFinal=false

ConstraintNumber=0

changeRateNumber=0

s2:

isInitial=false

isFinal=false

ConstraintNumber=1

-Infinity <= VariableVector <= 10.0

ParameterVariableNumber=1

```

        1.0,y
changeRateNumber=2
    d(x)=[1.0,1.0]
    d(y)=[1.0,1.0]
s3:
isInitial=false
isFinal=false
ConstraintNumber=1
    -Infinity<= VariableVector <=2.0
        ParameterVariableNumber=1
            1.0,x
changeRateNumber=2
    d(x)=[1.0,1.0]
    d(y)=[1.0,1.0]
s5:
isInitial=false
isFinal=false
ConstraintNumber=1
    -Infinity<= VariableVector <=2.0
        ParameterVariableNumber=1
            1.0,x
changeRateNumber=2
    d(x)=[1.0,1.0]
    d(y)=[-2.0,-2.0]
s4:
isInitial=false
isFinal=false
ConstraintNumber=1
    5.0<= VariableVector <=Infinity
        ParameterVariableNumber=1
            1.0,y
changeRateNumber=2
    d(x)=[1.0,1.0]
    d(y)=[-2.0,-2.0]

```

[TransitionInformation]

TotalNumber=5

t1:

s1->s2

ConstraintNumber=0

ResetNumber=2

x:=0.0

y:=1.0

t2:

s2->s3

ConstraintNumber=1

10.0<= VariableVector <=10.0

ParameterVariableNumber=1

1.0,y

ResetNumber=1

x:=0.0

t3:

s3->s4

ConstraintNumber=0

ResetNumber=1

x:=2.0

t4:

s4->s5

ConstraintNumber=1

5.0<= VariableVector <=5.0

ParameterVariableNumber=1

1.0,y

ResetNumber=1

x:=0.0

t5:

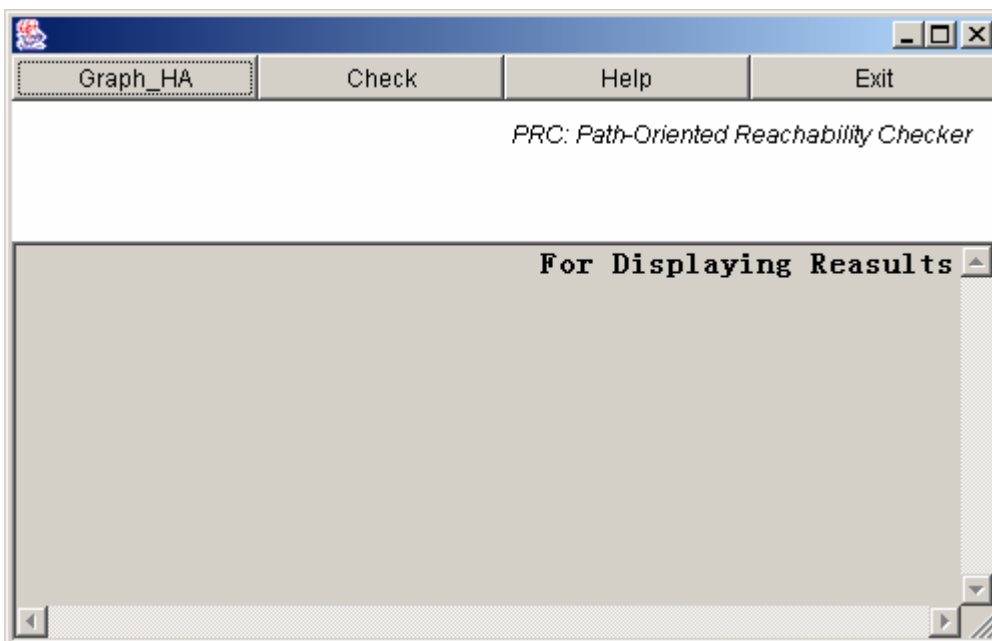
s5->s2

ConstraintNumber=0

ResetNumber=1

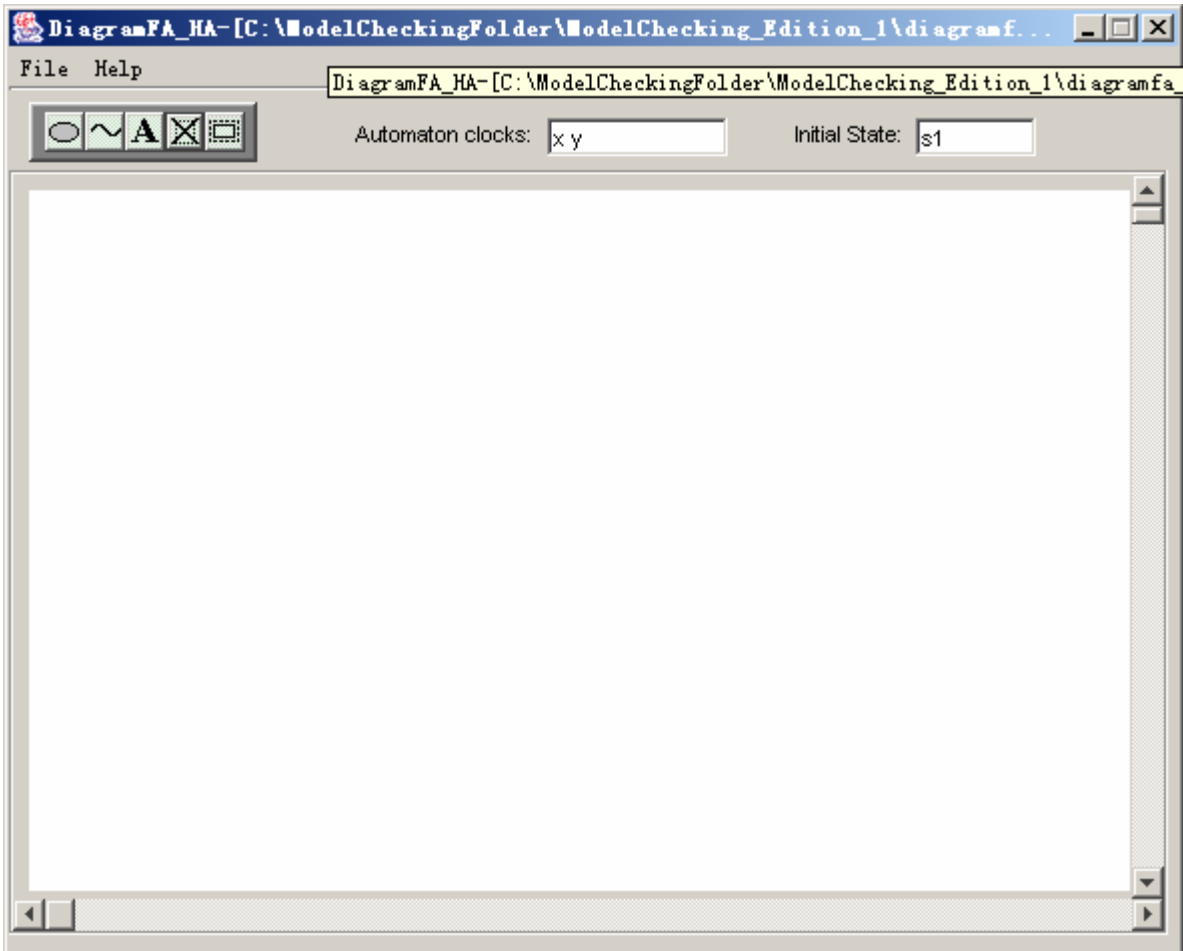
x:=2.0

### 3.2 Click The Button Graph\_HA



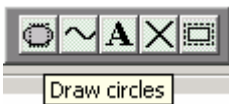


### 3.3 Set the clocks and set the initial state

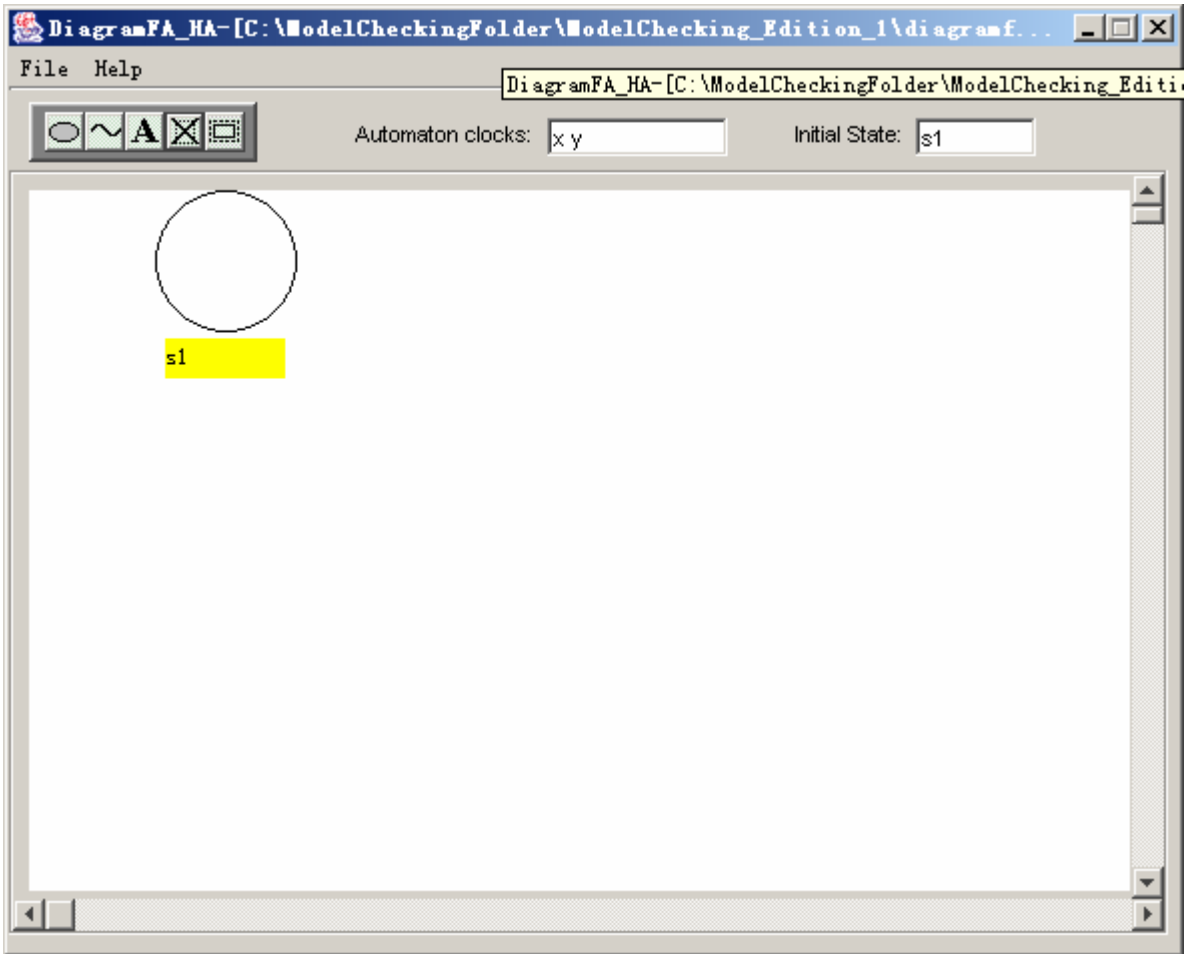


### 3.4 Draw the States

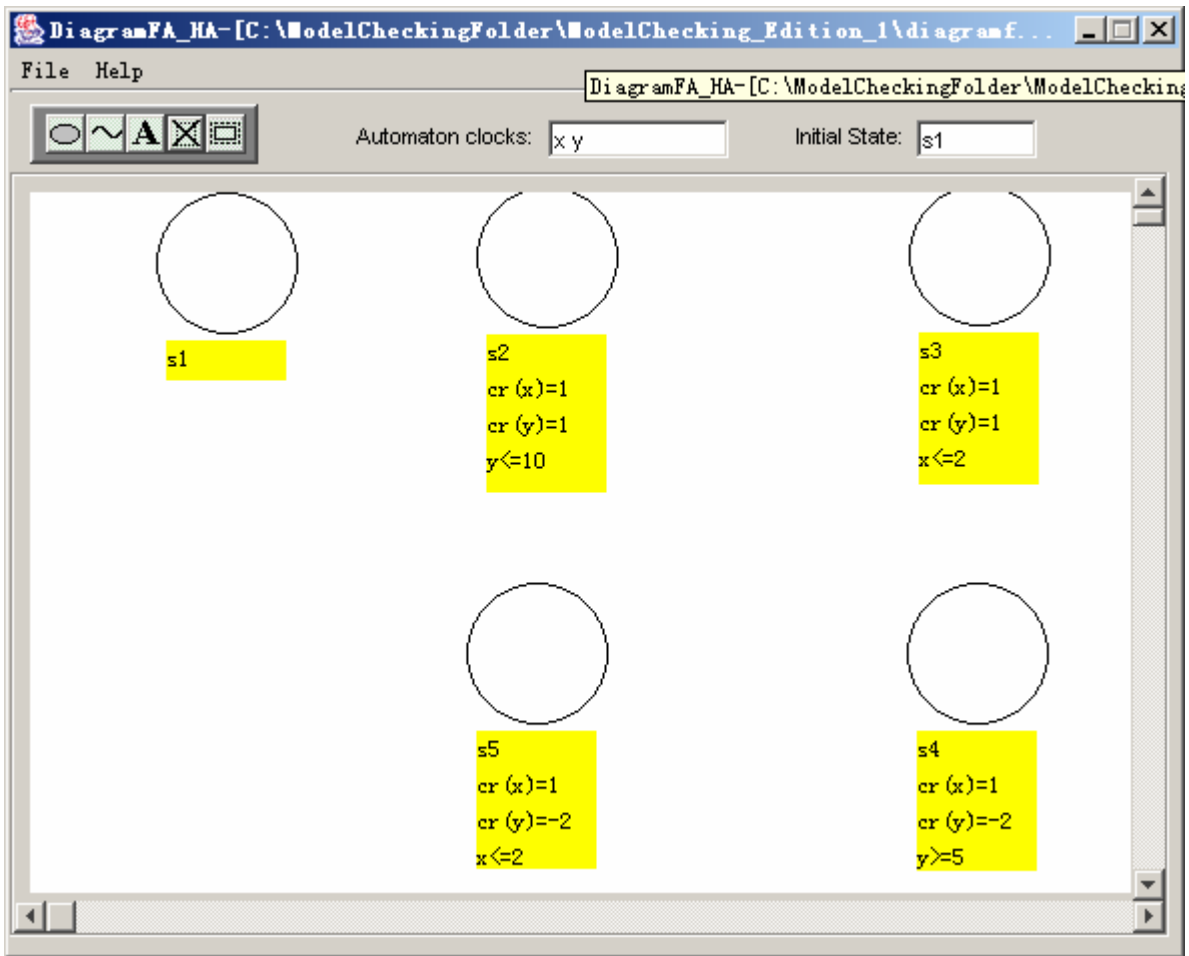
Click the button "Draw Circles"



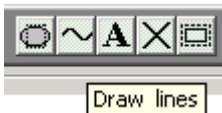
3.4.1 Add the initial State s1



3.4.2 Add all the states

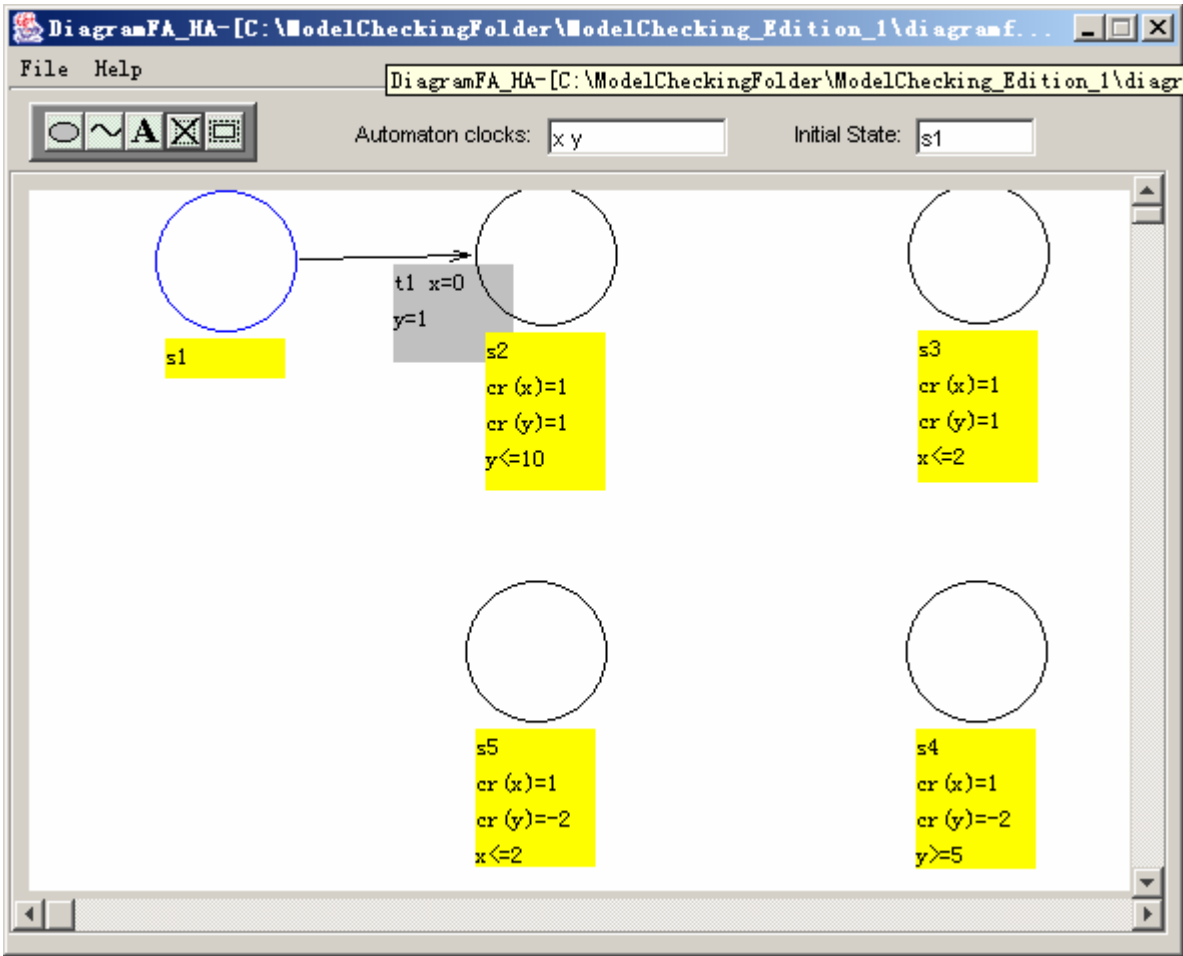


### 3.5 Draw the transitions

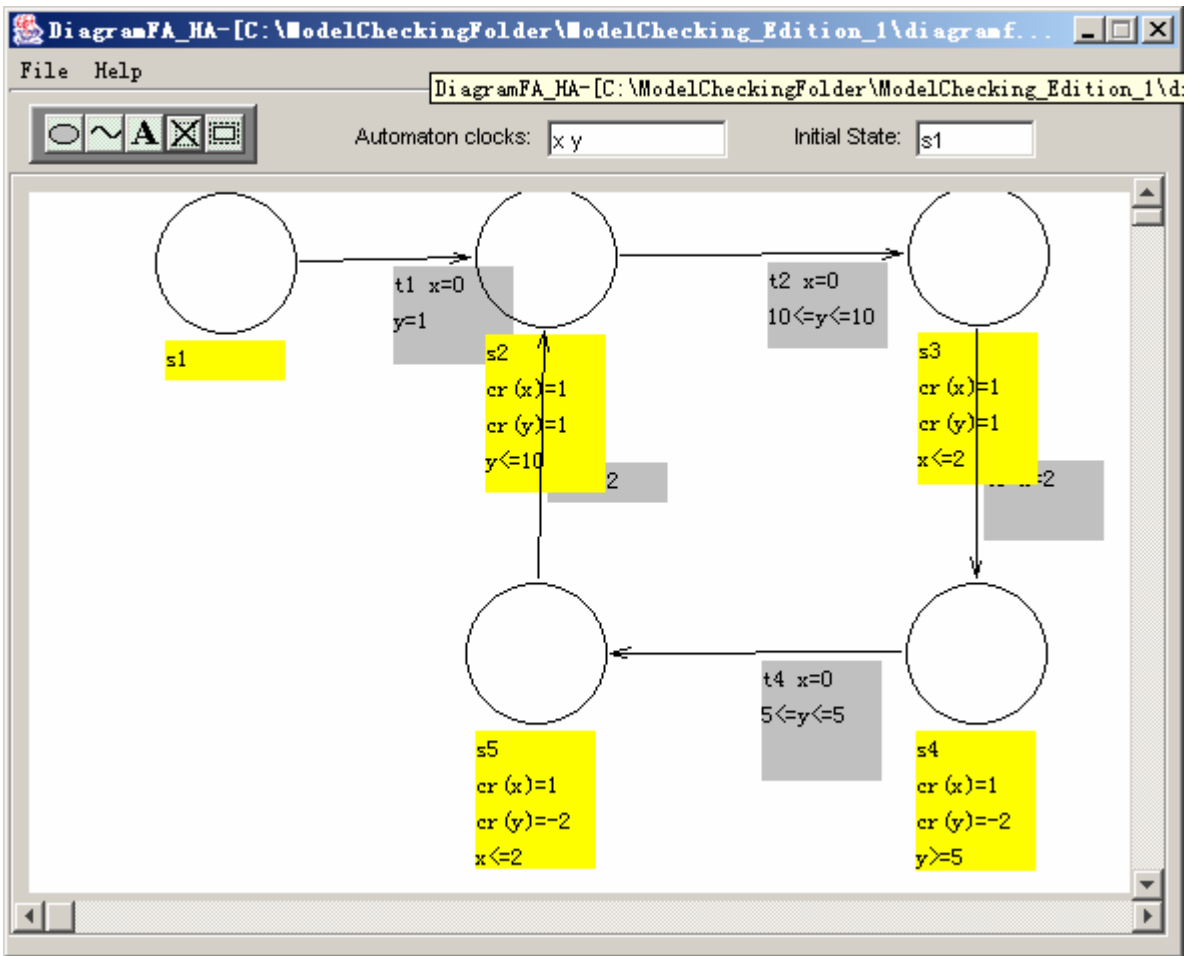


#### 3.5.1 Add the first transition t1

In this tool all the clocks should be initialized in the first transition!

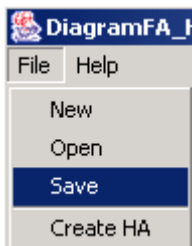


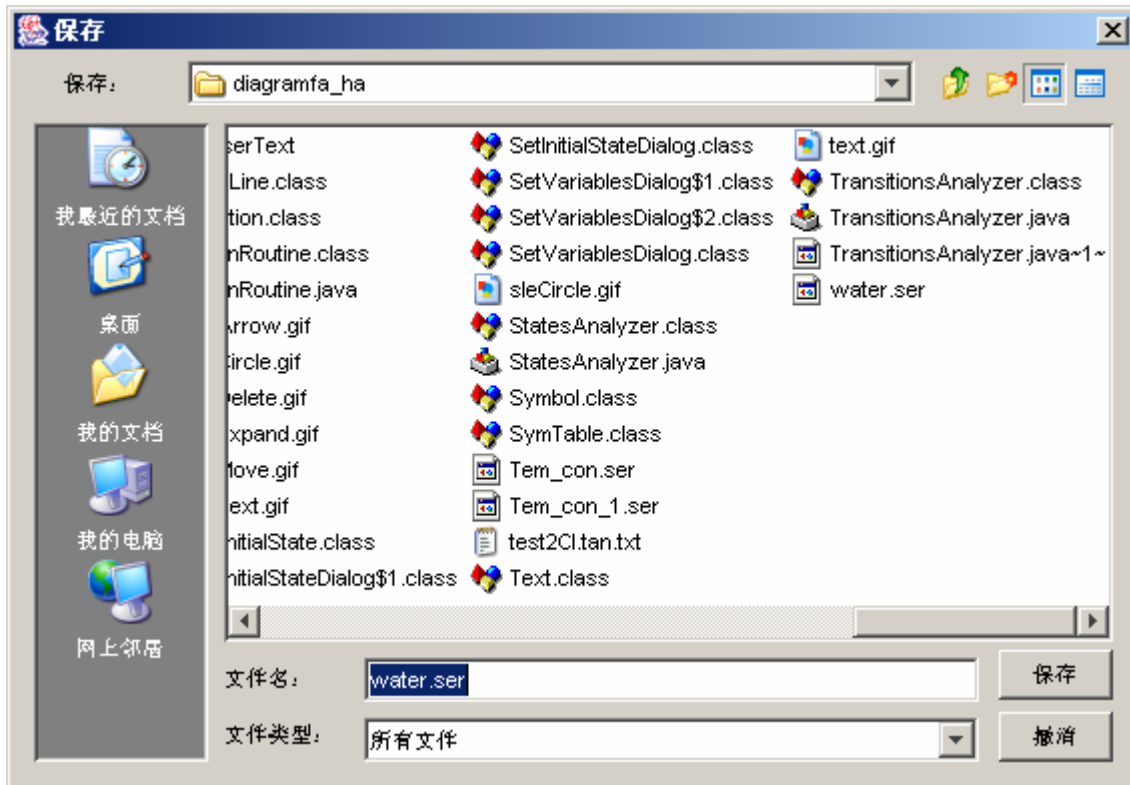
### 3.5.2 Add all the transition



### 3.6 Save the Graphical File

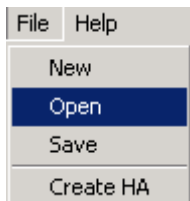
Choose the "Save" Button in the File Menu

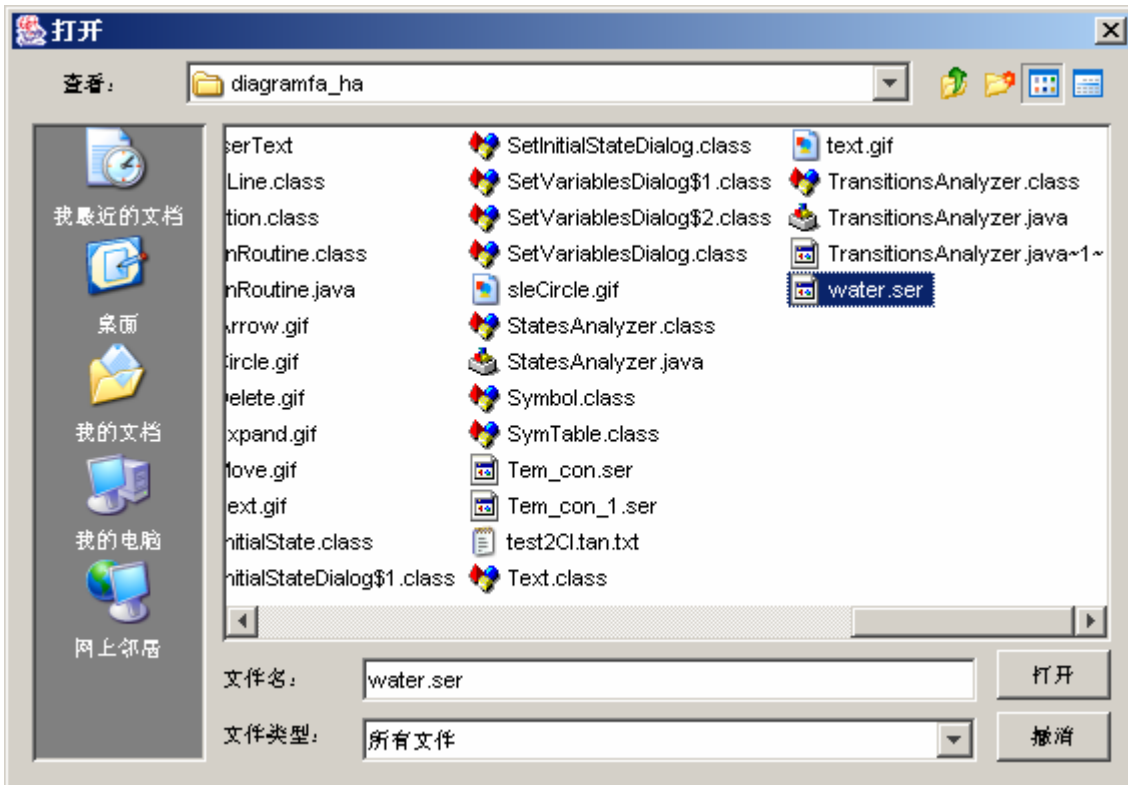




### 3.7 Open The Graphical File

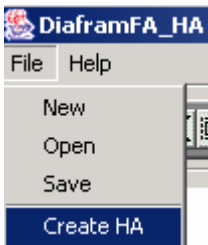
Choose the "Open" Button in the File Menu



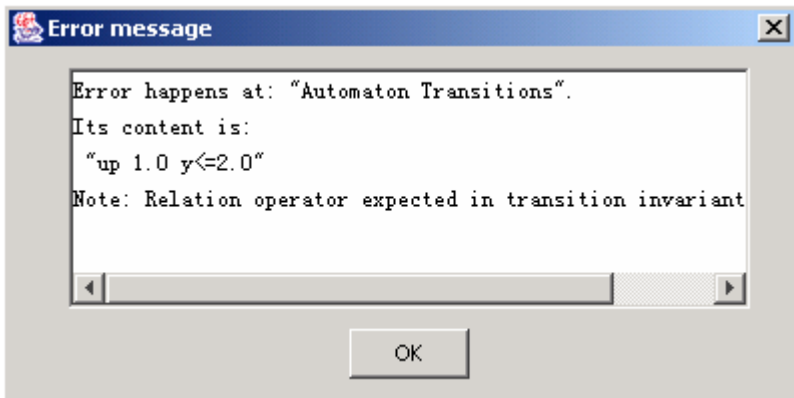


### 3.8 Make The HA File That Can Be Checked

Choose the "Create HA" Button in the File Menu

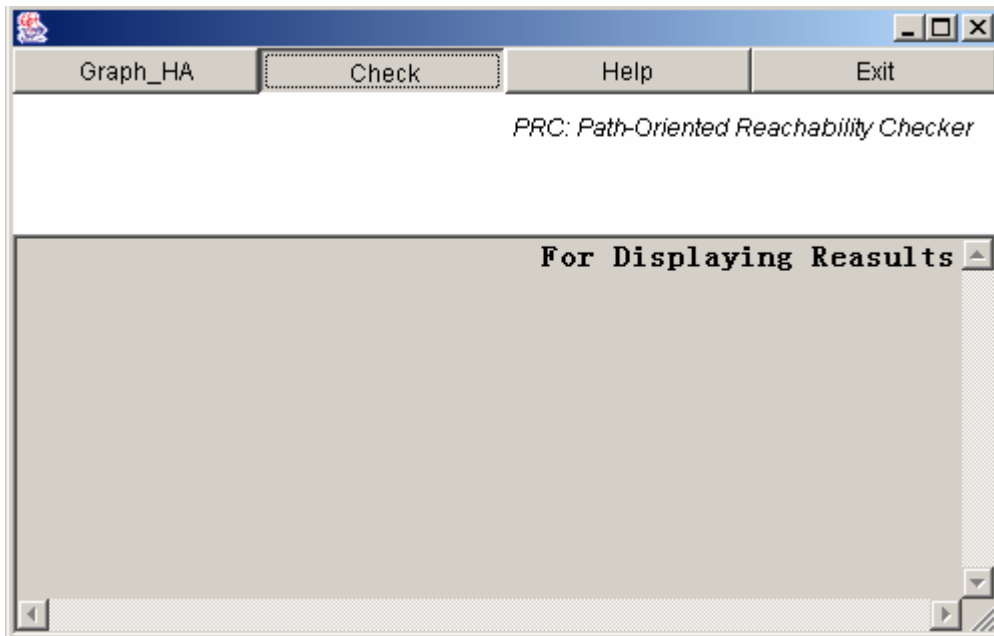


When you create ".ha" Files, we will do some simple syntax check, only when there is no errors we can create ".ha" Files.



#### 4 Checker Part

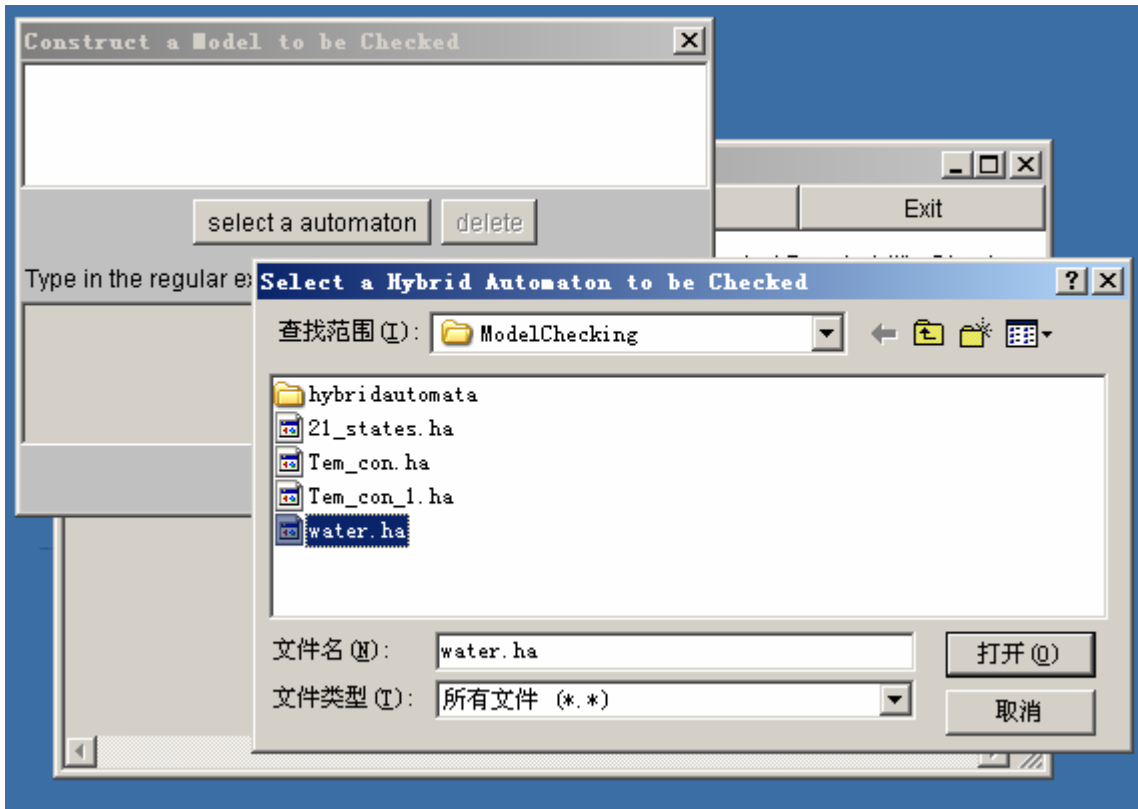
In the first UI, if you click the button of "check", you can enter the part of Check.



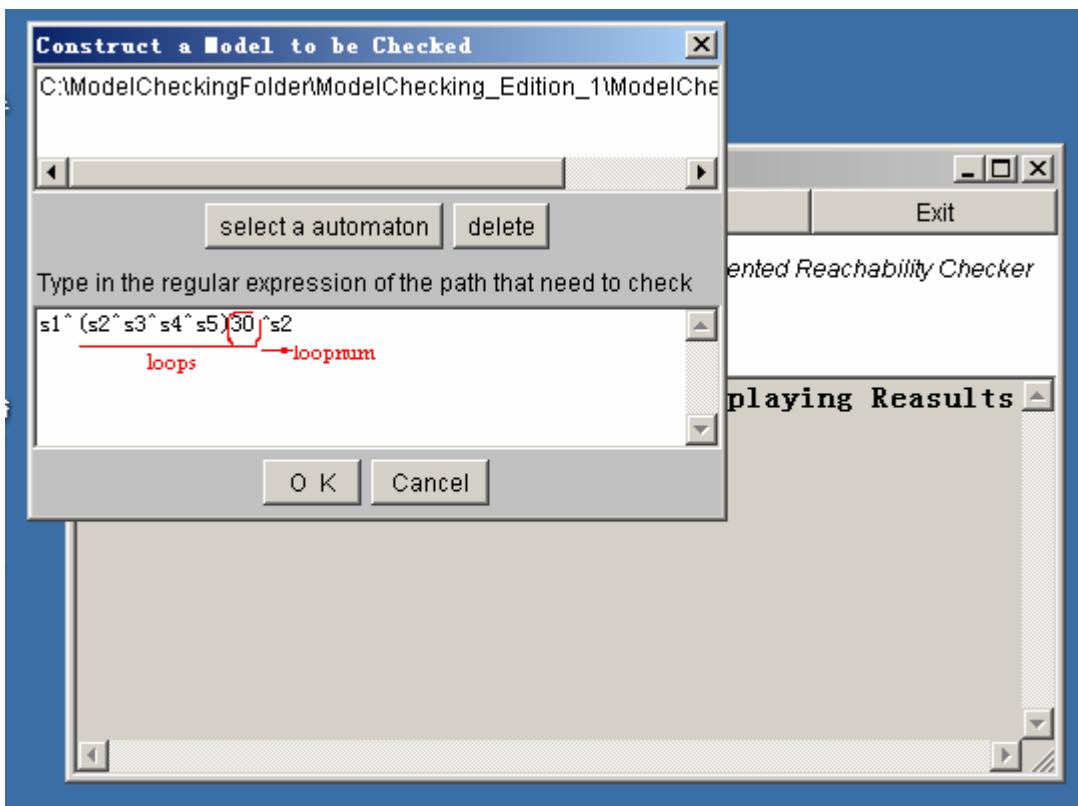
##### 4.1 Select HA File To Check

Choose a HA file that was created before to check.



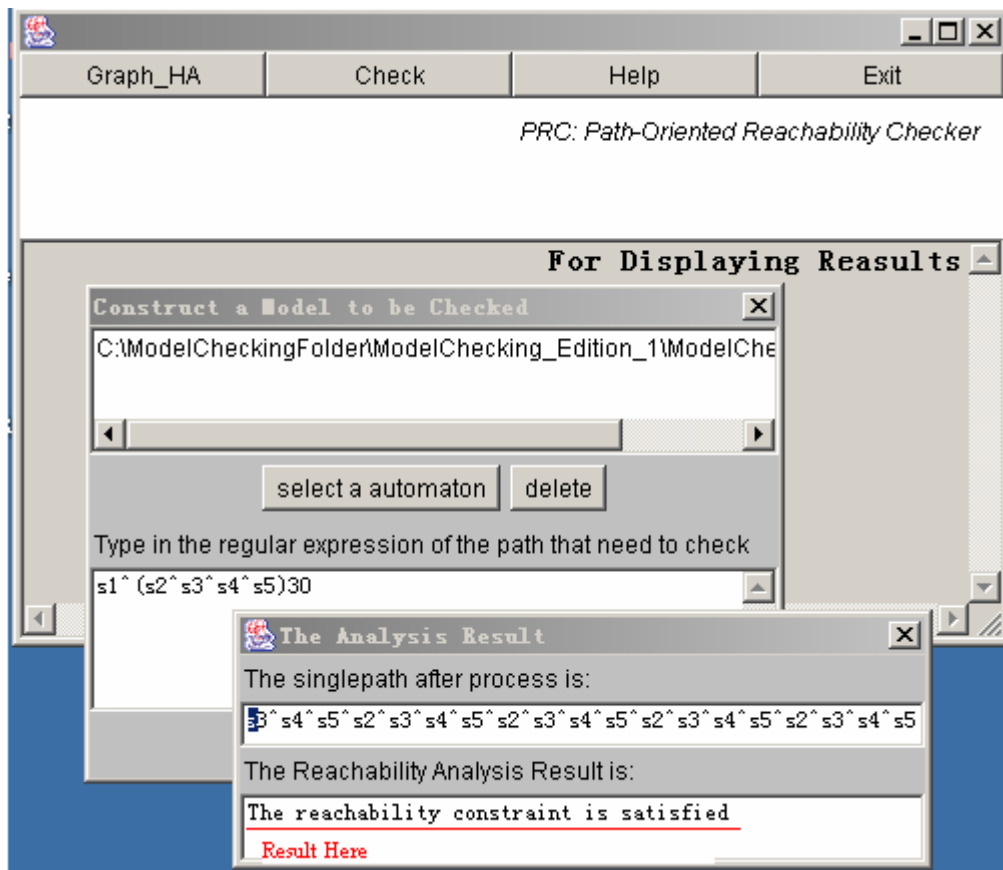


#### 4.2 Input the path you want to check



In this tool we can represent a path segment  $\rho$  in  $H$  by a simple form  $v_0v_1 \dots v_n$ , which is called *simple regular expression*, and if there is a loop we should write it in a pair of parentheses with the loop num after the right bracket.  $v_0$  stand for locations(State) here.

#### 4.3 Press OK To Start Check



The Check Result is show in the second text area

If you want to see the number of constraints and variables please check the console.

#### 5 Runtime environment

Your jdk edition should not earlier than jdk1.3.0