# THEORY OF COMPUTATION – CASE STUDY

**AIR CONDITIONER**

A picture containing text, logo, font, graphics

Description automatically generated

**Amrita School of Computing, Coimbatore Department of Computer Science and Engineering 2022 – 2023 Even Semester**

**BTech / II Year CSE / IV Semester**

**19CSE214 / Theory of Computation**

|  |  |
| --- | --- |
| ROLL NUMBER | NAME |
| CB.EN.U4CSE21010 | B.Thilaknath Reddy |
| CB.EN.U4CSE21011 | B.Sri Ganesh |
| CB.EN.U4CSE21051 | S.V.S Krishna Aditya |
| CB.EN.U4CSE21067 | Vedant DJ |
| CB.EN.U4CSE21068 | V.Vignesh |

# Table Of Contents

|  |  |  |
| --- | --- | --- |
| S. No | Contents | Page No |
| 1. | Problem Statement | 3 |
| 2. | Approach | 4 |
| 4. | Push Down Automata | 5 |
| 5. | Turing Machine | 14 |

**Problem Statement**

Design a push down automata and a Turing machine for Air Conditioner. To process it using multiple Functionalities and completing the needs of the user with exceptions kept in mind.

# Introduction

The Idea of our project, Air conditioner is that we provide the user utmost user-friendly environment where user can accordingly to his needs can set all the functionality of the AC and in case of faulty case the machine take cares of itself where all the functionality such as temperature, modes, timer and heating can be processed.

# Approach

On switching on, the AC does a systems check to make sure all the vital components are functioning and the power supply is under a certain threshold. In case of an anomaly in the systems check, an alarm is triggered and the AC automatically turns itself off.

There are two important modes pertaining to the machine

* Default mode
* Manual mode

The Default mode is included in case of the the user wants to access two modes Eco mode or efficiency mode where there is both timer and temperature can be set automatically by the AC itself.

The manual mode is included for the general case where the user might want to AC to perform either the heating or the cooling mode by us setting the temperature and timer manually.

The above AC is more focused on reducing human effort and optimize the process of sheer comfortability . The theory behind the design will be represented in our project using a PDA and a Turing machine.

# Pushdown Automata

## Input Symbols

**q -** Efficiency o – mechanical start

**x -** Economy

**s -** manual

**r -** Turbo

**d -** Dry\_mode

**o -** Mechanical\_start

**g -** Capacitor\_failure

**p -** Fans’off

**j -** Time-reached

**k -** Heater

**1 – ac on**

**2 – temperature**

**3 – time**

**I – alert state**

**8 – heater time**

**4 – residual fan**

**5 – cooling**

**6 – auto\_off**

**7 – manual\_off**

**i – default**

**RT – required temperature**

**Rt – required time**

# Stack Symbols

Z – initial stack symbol

A – checking air conditioner status

F - failure

H - heater

O - manual

G - turbo

V – dry mode

P - default

E - efficiency

t – time set

e - evaporation

f – residual fan

B - cooling

c – mechanical start

D – required temperature

h – 42\*C

U – hazard detected

n – required time

m – required time

|  |  |  |
| --- | --- | --- |
| **Set of states** |  | |
| AC on | ON | -- initial state |
| AC check | CH |  |
| Alert | RT |  |
| Default  16\*C  21\*C  3 hrs | Df T1 T2 E1 |  |
| 1hr | E2 |  |
| Evap\_coil | EC |  |
| Cooling | CO |  |
| Hazard | HZ |  |
| Mech\_start | MS |  |
| Req\_time | RE |  |
| Req\_temp | RT |  |
| Turbo | TR |  |
| Dry mode | DM |  |
| Manual | MN |  |
| Heater | HT |  |
| 42\*C | CE |  |
| Alarm | AL |  |
| Cleaning | CL |  |
| Manual\_off | MO | -> Final state |

AC\_off OF1 -> Final state

Off OF2 -> Final state

# Working

## Preparation

* Turn on, Systems check – On switch the machine on it reaches the turn on state (idle) .

Top of stack (before)– Z Top of stack (after)-- AZ

* A routine check up will be done automatically by the machine. Top of stack (before) – AZ

Top of stack (after) -- PA (default) , OA (manual),HA(heater)

## Mode selection

* Default – Upon selecting the present mode, there are two main presets for which the AC can work through certain process.
* 16\*C for Efficiency(input-q) and 21\*C for

Econoy(input-x)

Top of stack (before) – P

Top of stack (after) – EP

Top of stack (before) – P

Top of stack (after) – EP

* Efficiency takes 3 hours to cool and Economy takes 1 hour to cool

For this time duration:

Top of stack (before) – E

Top of stack (after) – tE

Top of stack (before) -- OCZ

In the default after the allocation of time and temperature the air conditioner turn on by starting the evaporation coil.

Top of stack (before) – t

Top of stack (after) – et

After starting the evaporation coil we turn on the residual fan which is set

Outside the house to take out all the waste.

Top of stack (before) – e

Top of stack (after) – fe

After turning on all the basic elements to run the AC we now start the cooling process which is the main goal

` Top of stack (before) – f

Top of stack (after) – Bf

After all these we now have the cleaning process that starts which is commenced after the use of the AC.

Top of stack (before) – B

Top of stack (after) – lambda

After the Allotted time is elapsed the which is given by the user the AC turns off itself

Top of stack (before) – Z

Top of stack (after) – Z

This above process was one of the case that represents the default case now let us discuss the manual state.

Now there are two transition from the manual state where the first one is turbo which provides the maximum result.

Top of stack (before) –o

Top of stack (after) – Go

The other possible transition is dry mode which does not use water in the AC and provides dry air with no moisture in it.

Top of stack (before) – o

Top of stack (after) – Vo

After the above two states we now set the temperature at which the AC is supposed to work.

Top of stack (before) (turbo)–G

Top of stack (after) (turbo)– DG

Top of stack (before) (dry)–V

Top of stack (after) (dry)– DV

Now after the temperature is set we now set the time for which the AC should work before it turns Off itself

Top of the stack(before) – D

Top of the stack (after)- mD

After all the functionality start we now start the machine by mechanically starting the machine .

Top of the stack(before) – lambda

Top of the stack (after)- c

Now after the user is notified about the hazard the machine reverts everything and turns off

Top of the stack(before) – A

Top of the stack (after)- Lambda

Now in this case if due to any issue the AC does not function properly the AC goes into the hazard state.

Top of the stack(before) – c

Top of the stack (after)- Uc

This hazard leads to the alarm ringing which notifies everyone about the AC not functioning properly.

Top of the stack(before) – U

Top of the stack (after)- Lambda

## Default

* Since it is the Default Mode, There are two mode which is Efficiency Mode and Eco Mode and the temperature and time is set to 16\*C and 3Hr for Efficiency and 21\*C and 1hrs for Eco mode
* Efficiency Mode

Top of stack (before) – EPAZ Top of stack (after) -- etEPAZ

* Eco Mode

Top of stack (before) -- EPAZ Top of stack (after) -- etEPAZ

* Next we have evapourating coil which allows to take the heat out from our home

Top of stack (before) – etEPAZ

Top of stack (after) – fetEPAZ

* Here the Residual fans helps to get the air faster

Top of stack (before) – fetEPAZ

Top of stack (after) – BfetEPAZ

* This is the state where the cooling of the room starts

Top of stack (before) -- BfetEPAZ

Top of stack (after) –UBfetEPAZ(Hazard), BfetEPAZ(manual\_off), BfetEPAZ(Auto\_off)

## Manual

* Here there are two modes for manual which are Turbo and Dry mode where turbo mode is for highest fan speed and dry mode is to remove the moist from the room

Top of stack (before) -- OAZ

Top of stack (after) – GOAZ (Turbo), VOAZ (Dry)

* This is to set the temperature of our desired wish – req\_temp

Top of stack (before) – GOAZ, VOAZ

Top of stack (after) – DGOAZ (Turbo), DVOAZ (dry mode)

* Here it is Req\_time i.e., the amount of time which is required to run the AC

Top of stack (before) – DGOAZ, DVOAZ

Top of stack (after) – mDGOAZ (Turbo), mDVOAZ (Dry mode)

* Here it is the mechanical start of the fan, Mech\_fan

op of stack (before) – mDGOAZ, mDVOAZ

Top of stack (after) – cDGOAZ (Turbo), cDVOAZ (Dry),

## Hazard Management

* After the cooling or the mechanical fans get started if there is any issue then it will go to the hazard state

Top of stack (before) –cDGOAZ

Top of stack (after) – UcDGOAZ(manual), UBfetEPAZ(default)

Now it will ring an alarm and the AC will OFF automatically

Top of stack (before) – UcDGOAZ

Top of stack (after) – cDGOAZ(manual), BfetEPAZ(default)

* Now from alarm to the final state

Top of stack (before) – cDGOAZ(manual), BfetEPAZ(default)

Top of stack (after) – Z

## Final

* There are two ways to reach the final state which is manual off and auto off
* Manual\_off

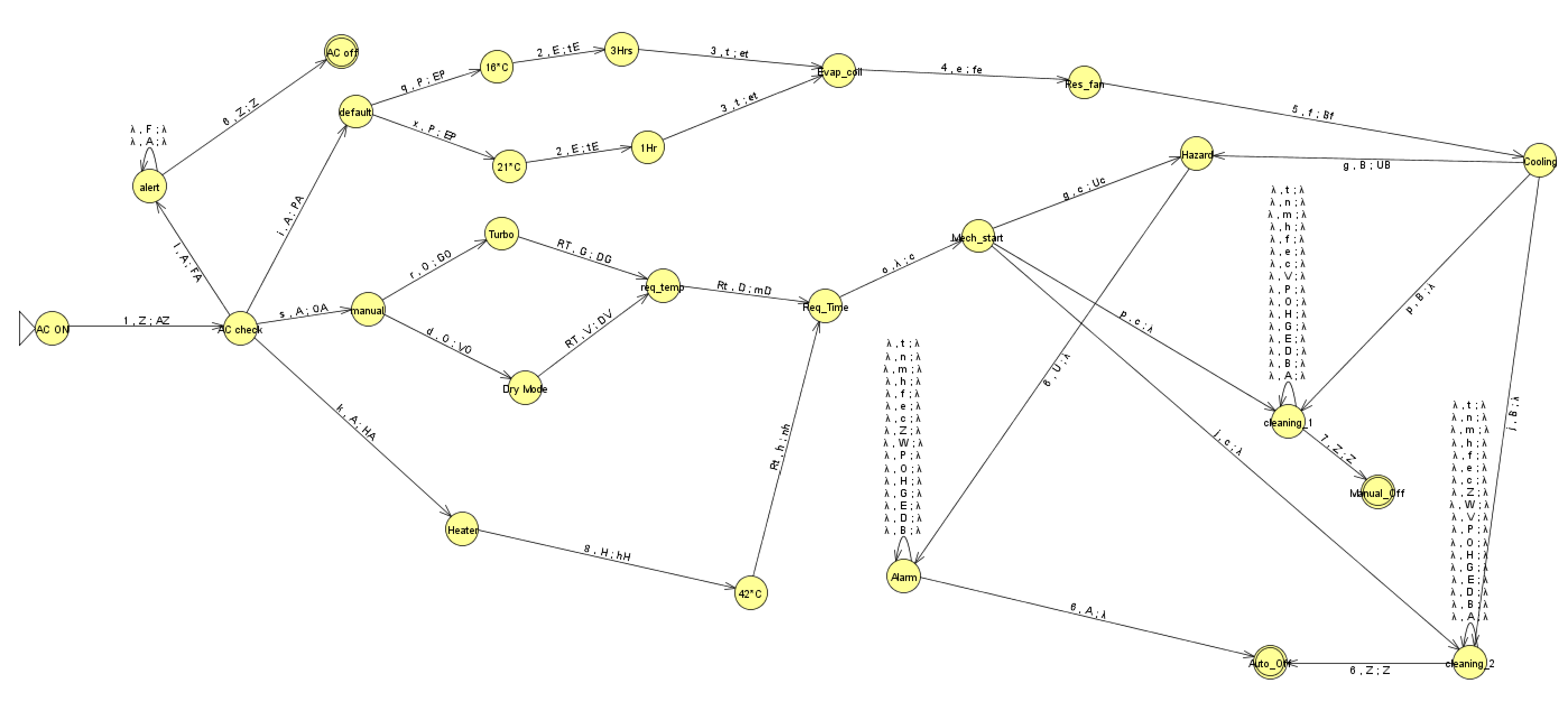
Top of stack (before) – cDGOAZ(manual-cleaning\_1), BfetEPAZ(default-cleaning\_2)

Top of stack (after) –Z(manual-manual\_off), Z(default-manual\_off)

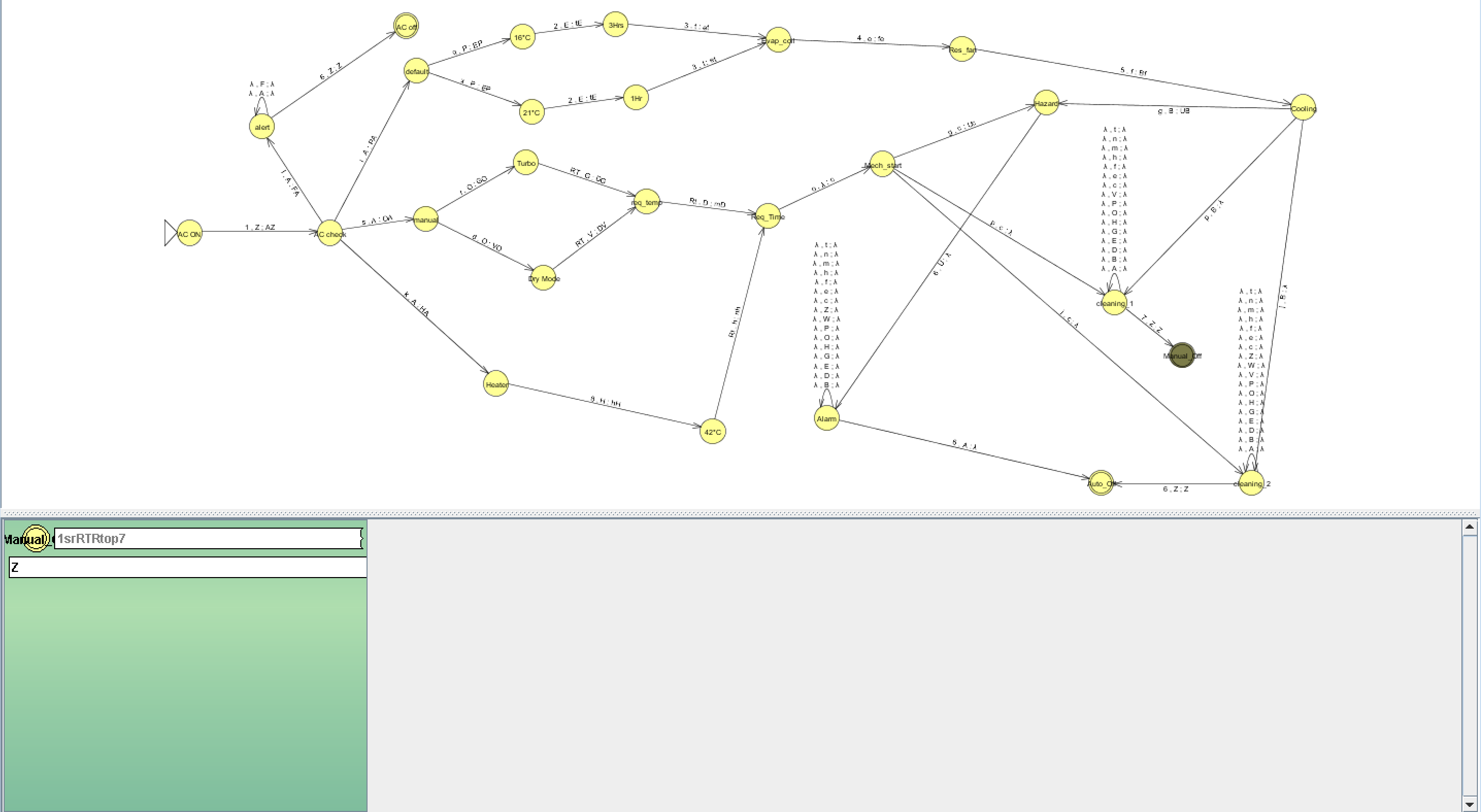
* Auto off

Top of stack (before) – cDGOAZ(manual-cleaning\_1), BfetEPAZ(default-cleaning\_2)

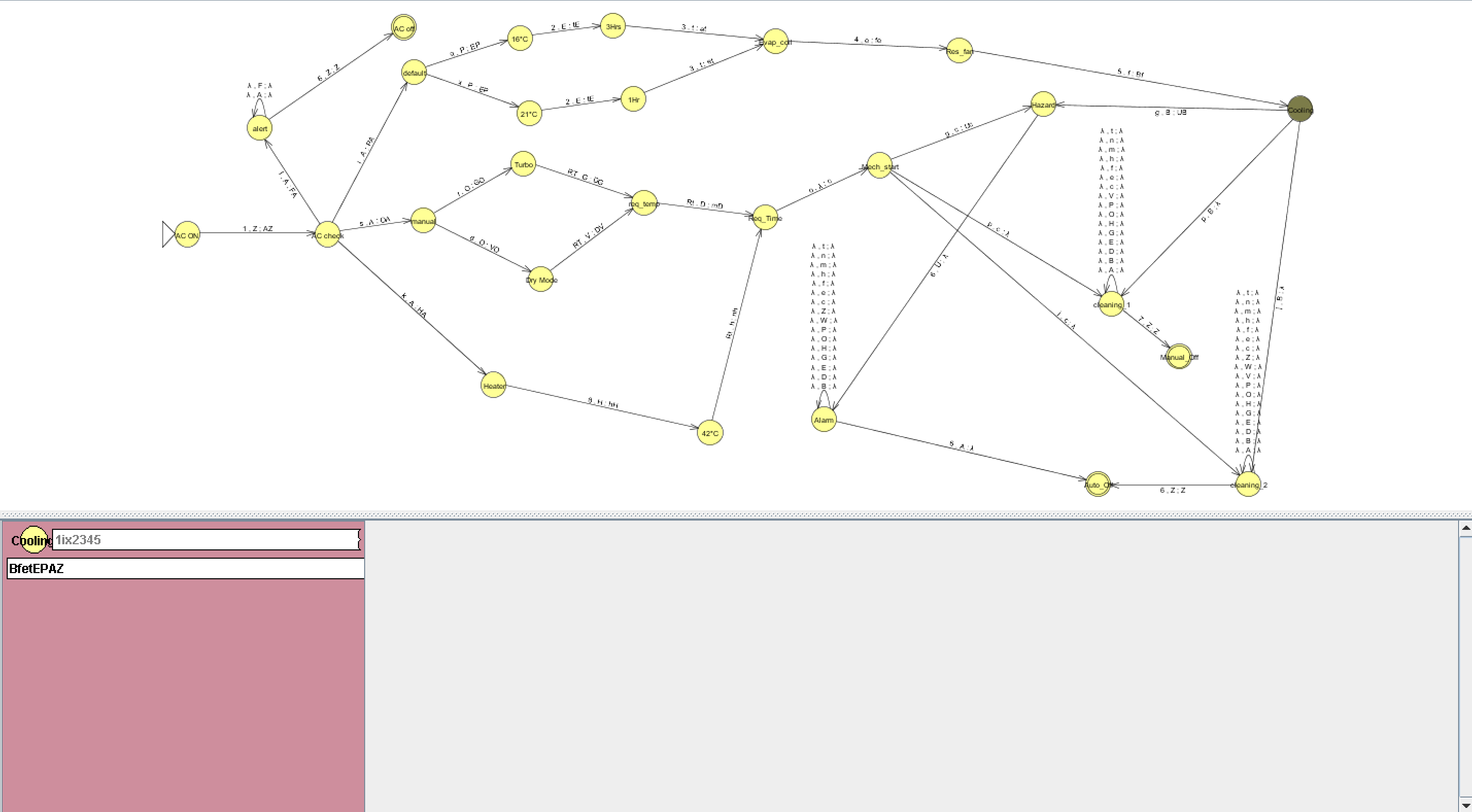
Top of stack (after) – Z(manual-manual\_off), Z(default-manual\_off)



**FOR VALID INPUT**

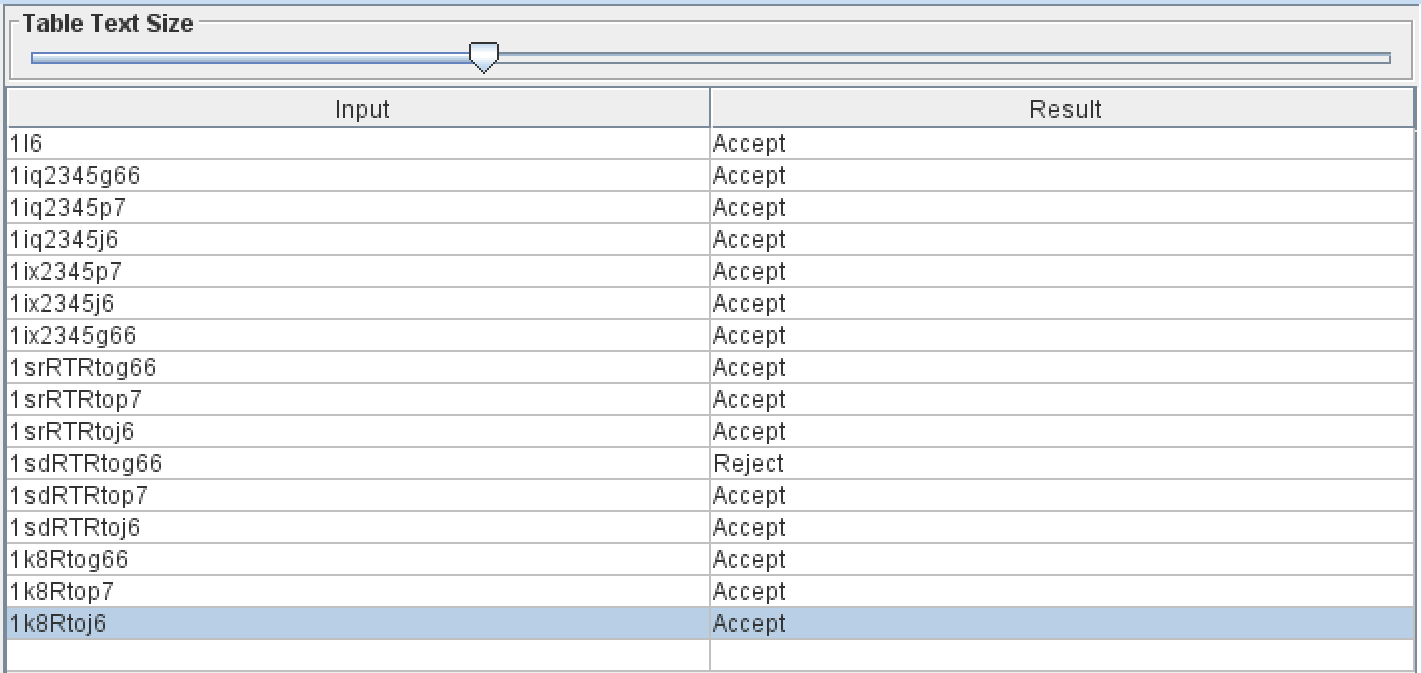


**FOR INVALID INPUT**



# LANGUAGE:

1I6 U 1iq2345g66 U 1iq2345p7 U 1iq2345j6 U 1ix2345g66 U 1ix2345j6 U 1iq2345p7 U 1sdRTRtog66 U 1sdRTRtop7 U 1sdRTRtoj6 U 1srRTRtog66 U 1srRTRtop7 U 1srRTRtoj6 U 1k8Rt0g66 U 1k8Rt0j6 U 1k8Rt0p7



GRAMMER:

Grammar rules:

S -> 1I6 | 1qXg66 | 1qXp7 | 1qXj6

X -> 2345X | ε

Explanation:

Non-terminal symbol S is the start symbol.

S generates strings that start with '1', followed by either 'I6' or 'qXg66', 'qXp7', or 'qXj6'.

Non-terminal symbol X generates the middle part of the string, which can be zero or more repetitions of '2345' followed by X. The use of X -> ε allows for an empty derivation, which means X can be absent from the string.

This CFG generates all the strings in the given language

L: {1I6, 1iq2345g66, 1iq2345p7, 1iq2345j6, 1ix2345g66, 1ix2345j6, 1iq2345p7, 1sdRTRtog66, 1sdRTRtop7, 1sdRTRtoj6, 1srRTRtog66, 1srRTRtop7, 1srRTRtoj6, 1k8Rt0g66, 1k8Rt0j6, 1k8Rt0p7}.

# Turing Machine

## Introduction:

The concept of a Turing machine can be applied to understand the control unit operating. The control unit in air conditioner acts a computational system receiving input from user executing predefined logic and making decision to regulate the cooling or heating. In this way, the control unit operates like a simplified version of a Turing machine, facilitating efficient and automated temperature control.

## State Definitions:

The Turing machine consists of the following states:

1. AC Off 17.mode

2.AC On 18.Req\_temp

3.Ac Check 19.Req\_time

4. Alert 20.mech\_start

5. Off 21.Heater

6. default 22.42\*C

7.Temp

8.Hours

9.Eval\_re

10.Cooling

11.Hazard

12.Alarm

13.Shutdowning

14.Cleaning

15.Off

16.Manual

## Tape Alphabet (T):

The tape alphabet of the Turing machine includes the following symbols:

* + A - ON
  + F - FALIURE
  + D - DEFAULT
  + T - TEMP
  + H - HOURS
  + C - CLEANING
  + P - HAZARD
  + M - MANUAL
  + m - MODE
  + r – required temperature
  + t – REQUIRED TIME
  + S – MECHANICAL START
  + G - Heater
  + B – Cleaning from mechanical start

## Input Alphabet (∑):

The input alphabet for the Turing machine is defined as:

A – ac check

D - default

T - temperature

H - hours

e – evaporating coil

P - hazard

S – mechanical start

B - cleaning

x - off

4 – 42\*c

F – faliure

M - manual

G - heater

r – required temperature

t – required time

c - cleaning

z, w, y, g, f, t, b ,r – shut downing process

## Transition Functions:

The Turing machine's transition functions define the state transitions based on the current state and the symbol read from the tape. Here are the transition functions:

(AC ON, A) -> (ac check, A, R): when the input is A it changes from ac on state to ac check state by moving pointer to right.

(ac check, F) -> (ALERT, F, R): if before starting the process if there is an issue it goes to alert state by the moving pointer to right.

(alert, e) -> (ac off, e, L): when it went to ac off state it takes no more input and goes to left until it reaches A and changes it to x

(ac off, ) -> (off, , R): from ac off it turns off the machine completely and reaches the final state

(ac check, D) -> (default, D, R): from ac check if the user wants default mode it goes to default state by moving the pointer to right

(default, T) -> (temp, T, R): by the mode the user has selected from default it goes to the temp which is already set temperature of the air

(temp, H) -> (hours, H, R): from the mode, now it goes to the time for how much the air will be provided by the ac

(hours, e) -> (eval\_res, e, R): here the process undergoes it self no need of any input and it reaches the cooling state

(cooling, P) -> (hazard, o, L): so if there is an problem or issue in the process of cooling the system will result an hazard and then it will change the alphabet P to o by moving the pointer to left

(Hazard, A) -> (alarm, x, R): after reaching the state hazard it will not change all the alphabets since there is an issue. After reaching the end A that will be passed as an input to state alarm after reaching alarm it will change A to x

(alarm, x) -> (shutdowning, x, S): after reaching the state alarm if will change every alphabet until it reaches x after reaching x it will pass as an input to shutdown

(shutdown, x) -> (off, x, S): here nothing will be changed it will reach the final state of by pointing the pointer to start of the tape

(cooling, c) -> (cleaning, c, L): from cooling if there is no issue it will reach the cleaning state until the time is reached or user opts to off the ac

(cleaning, A) -> (shutdowning, x, R):in cleaning it changes every alphabet and keeps on moving the pointer to left until it reaches A

(shutdowning, x) -> (off, x, S): as mentioned above when shutdowning has an input x it reaches the final state off by keeping the pointer to the start of the tape

(ac check, M) -> (manual, M, R): from ac check if the user chooses manual mode it will go to the manual state by moving the pointer to right

(manual, m) -> (mode, m, R): in the manual state user has to select his mode (turbo, dry mode) after choosing the mode it will reach the mode state by moving the pointer to right

(mode, r) -> (req\_temp, r, R): he has to set his required temperature by passing an input r

(req\_temp, B) -> (cleaning, b, L): after reaching the cleaning state its starts changing the alphabets by first reaching A

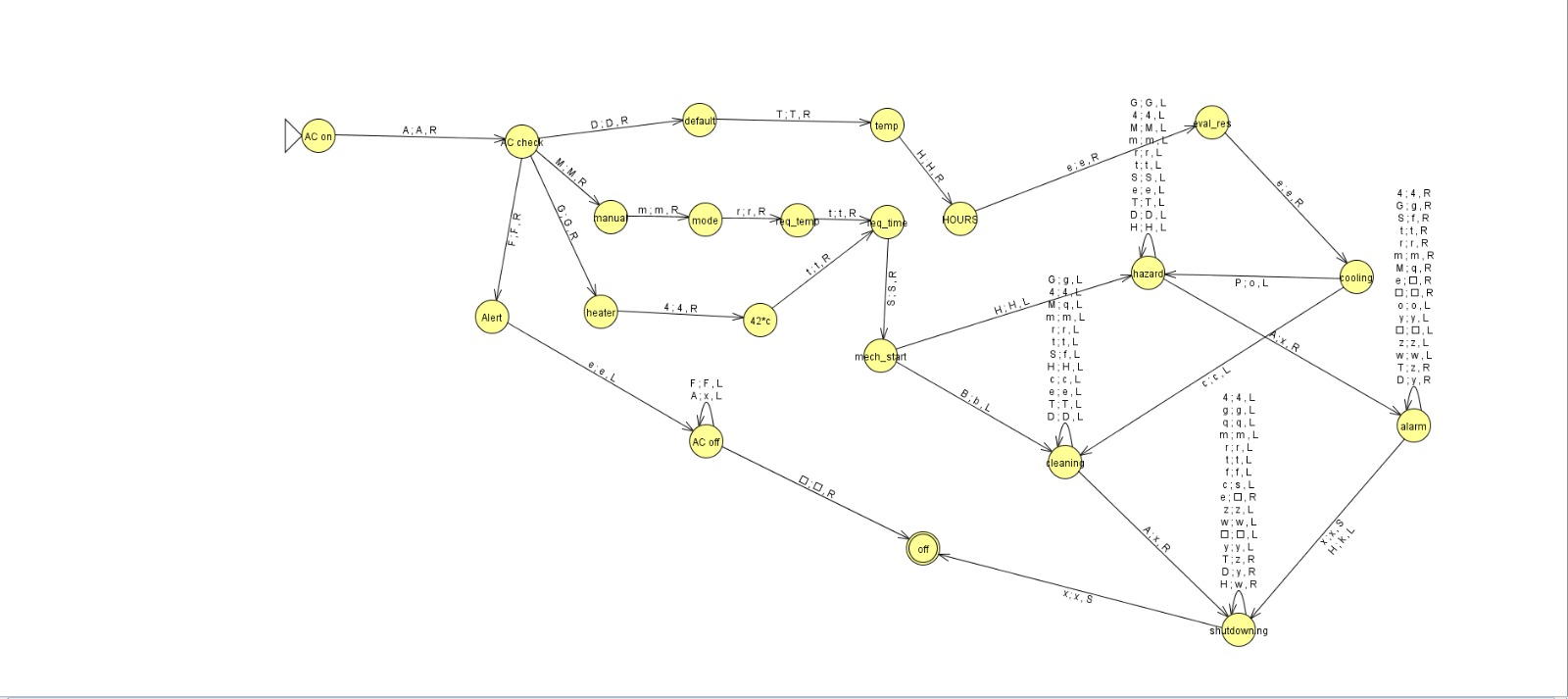
(cleaning, A) -> (Shutdowning, x, R): finally after changing all alphabets it will change A to x and reaches the final state as mentioned above

(mech\_start, H) -> (hazard, H, L):after starting the process in manual mdoe if there is any issue it started moving to left and reaches the hazard state

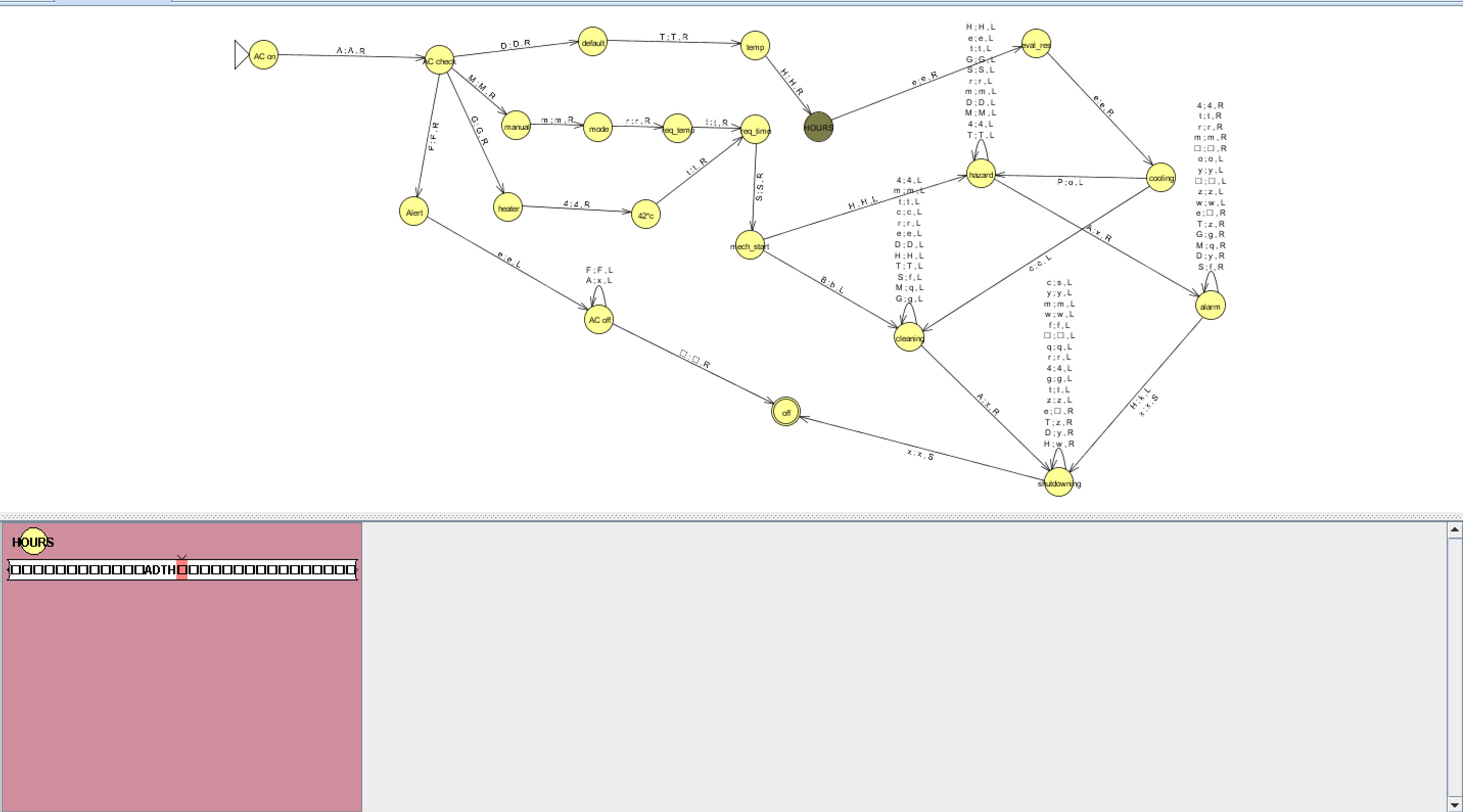
(ac check, G) -> (heater, G, R):when user wants heater he presses the heater mode where it reaches heater state then the default temperature would be 42\*c which will be started by the mechanical start process

## Functionality:

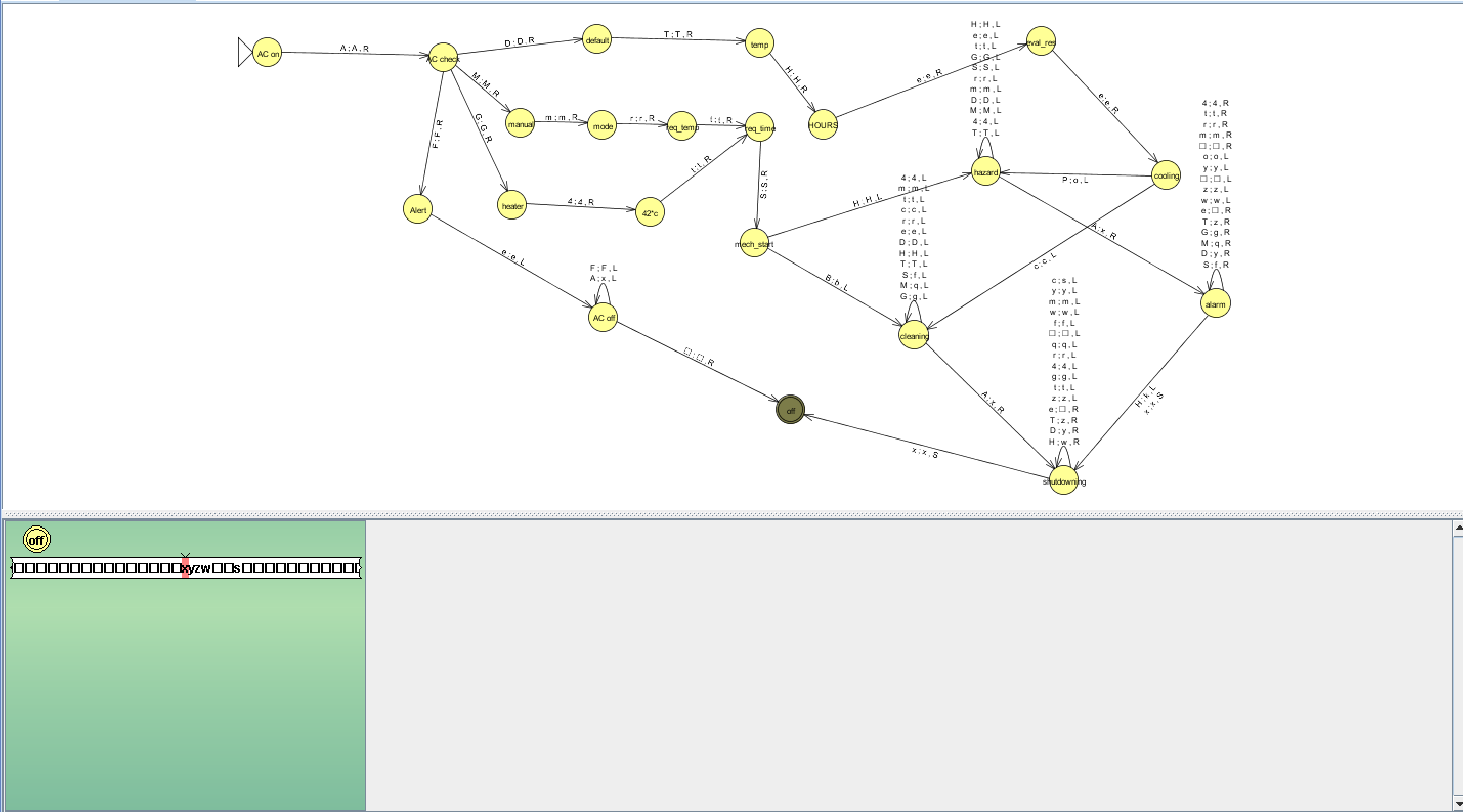
The control unit in an air conditioning system plays a crucial role in maintaining indoor comfort. It receives input from temperature sensors and compares the actual temperature to the desired set temperature, using programmed algorithms, it determines whether cooling or heating is required. The control unit then activates the appropriate components such as the compressor and fans to regulate temperature and airflow. It continuously monitors the temperature and adjusts system operation to maintain the desired comfort level.



**FOR INVALID INPUT**



**FOR VALID INPUT:**

****