The Implementation of Specification-based Testing System for Real-time Reactive System in TROMLAB Framework

Minghua Chen

A MAJOR REPORT

IN

THE DEPARTMENT

Of

COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

CONCORDIA UNIVERSITY

MONTREAL, QUEBEC, CANADA

DECEMBER 2002

Abstract

The Implementation of Specification-based Testing System for Real-time Reactive System in TROMLIB Framework

Minghua Chen

This major report describes the analysis, design, and implementation of the "TROM-SBTS: specification-based testing system for real-time reactive system in TROMLIB framework" in Java. Specification-based testing is a new approach in black box testing for real-time reactive system developed by Mao Zheng [Mao02]. The TROM-SBTS has specification-based unit testing, pair testing, and system testing functionalities. After studying the specification-based testing algorithms reported in Mao's thesis [Mao02], the updating of the algorithms is reported. Object oriented design technology is used in the system architecture and detailed design of the TROM-SBTS. Some existing softwares are integrated into the system. Finally, the train-gate-controller problem is taken as the case study for unit testing, pair testing, and system testing. The empirical result proves the correctness of algorithms and their implementation.

Key words: specification-based testing, real-time, reactive system

Acknowledgements

I would like to express my special gratitude to all the people who gave me the great help during this major report.

I am greatly indebted to my supervisor, Professor Olga Ormandjieva. She gave me a lot of helpful suggestions and encouraged my interest in the specification based testing of real-time reactive application field. With her patient advice and constant help, I learned much useful knowledge and completed my major report successfully.

I am pleased to thank Dr. Alagar as the examiner for the report. He gaves me many useful comments during the work. My sincere thanks are extended to Halina Monkiewicz, the Graduate Program Secretary, for her collaboration and support.

Finally, my special thanks are also due to the faculties and staffs in the Computer Science Department at Concordia University, who provided the large support during my master program's study.

Contents

Figures

Chapter 1 Introduction

1.1 Real Time Reactive System

Reactive systems [HP85] interact continuously with their environment through stimulus and responses. The behavior of a reactive system is infinitively ongoing: the process in a reactive system is usually continuously responding to the stimulus from its environment. Real-time reactive system has two important properties:

- Stimulus synchronization: the process always reacts to a stimulus from its environment;
- Response synchronization: the acceptable time elapsed between a stimulus and its response is relative to the dynamics of the environment, so that the environment is still receptive to the response.

The correct behavior of non-real time systems is determined by the functional correctness of the result. By contrast, its stimulus-response behavior of real time reactive systems is regulated by strict time constraints for the real-time aspects; the real-time systems require both the functional correctness and timing correctness.

1.2 TROM Formalism & TROMLAB

Timed Reactive Object Model (TROM) formalism, which is founded on merging object-oriented and real time technologies, is introduced in [Ach95] for describing functional and timing properties of real time reactive system with

formal notations. In TROM formalism reactive objects are described as labelled transition systems augmented with ports, attributes, logical assertions on the attributes, and time constraints. TROMLAB [AAM98] is a framework for rigorous development of real-time reactive system. The framework includes a number of tools to support a rigorous development. The prototype TROMLAB environment, shown in Figure 1.01, has been constructed over the last six years and used as a test bed for real-time reactive systems development. During this time a number of tools in the framework have been developed [Tao99] [Mut96] [Nag99] [Pop99] [Pom99] [Sri99] [Hai99] [Zha00]. With these tools' support, engineers can develop real-time reactive systems with a high-level of reliability.

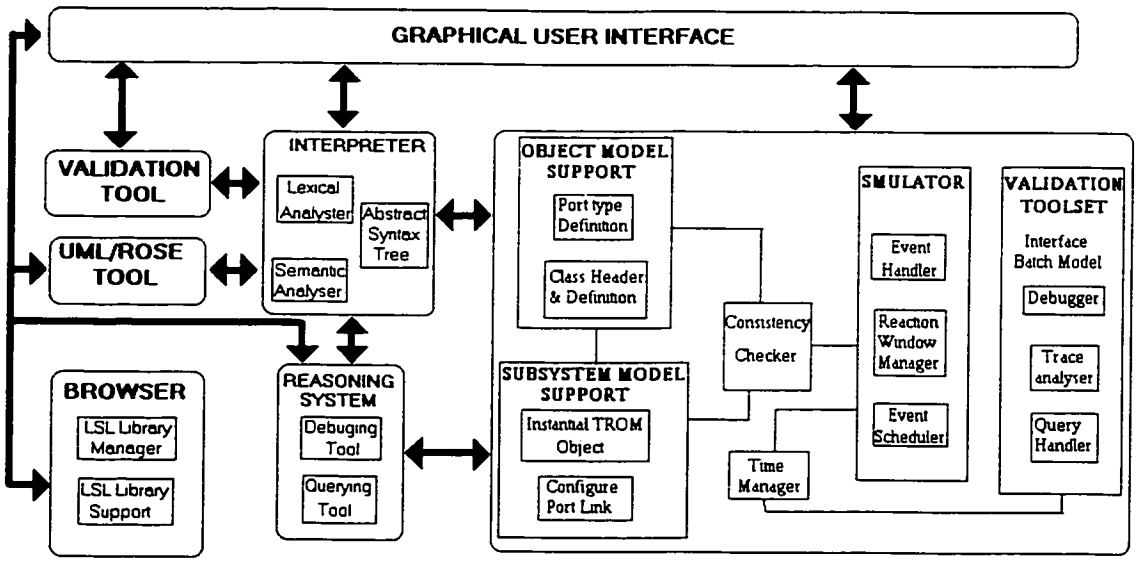

Figure 1.01 The Framework of Diagram

Software development activity in TROMLAB framework is carried out according to TROM formalism integrated in the process model. The TROMLAB framework is a three-tire structure. The three tires, from low to high, are abstract data types, timed reactive object model, and system configuration specification.

In the architecture, high tire can include lower tires. The formalism is sufficiently expressive for modeling large real-time reactive systems.

1.3 Specification-Based Testing

Software that operate and control safety-critical applications, which is reactive real-time system, must be thoroughly tested before the software is installed in the operation environment. Because reactive systems are mostly safety-critical, verification and testing must be conducted at the early stages of specification and design in order to ensure as much design problems as possible could be found before implementation and correct implementation of safety and time- dependent behaviours. The requirements lead to the specification-based testing technologies to be developed.

Specification-based testing [AOZ00a] [RM96] uses information derived from a specification to assist testing. When formal specification is used, it is possible to automate specification-based testing through the semantics of the specification formalism. Specification-based testing includes black-box test case generation from specifications.

Some approaches in the specification-based testing of real-time reactive system were given in the thesis of Mao Zheng [Mao01]. In this thesis, a Homorphism theorem was given that served as a basis for automated test case generation from the grid automaton associated with extended state machine formalism. Furthermore a number of algorithms are also given for generating test cases for black box testing of reactive class implementations, implementations of

class refinements and system configurations. The testing relies on the quality of the specification of the system.

1.4 Purpose

The main goal of this report is to develop a specification-based testing system for real-time reactive system in TROMLAB framework (TROM-SBTS) from the object–oriented design specifications. The system developed includes unit testing, pair testing, and system testing. The implemented specification-based testing methods for real time reactive system are described in [Mao02]. The main achievements of my report are:

- Studying the algorithm proposed by Mao Zheng [Mao02], and modifying Mao's algorithm;
- Implementation of the updated algorithm;
- Testing the implementation on the train-controller-gate cases study and comparing the empirical results to the theoretical results are reported in [Mao02];
- Integrating the existing softwares on unit testing and developing a new testing system.

Chapter 2 Specification Based Testing for Reactive Systems

2.1 Introduction

This chapter is a brief survey of the basics of specification based testing for a reactive real-time system, and the chapter also introduces the concepts and terminology used in the rest of the paper.

A reactive system is composed of several synchronously communicating TROM objects. We use the specification of each TROM [Ach95] class and the specification of the reactive system consisting of objects from the classes to generate test cases to test the implementation.

2.2 TROM: a Generic Reactive Model

A generic reactive object [Ach95] is an 8-tuple (P, E, Θ, X, $\pounds$, Φ, Λ, Γ) such that: P is a finite set of port-types, E is a finite set of events and includes the silent-event ticket; Θ is a finite set of states; X is a finite set of typed attributes; $\pounds$ is a finite set of LSL traits introducing the abstract data types used in X; Φ is a function-vector (Φ_s,Φ_{at}) where Φs: $\Theta \rightarrow 2^{\Theta}$ associates each states θ with a set of states, which, possibly empty, called sub states, and Φ_{at}: $\Theta \rightarrow 2^{X}$ associates each state θ with a set of attributes, possibly empty, called active attribute set; Λ is a finite set of transition specifications, and Γ is a finite set of time-constraints. The language for describing a generic reactive class is derived from the formal definition of class.

A TROM object is represented with template for a class specification, which is known as below (figure 2.01):

Class<name>

Events:

States:

Attributes:

Traits:

Attribute-Function:

Transition-specifications:

Time-Constraints:

End

Figure 2.01: Specification of Class Template

2.3 Grid Automaton- Equivalent to TROM

In abstracting a model of a reactive entity, after discretizing each clock region to a finite number of integer value, we describe an object with a grid automaton, a finite state machine, in which every state is augmented with ports attributes, logical assertions on the attributes, and time constraints, to describe the real object behavior. The grid automaton has only finite number of states and is equivalent in behaviors to the TROM. In the automaton, we use object oriented concept in the states, which can be inherited or can be encapsulated by another automaton or states. So a complex state could be an encapsulation of another finite machine, with an initial state, and the state can include other complex states.

The grid automaton Gd(A), corresponding to the TROM state machine A is the automaton (Θ, θ_0, L, T), where

State $\Theta = \{<s, v'> \mid <s,v>$ is an extended state of A *and* $v'=v+kd$, $v' < \infty$, $k>=0\}$

Initial State $\theta_0=<s_0,v_0>$, *the extended initial state in A*

Transition Label $L=\mathcal{E} \cup \{d\}$

Transition Function T: $\Theta X\ L \rightarrow \Theta$ defined by

$T(<s_i,v_i>, e) = <s_{i+1},v_i>$ if $e \in \varepsilon$

$T(<s_i,v_i>, d)= (<s_i,v_i+d)$

Subject to the following conditions:

- The transition $<s_i,v_i> \xrightarrow{e} <s_{i+1},v_i>$ in A, where e is not a time constraint event, with or without a clock initialization, corresponds to the transition $<s_i,v'_i> \xrightarrow{e} <s_{i+1},v'_i>$ of $G_d(A)$.
- If the transition $<s_i,v_i> \xrightarrow{e} <s_{i+1},v_{i+1}>$ where e is event constrained to occur within the time shift [l,u], then $v_i + l <= v_i + u$ holds.

2.4 Testing Method and Concepts

There are three stages in testing a real-time reactive system developed in TROMLAB framework. Each stage focuses on testing the artifacts developed in one tire of TROM architecture. During the first stage, abstract data type implementations are tested [AC95, Pro96]. The second stage has two parts: the generic reactive class implementations are tested in the first part of testing, which we refer to, in the object oriented terminology, as unit testing; implementations of inherited classes are tested in the second part. During the third stage a reactive system composed of classes is tested.

Test Cases: a sequence of events e_0, e_1..., e_i, such that the transition triggered by the sequence of events form a path from the initial state to the test state.

State Cover (SC): it is a minimum set of test cases which are required to identify each state in the design to some states in the implementation.

Transition Cover (TC): it is a set of test cases required to identify each transition in the design to some actions in the implementation.

The test suite for testing a TROM is defined by $T=SC \cup TC$.

Chapter 3 System Design of TROM-SBTS

3.1 Introduction

The system, specification-based testing system for real-time reactive systems in TROMLAB environment (TROM-SBTS), will be developed in the project. The chapter describes the system architecture and detail design of the system.

3.2 Description of the System

The TROM-SBTS inputs system class specification files of an object, and outputs the test cases of the object, pair testing object, which is a sub system object that contains information of objects pair with communication, and system objects. The system will be developed in Java whose advantage of portability is taken into consideration. Therefore, the system-specific JVM and JDK have to be used. The required minimum Java version is 1.2. The following introduces the implementation of the system, which is implemented in Java programming language. It contains unit testing, pair testing, and the system testing. The unit testing part includes three subsystems: Spec_Parser, Grid_Automaton Unit _Test_Case_Generator, and AutomatonClaberation. The unit testing module gets specification input file from TROMLAB, then outputs a test cases file. The pair testing part introduces automatonClaberation, which composes two objects into one composite object, and then reuse Grid_Automaton, and Unit_Test_Case _Generator to generate grid automaton and test cases.

3.2.1 Spec_Parser

The Spec. Parser receives the input file, the class specification file of an object, parses the file, and create an object of an intermediate data structure, named "TROM", which contains the equivalent information of the input file.

3.2.2 Grid_Automaton

The Grid Automation processes the data structure "TROM" and generates a data structure representing the Grid Automaton, in which there are no time constraint components, of the class component in real time system.

3.2.3 Unit_Test_Case _Generator

The Unit Test Cases Generator produces the system output containing the test cases from the Grid Automation data structure.

3.2.4 AutomatonClaberation

The AutomatonClaberation combines two TROM objects for pair testing into one TROM object that contains the informations of two objects.

3.3 Architecture diagram

The architecture diagram of the specification-based testing system is shown in the Figure 3.01.

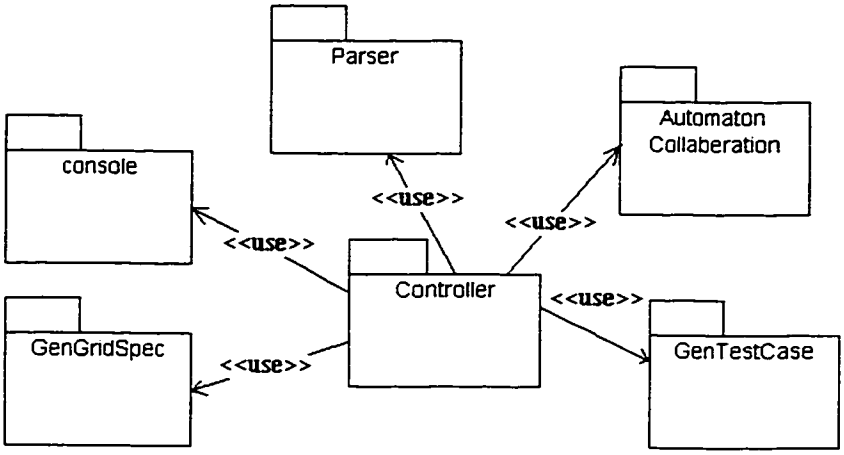

Figure 3.01:The Architecture Diagram

3.8 Interface Design

3.8.1 Unit Testing Interface Design

The Spec_Parser creates a TROM object as an input of Grid Automation. The Grid_Automaton puts more information in the object as an input of Unit_Test *Cases*_Generator, GenTestCase, and in the MyTrom object the time constraints have been decomposed. The test cases generator produces the test cases profiles, for which data flow diagram is given in Fig. 3.02.

3.8.2 Pair Testing Interface Design

In the pair testing, two TROM objects, which share some events, are inputted into AutomatonClaberation and a composite object is outputted, and the object has similar characters with a simple object, such as states list, transitions list, time constraints list, and so on. Then, the controller uses Grid_Automation

and Unit_Test_Cases_Generator to produce a grid automaton and the test case file for the object. The data flow diagram is shown in Figure 3.03.

3.8.3 System Testing Interface Design

In our OO design architecture, we divide all objects into object pairs that have some share events. For each pair of objects of them, the controller to do pair testing, in which a composite object is created. We continue to find an object with share events with the composite object and to do pair testing, in which another composite object is produced. We continue the process till only a composite object left, which is called system object. Then the grid automaton and the test cases profile are produced for the system object. The data flow diagram of system testing is given in Figure 3.04.

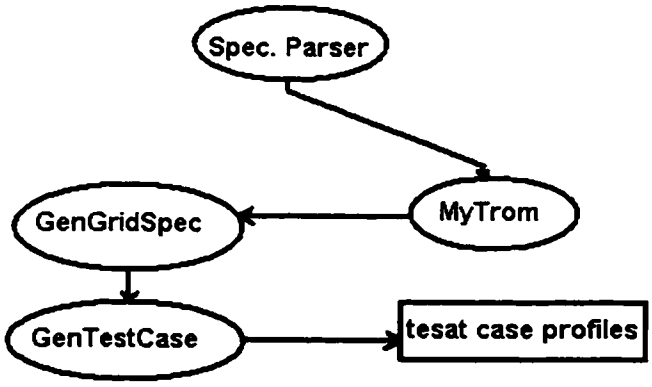

Figure 3.02: The Data Flow Diagram of Unit Testing

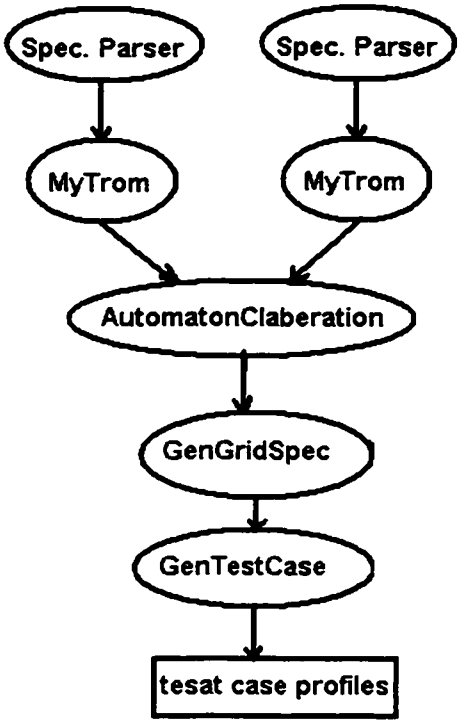

Figure 3.03: The Data Flow Diagram of Pair Testing

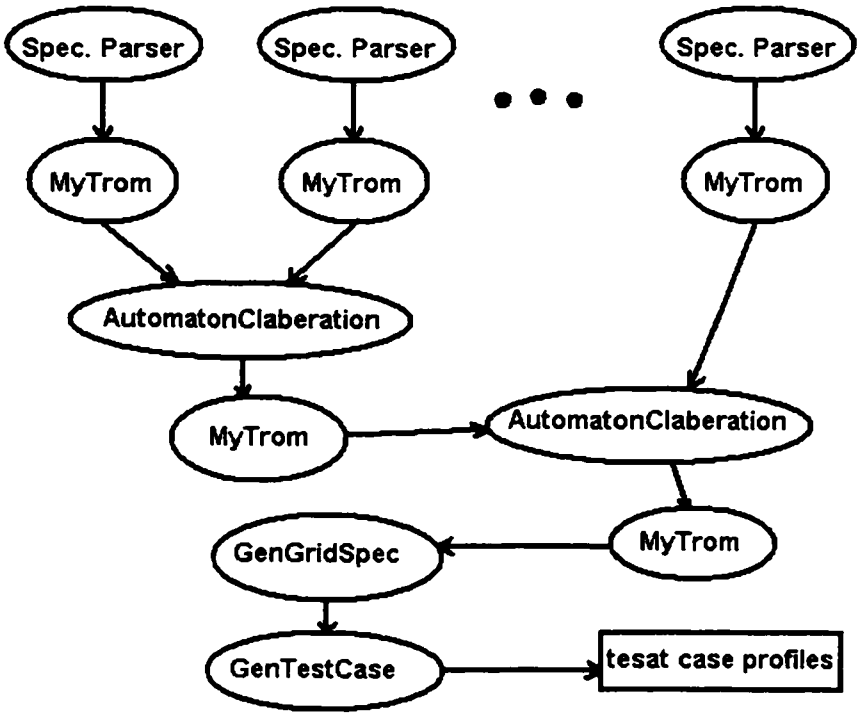

Figure 3.04: The Data Flow Diagram of System

3.9 Class diagram of specification-based testing system

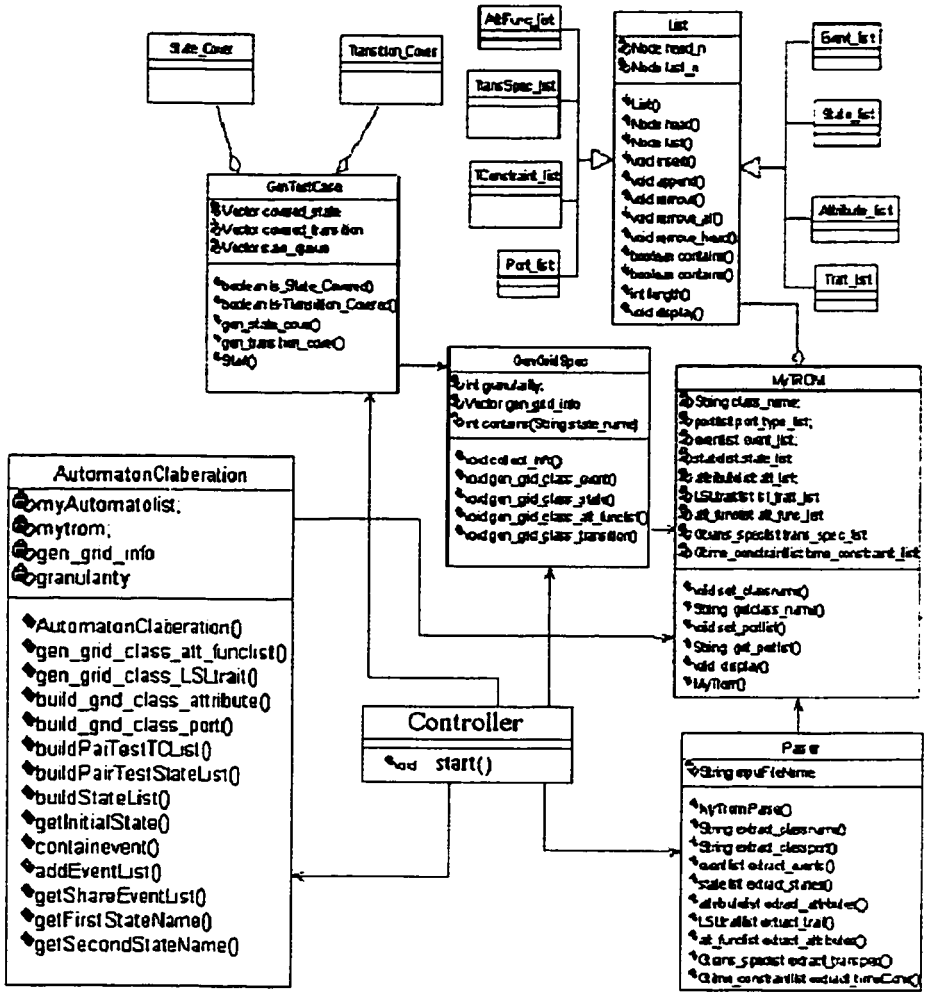

Figure 3.05: The Class Diagram of System

3.10 Sequential diagram of unit testing, pair testing, and system Testing

3.10.1 Sequential diagram of unit testing

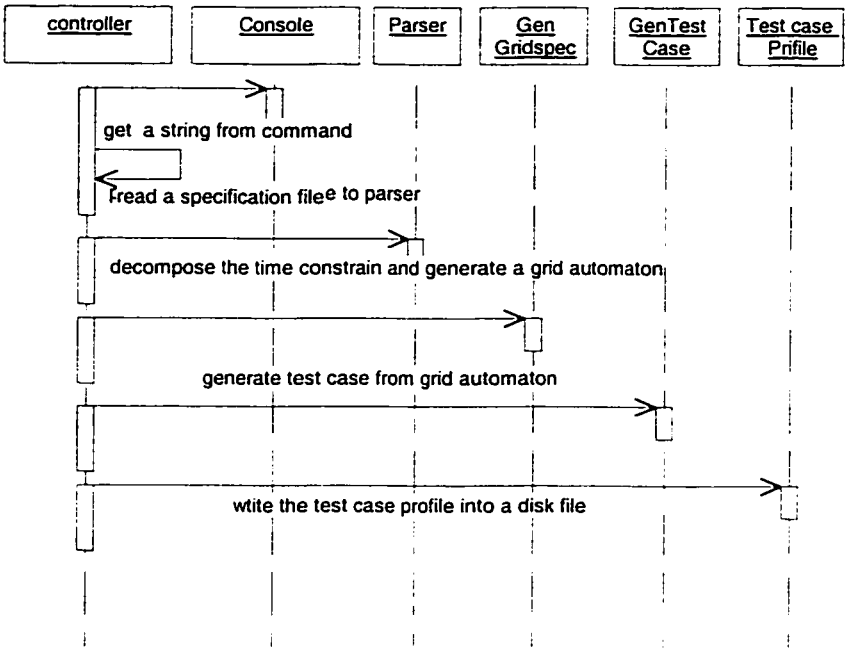

Figure 3.06: Sequential Diagram of Unit Testing

3.10.2 Sequential diagram of unit testing

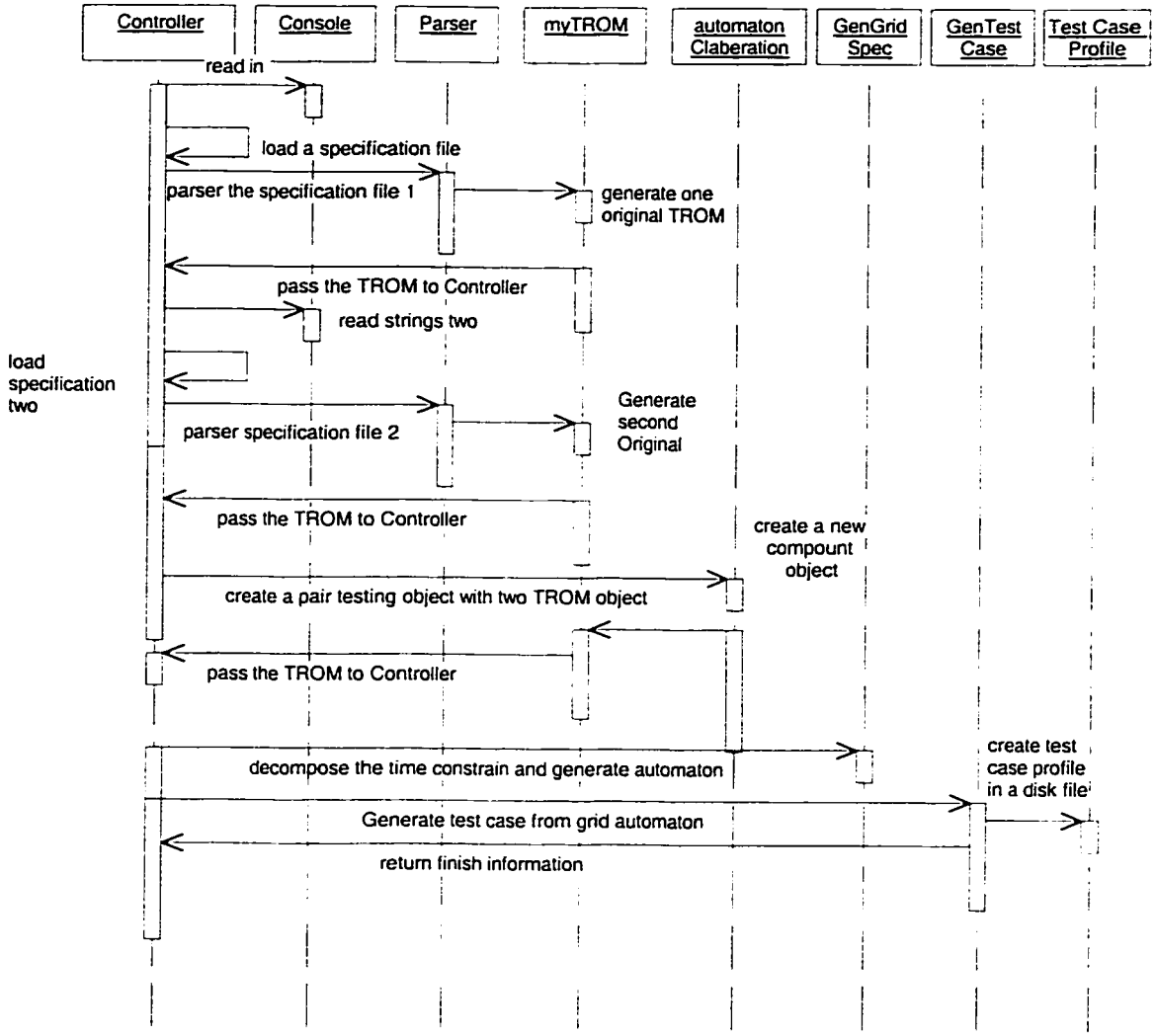

Figure 3.07 Sequential Diagram of Pair Testing

3.10.3 Sequential diagram of System Testing

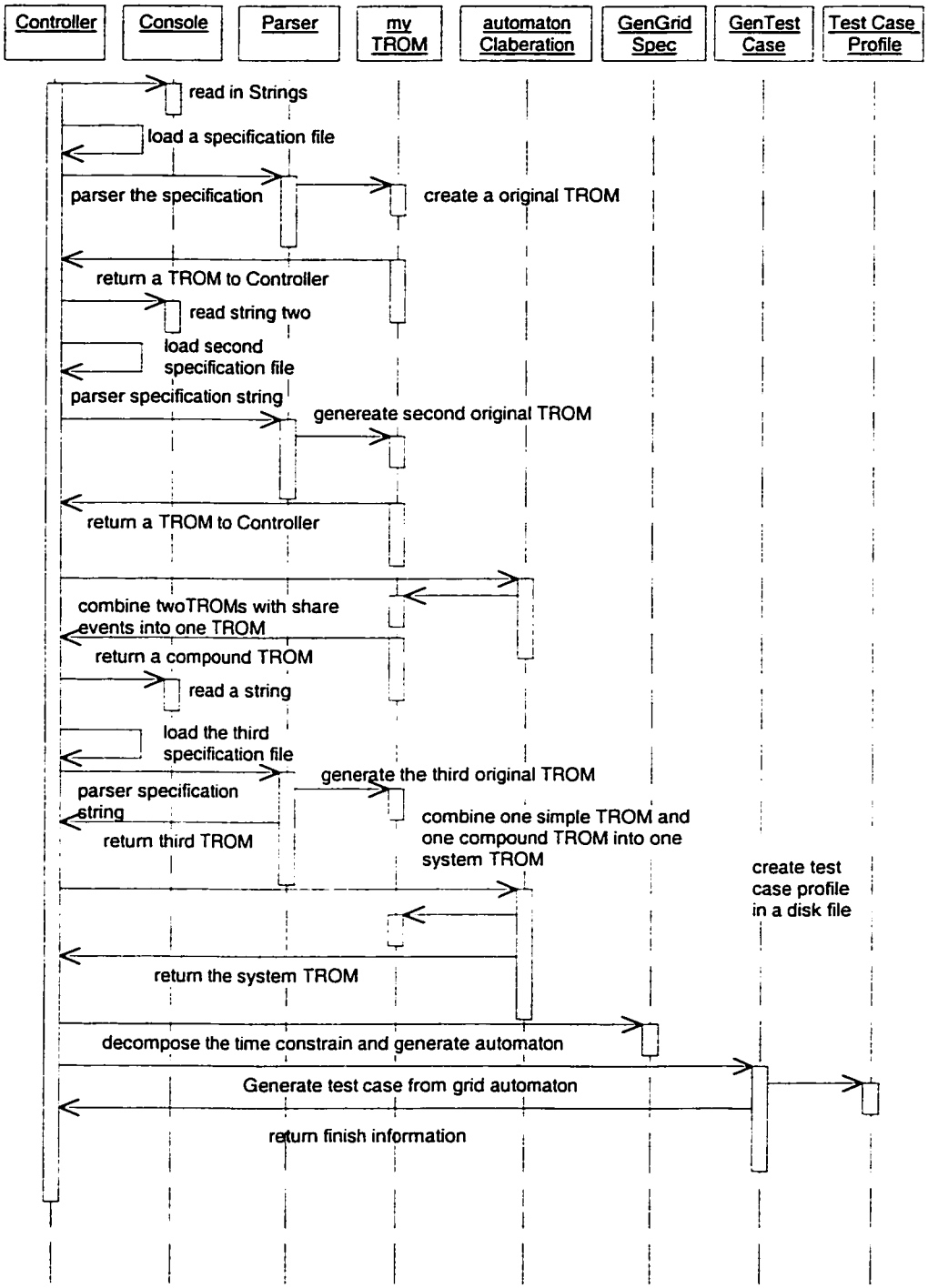

Figure 3.08: Sequence Diagram of System Testing

Chapter 4 Common Algorithms in Specification Based Testing System

4.1 Introduction:

In this chapter, some general algorithms and their implementation are introduced. These algorithms are used in unit testing, pair testing and system testing. In unit testing, after Spec_Parser generates an initial TROM object from class specification file, we use algorithm GA (Grid Automaton Generation) to generate a grid automaton from the original TROM, in which its time constraints are decomposed, and then we use test case generation algorithm to product test case profiles. In the pair testing and system testing, after producing system TROM object from a pair of objects, we also reuse the same algorithm to product test cases for the object.

4.2 Algorithme GA: Grid Automation Construction

The algorithm accepts a TROM class specification before time constraints decomposition, and outputs a grid automaton specification.

Step 0: *Initialize*:

ES: events of TROM; SS: states of TROM; TS: transition specifications of TROM.

Step 1: *Find the transition specification of TROM constrained by* TC_i:

Ri:<s_i, s_j>; e(port condition); enabling condition =>poste condition

Step 2: *Compute the grid points:* $< s_i, v' +0 >, < s_i, v' +d >,..., < s_i, v' +md >$, where m is the largest position integer for which $md=t_j$;

Step 3: *Add the internal event d to ES; remove si from SS, and add the grid points computed in Step2 to SS. Attribute-function section (active attributes) are not changed;*

Step 4: *Do the following changes to TS*:

Step 4.1: For n, 0<=n<=tj-d, increased by d each time, add transitions (time increment) to TS: $<< s_i, v' +n >,< s_i, v' +n+d >>$; d(true); true=>true;

Step 4.2: Remove the transition from TS:

Ri:$<s_i, s_j>$; e(port condition); enabling condition =>poste condition

Step 4.3: For n ti<=n<=tj increased by d each time, add the transitions (time constraint) to TS: $<< s_i, v' +n >,< s_i, v' +n+d >>$ e(port condition) enabling condition =>post condition;

4.2.1 Pseudo Code:

The main function of implementation of the algorithm is like below, which include some functions, whose pseudo code will be given after the function start ().

Start ()

Begin

Collect information of original TROM ();

Name ← generate grid class name ();

generate grid class port();

generate grid class event();

generate grid class state();

generate grid class attribute();

generate grid class LSL trait();

generate grid class attribute function list();

generate grid class transition();

End

Figure 4.01: The Pseudo Code of Start

The pseudo code of functions in the start are listed below:

collect information of original TROM()

begin

k ← generate granularity;

get the time constraint related transition and write into the grid information list

end

Figure 4.02: The Pseudo Code of Collect Information of Original TROM

generate grid class port()

begin

port element ←read the original port list element;

write the element into the generating port list;

end

Figure 4.03: The Pseudo Code of Generate Grid Class Port

generate grid class event()

begin

event ←read the original event list element;

write the element into the generating event list;

```
end
```

Figure 4.04 The Pseudo Code of Generate Grid Class Event

```
generate grid class attribute()
begin
    attribute ←read the original attribute list element;
    write the element into the generating attribute list;
end
```

Figure 4.05: The Pseudo Code of Generate Grid Class Attribute

```
generate grid class LSL trait()
begin
  LSL trait ←read the original LSL trait list element;
  write the element into the generating LSL trait list;
end
```

Figure 4.06 The Pseudo Code of Generate Grid Class LSL Trait

```
generate grid class attribute function list()
begin
  afo ←read current attribute function of original attribute function list;
  if ( the grid information list contain the state of the attribute function )
  begin
      for every d (1/k) unit, create a attribute function af;
      write af into the generating attribute function list;
  end if
  else
  begin
```

```
    create an attribute function that equal to afo;
    write afo into the generating attribute function list;
  end else
end
```

Figure 4.07: The Pseudo Code of Generate Grid Class Attribute Function List

generate grid class state()

```
begin
  while(st ← read in current state of original state list)
  begin
    if ( the grid information list contain the state)
    begin
      for every d unit of time constraint terminal
      create a new state named st's name+d*l;
      write the state into the generating state list;
    end if
    else
      create a state same as st and write into the generating state list;
    next node;
  end while
end
```

Figure 4.08 The Pseudo Code of Generate Grid Class State

generate grid class transition()

```
begin
  while(st ← read in start state of current transition of original transition list)
```

```
begin
    if ( the grid information list contain st)
    begin
      for every d unit of time constraint terminal i
          for every j between o to k-1
              begin
              create a new transition tt, which source state is st's name+(k*i+j)/k;
              if (the original transition is circle transition)
                  set tt's destination state as source state is st's name+(k*i+j)/k
              else
                  begin
                  copy the original destination state name into tt's
                  write the tt into the generating transition list;
                  end else
              create new transition t1 which content is same as st;
              modify t1's source state as st'source name+ ((i-1)*k+ j)/k;
              if st is circle transition
                  set st's destination state as st's source name+ ((l-1)*k+j)/k
              else
                  no change in the st's destination sate
              write t1 into new transition list
     end for
     else, which means the start state is not time constraint related
```

```
begin
  if st's destination state is time constrained
  begin
    if the current min time less min time or synchronic equals 0
    begin
      create a new transition t1, which equals st
      modify t1's destination state as st's destination name+ current
      time*k/k;
      write t1 into new transition list
    end if
    else
    begin
      for every integer number i between current min and min time
        for any number between 0 and k
        begin
          create a new transition t1, which equals st
          modify t1's destination state as st's destination name+ (i*k+ j)/k;
          write t1 into new transition list
        end for
        create a new transition t2, which equals st
          modify t2's destination state as st's destination name+ ((l-1)*k+
          j)/k;
          write t2 into new transition list
```

end if

else, which means both start and destination state are not time constrained;

copy the transition into the new transition list;

end

for any transition that its source state or destination state is time constrained

begin

if st's source state is time constrained

begin

create a new transition t2,

*set t2's source state as st's source name+ (i*k+ j)/k;*

*set t2's destination state as st's source name+ (i*k+ j+ 1)/k;*

write t2 into new transition list

end

else

begin

create a new transition t2,

*set t2's source state as st's destination name+ (i*k+ j)/k;*

*set t2's destination state as st's destination name+ (i*k+ j+ 1)/k;*

write t2 into new transition list

end else

end else

end if

next node;

end while

end

Figure 4.09: The Pseudo Code of Generation of Transition List

4.3 Algorithm TC: Generating Test Cases from Grid Automaton

The algorithm includes two algorithms: one is state coverage generation algorithm; the other is transition coverage generation algorithm.

4.3.1 STC: State Coverage Algorithm

The algorithm is used to generate state cover from grid automaton. For every state θ_i, the algorithm generates a set of sequences of events that brings the grid automaton from its initial state θ_0 to θ_i.

Step1. Initially, Θ: set of states; Θ_0: initial states

$\Theta = \{ \theta_0, \theta_1, \ldots, \}$, $\Theta_0 = \{ \theta_0 \}$, $l=0$; C_i: set of event sequences covering θ_l, $C_0 = \{ \varnothing \}$

Step2. Remove θ_i from Θ if Θ become empty, stop, SC=C, $C=\cup'_{k=0}C_k$

Step3. $l=l+1$

$C_i = \{ c=p.q \mid p \in C_{i-1}$, p is the test sequence for covering θ_{l-1} and q is labeling the transition $\theta_{l-1} \rightarrow \theta_l \}$

Step 4. Go to step 2.

4.3.1.1 Pseudo code:

void generate state cover()

```
Begin // initial the parameters
Sl← get grid automaton's state list;
St← find the initial state of Sl;
Push St into State queue;
While(state queue is not empty)
Begin // generate the state cover
  statenode ← delete the head of state queue;
  if statenode is not in the cover state list
  begin
    add it into cover state list;
    add it into save queue;
  end if
  if the loop is not first turn in the while loop
  begin
      display the statenode name
      for all the transition in the statenode
      begin
        display all the events in the transition ;
        if the transition is not in the transition cover
            add the transition in the transition cover;
      end for
      while it is not the end of transition list of the automaton
      begin
```

```
    if current transition's source state is statenode
    begin
      create a new state node which's name is destination state of the
      transition;
      add the transition into transition set of the state node;
      add the state node into the state queue;
    end if
    get next state;
  end while
 end if
end while
end
```

Figure 4.10: The Pseudo Code of State Cover Generation Algorithm

4.3.2 TRC: Transition Coverage Algorithm

This algorithm is used for transition cover generation. For every transition $\theta_l \rightarrow^e \theta_j$, the algorithm generates a set of sequences of events x, y=x.e such that x brings the grid automaton from its initial state θ_0 to θ_j. The event sequences will completely cover all the transitions except those that are already covered in the state coverage SC.

Step 1. Initially, T: set of transitions; i=0,

For every transition R_i: $\theta_l \rightarrow^e \theta_j$ D_i is the set of event sequences covering the transition R_i: $D_i=\{t.e \mid t \in Ci, t.e \in SC\}$

Step 2. Remove transition Ri from T, if T become empty, stop, TC=D, $D=\bigcup^i{}_{k=0} D_k$

Step3: i=i+1

Step 4. Go to *Step 2.*

4.3.2.1 Pseudo Code:

While it is not the end of transition list of the automaton

Void transition cover generation()

begin

ts← current transition;

if the transition cover contain ts

begin

for the element statenode which equal ts' source state in the save queue

begin

output transition into transition cover;

output statenode all transition event into transition cover;

end for

end if

get next transition;

end while

Figure 4.11: The Pseudo Code of Transition Cover Generation Algorithm

Chapter 5 Unit testing Algorithm and Implementation

5.1 Introduction

In the chapter, the unit testing algorithms and their implementation details are introducted.

In our specification based testing methods, we use object-oriented architecture in our test method. Unit testing is basic testing method, and system testing is developed on the unit testing. In the Unit Testing, after generating an original TROM from specification file of the object's class, which is described below, the algorithm GA (Grid Automaton Generation) generates a grid automaton file from the TROM. And then the test state covers and transition covers are generated from the grid automaton file.

5.2 TROM Generator

The Spec_Parser outputs a file as an input of Grid_Automation. The class is used to translate a class specification file to an original MyTrom file, which can be read by Grid_Automation subsystem. Grid_Automation subsystem outputs a grid MyTrom file as an input of Unit_Test_Cases Generator. Thus the subsystem translates a MyTrom original file to MyTrom grid file.

We consider MyTrom class as the interface between Spec_Parser and Grid Automation, and MyTrom class as the interface between Grid_Automation and Unit_Test_Cases_Generator. Grid_Automation can get a MyTrom file from the interface class MyTrom. Unit_Test_Cases Generator can get a MyTrom grid file

from the interface class MyTrom. The interface design diagram is shown as Figure 5.01.

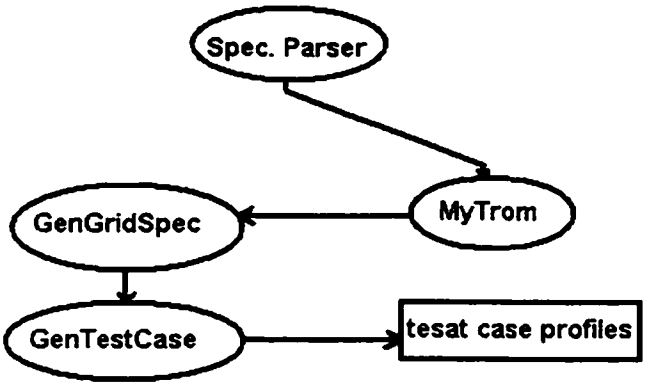

Figure 5.01 Unit Testing DFD Diagram

5.2.1 Pseudo Code:

Parser ()

Begin

Extract class name and set into the class name of the TROM;
Extract ports and set into the port list of the TROM;
Extract events and set into the event list of the TROM;
Extract states and set into the state list of the TROM;
Extract attributes and set into the attribute list of the TROM;
Extract traits and set into the LSL trait list of the TROM;
Extract attribute functions and set into the attribute functions list of the TROM;
Extract transition and set into the transition list of the TROM;
Extract time constraints and set into the time constraint list of the TROM;
return the TROM;

End

Figure 5.02 The Pseudo Code of Parser

Chapter 6 System Testing Algorithm and Implementation

6.1 Introduction

This chapter will give a solution of system testing after analyzing and updating the solution reported by Mao Zheng [Mao02], and then give the algorithm and the pseudo code.

6.2 System Testing Algorithm

From the thesis [Mao02], two kind of system testing methods are developed. One is pair testing from grid automatons: algorithm SP and Synchronous Product for a System (PP95-104); the other is generating test case from object's test case: Partition Algorithm and Algorithm GSP (Pp104-117). For the first one, the system testing bases on the pair testing, which is algorithm SP. In the algorithm described in [Mao02], the input finite state machines have three parameters, which are states set S; events set E, and transitions set T. The finite state machines are generated from TROM by decomposing its time constraints. In the page 104, the author points out the approach's drawbacks:

- The grid automaton of the synchronous product machine (involving 3 or 4 machines) will be huge, especially when the number of clocks in the system modeled by the product machine is high;
- There is a potential for exponential growth in the generation of the test case sequencing, making it impractical to test the system satisfactorily.
- The test case generated during the unit-testing phase cannot be used for system testing.

In order to overcome the drawbacks in her algorithm, I use the TROM to replace the automaton in the pair testing to reduce the number of states and transitions because the numbers of states and transition in automaton are much bigger than in TROM. After combining two TROMs into one TROM, The algorithm Grid Automaton is used to generate its grid automaton from the TROM, and then test case algorithms is used to generate its state cover and transition cover from grid automaton. My programs produce same results as in Mao's approach (in PP100 and in PP103); the result is described in the chapter 7, the case study.

6.3 The algorithm SP

The algorithm generates a TROM for pair test from two TROMs.

Input: Finite states machines of TROM are $M_1=(P_1, E_1, S_1, A_1, AF_1, LSL_1, T_1, TC_1)$, $M_2=(P_2, E_2, S_2, A_2, AF_2, LSL_2, T_2, TC_2)$, the P_1 and P_2 are port-types set, E_1 and E_2 are events set, S_1 and S_2 are state set, A_1 and A_2 are attribute set, AF_1 and AF_2 are attribute function set, LSL_1 and LSL_2 are LSL trait set, T_1 and T_2 are transition set TC_1 and TC_2 are time constraints set.

Output: Synchronous Product Machine $M=M \oplus M$, $M=((P, E, S, TA, LSL, T, TC)$.

Step1 Initialization

Step1.1 construct the initial state set $NU=\{(s_0^*, s0'^*)\ s_0^* \in S_1, s_0'^* \in S_1\}$

Step1.2 construct the share event set $ES=\{e_1\ e_1 \in E_1 \cap E_2\}$;

Step1.3 S: set of states in the product machine. These are the states that have been reached by transitions.

$S=\varnothing$

Step1.4 T: set of transition. T=∅

Step2: Construct the states set and transitions set which relate with the share event set.

Step2.1 Add the initial set into S,

Step2.2 Add the share events related states into the S if the states don't exist in S.

Step2.3 Add the share events into event set;

Step 3 Construct the internal events related states into S

Step3.1 Search from the states set; for any state in the set such as (si, si'), get the TROM's transition sets T_1, and T_2, and then search to find the output transitions t_i and t_i' which's source states are given states s_i and s_i'. The related events are e_1 and e_2, the destination states which t_i and t_i' lead to are s_j and s_j',

Step3.2 For each internal event $e_1$search the M_1's time constraints set, to find if e_1 has any time constraints, if yes, let the time constraint be tc_1. For each internal event e_2, search the M_2's time constraints set, to find if e_2 has any time constraints; if yes, let the time constraint tc_2. If e_1 and e_2 both have time constraints, which are $[u_1,v_1]$, $[u_2,v_2]$, if $u_1<u_2$ add the state (s_j, s_i') into S and transition $<(s_i, s_i'), (s_j, s_i')>$ ne1 into T; if $u_1> u_2$, add the state (s_i, s_j') into S and transition $<(s_i, s_i'), (s_i, s_j')> e_2$ into t; else add the state (s_j, s_i') and (s_i, s_j') into S and transition $<(s_i, s_i'), (s_j, s_i')> e_1$ and $<(s_i, s_i'), (s_i, s_j')> e_2$ into T;

Step4 Construct port-types set, attribute set, attribute function set, LSL trait set, and time constraints set.

6.3.1 Pseudo code:

AutomatonClaberation (MyTrom a1, MyTrom a2)

Build event list ();

Build pair test state and transition list ();

Build pair test time constrain list ();

Build grid automaton class port();

Build grid class attribute list();

Build grid class attribute function list();

Build grid class LSL trait list ();

End

Figure 6.01: The Pseudo Code of AutomatonClaberation

Build pair test state and transition list ()

Begin

Build share events list ();

Construct state nodes and transition nodes from shared events related nodes ();

Add internal event related to state into state list of system TROM();

End

Figure 6.02 The Pseudo Code of Build Pair Testing State and Transition List

Construct state nodes and transition nodes from shared events related nodes ()

begin

Statelist 1 a1←getstatelist from TROM 1;

Statelist 2 a2←getstatelist from TROM 2;

State s1 ← get initial state from a1;

State s2 ← get initial state from a2;

State s ←construct a pair test state from s1.and s2

Install state s in the state list of pair test TROM node

Event list el1←construct shared event list (TROM 1, TROM 2);

Search every event of el1;

While(not the end of el1)

begin

event e1 ← get current event(el1);

search transition list of TROM1

while(the event of current transition node not equal to e1)

continue;

t1← get current transition(TROM 1);

search transition list of TROM2

while(the event of current transition node not equal to e1)

continue;

t2← get current transition(TROM 2);

state s1← get source state from TROM 1;

state s2 ← get source state from TROM 2;

State s ←construct a pair test state from s1.and s2

If (s is a new one)

Install state s in the state list of pair test TROM node

state s1← get destination state from TROM 1;

state s2 ← get destination state from TROM 2;

```
    State s ←construct a pair test state from s1.and s2
    If ( s is a new one)
      Install state s in the state list of pair test TROM node
next element
end while
end
```

Figure 6.03: The Pseudo Code of Construct State Nodes and Transition Nodes from Share Events

```
construct share event list ( TROM 1, TROM 2)
begin
  event list el1 ← get event list from TROM 1;
  event list el2 ← get event list from TROM 2;
  search event from el1
  while(is not end of el1)
  begin
    event e1 ← current event of el1;
    search event from el2
    while(is not the end of el2 and current event of el2 not equal to e1)
      continue;
      if found share event
        construct an event from e1 and install into event list;
  next element of while loop
  end while
  return the event list;
end
```

Figure 6.04: The Pseudo Code of Construct Share Event List

Add internal event related to state into state list of system TROM ()

begin

state list sl1 ← get state list from TROM;

search sl1 for every state s;

state s1 ← get state 1 name from parser s' state name

state s2 ← get state 2 name from parser s' state name

search transition list of TROM1 which is el1

while (is not end of el1)

begin

transition t1 ← current transition of el1;

if (t1's source state equal to s1)

begin

state s1_1 ← get the destination state of t1;

create a pair testing state node s11 from s1_1 and s2;

create a transition node (<s1,s2>, <s1_1, s2> event) as t11

break;

end if

end while

search transition list of TROM2 which is el2

while (is not end of el2)

begin

transition t2 ← current transition of el2;

if (t2's source state equal to s2)

begin

state s2_1 ← get the destination state of t2;

create a pair testing state node s22 from s1 and s2_1;

create a transition node (<s1,s2>, <s1, s2_1> event) as t22

break;

end if

next element;

end while

search time constraint list of TROM1 which is tcl1

while(is not end of tcl1)

begin

time constraint tc1← get current time constraint from tcl1;

if (event of tcl1 equals to event of t1)

break;

end while

search time constraint list of TROM2 which is tcl2

while (is not end of tcl2)

begin

time constraint tc2← get current time constraint from tcl2;

if (event of tcl2 equals to event of t2)

break;

next element

end while

if (s1 s2 both have time constraint)

begin

if (s1's event is before s2's)

insert s11 to state list of TROM;

insert t11 to the transition list of TROM;

else

```
        insert s22 to state list of TROM; and
      insert t22 to the transition list of TROM;
    end if
  else if (s1 has time constraint and s2 has not)
    insert s11 to state list of TROM ;
    insert t11 to the transition list of TROM;
  else if (s2 has time constraint and s1 has not)
    insert s22 to state list of TROM; and
    insert t22 to the transition list of TROM;
  else
    begin
      insert s11 to state list of TROM; and
      insert t11 to the transition list of TROM;
      insert s22 to state list of TROM; and
      insert t22 to the transition list of TROM;
    end else
  end
```

Figure 6.05 The Pseudo Code of Add Internal Event Related State Into State List of System TROM

Chapter 7 Case Study of Train, Controller, and Gate Problem

7.1 Introduction

In this chapter, we introduce the railroad-crossing problem, and use it as an example to illustrate our testing methodology. The railroad-crossing problem has been considered as a benchmark example by researchers in real-time systems community [HM96]. A generalized version of this problem has been considered by Muthiayen and Alagar [MA99] to formally prove safety properties in their design. We take their verified design and generate test cases to test the correctness of code generated by Zhang [Zha00].

In the paper, I use the example to test my algorithm and program, and then discuss the result.

7.2 Description of the Scenario of the Example

According to their design, several trains cross a gate independently and simultaneously using non-overlapping tracks. A train chooses the gate it intends to cross; there is a unique controller monitoring the operations of each gate. When a train approaches a gate, it sends a message to the corresponding controller, which then commands the gate to close. When the last train crossing a gate leaves the crossing, the controller commands the gate to open. The safe operation of the controller depends on the satisfaction of certain timing constraints, so that the gate is closed before a train enters the crossing, and the gate is opened after the last train leaves the crossing.

A train enters the crossing within an interval of 2 to 4 time units after having indicated its presence to the controller. The train informs the controller that it is leaving the crossing within 6 time units of sending the approaching message. The controller instructs the gate to close within 1 time unit of receiving an approaching message from the first train entering the crossing, and starts monitoring the gate. The controller continues to monitor the closed gate if it receives an approaching message from another train. The controller instructs the gate to open within 1 time unit of receiving a message from the last train to leave the crossing. The gate must close within 1 time unit of receiving instructions from the controller. The gate must open within an interval of 1 to 2 time units of receiving instructions from the controller.

Figures 7.02, 7.04, 7.06 show the state chart diagrams for the controller, train and gate. Together, they specify the behaviour of a Train-Gate-Controller system. When there is no train in the system, the train is in the idle state, the controller is in the idle state, and the gate is in the opened state. When a train wants to pass through a crossing, it sends the message to the controller; the train and controller change their states simultaneously. In the state activate, the controller may receive "Near" messages from other trains or it sends the message "lower" to the gate. In the latter case, the controller and gate change their states simultaneously. In the state monitor, the controller continues to receive "Near" messages from other trains or monitor the gate. The train that is in the crossing state may send an "Exit" message to the controller before exiting the gate. Both the controller and the train synchronize on the event "Exit". The train

leaves the gate within 6 unit of time from the instant originally sends the message "Near" to the controller. The controller changes from monitor state only when all the train have exited from the gate. At that instant, it sends the message to the gate and returns to the idle state. The events "Down" and "Up" are time constrained internal events for the gate. The evaluation for the local clocks for the controller is the variables "TCvar1" and "TCvar2". Similarly, their local clocks shown in the extended state chart diagrams govern the timed constrained events for the controller and train. Figures 7.01, 7.03, 7.05 show the textual speciation corresponding to their extended state chart diagrams.

```
Class Train [@R]
   Events: Near!@R, Out, Exit!@R, In
   States: *idle, cross, leave, toCross
   Attributes: cr:@C
   Traits:
   Attribute-Function:
      idle →   {}; cross →   {};
      leave →   {}; toCross →   {cr};
   Transition-Specifications:
      R1: <idle,toCross>; Near(true); true=>cr'=pid;
      R2: <cross,leave>; Out(true); true => true;
      R3: <leave,idle>; Exit(pid = cr); true => true;
      R4: <toCross,cross>; In(true); true => true;
   Time-Constraints:
      TCvar2: R1, Exit, [0, 6], {};
      TCvar1: R1, In, [2, 4], {};
end
```

Figure 7.01: Formal Specification of Class Train

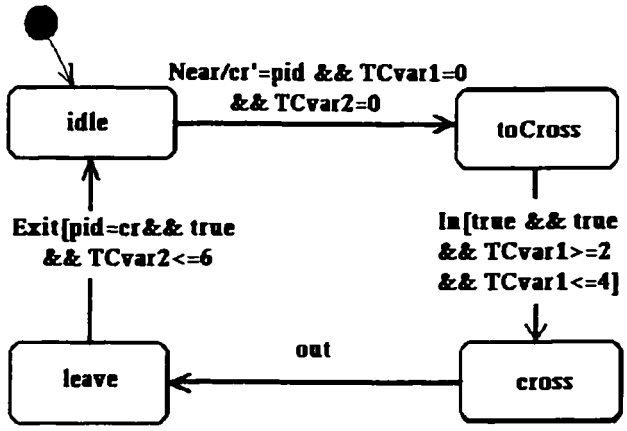

Figure 7.02 State Diagram of Class Train

```
Class Controller [@P, @Y]
    Events: Lower!@Y, Near?@P, Raise!@Y, Exit?@P
    States: *idle, activate, deactivate, monitor
    Attributes: inSet:PSet
    Traits: Set[@P,PSet]
    Attribute-Function:
        activate →  {inSet}; deactivate →  {inSet};
        monitor →  {inSet}; idle →  {};
    Transition-Specifications:
        R1: <activate,monitor>; Lower(true);
            true => true;
        R2: <activate,activate>; Near(NOT(member(pid,inSet)));
            true => inSet' = insert(pid,inSet);
        R3: <deactivate,idle>; Raise(true);
            true => true;
        R4: <monitor,deactivate>; Exit(member(pid,inSet));
            size(inSet) = 1 => inSet' = delete(pid,inSet);
        R5: <monitor,monitor>; Exit(member(pid,inSet));
            size(inSet) > 1 => inSet' = delete(pid,inSet);
        R6: <monitor,monitor>; Near(!(member(pid,inSet)));
            true => inSet' = insert(pid,inSet);
        R7: <idle,activate>; Near(true);
            true => inSet' = insert(pid,inSet);
    Time-Constraints:
        TCvar1: R7, Lower, [0, 1], {};
        TCvar2: R4, Raise, [0, 1], {};
end
```

Figure 7.03 Formal Specification of Class Controller

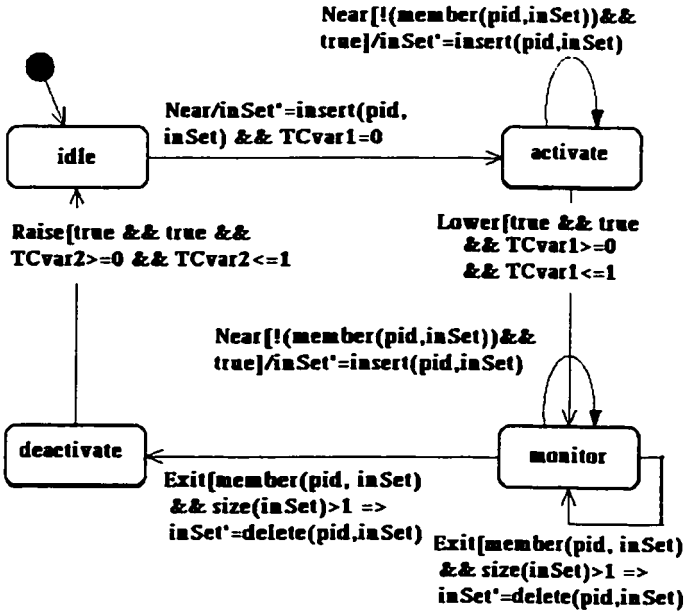

Figure 7.04 State Diagram of Class Controller

```
Class Gate [@S]
   Events: Lower?@S, Down, Up, Raise?@S
   States: *opened, toClose, toOpen, closed
   Attributes:
   Traits:
   Attribute-Function:
      opened →  {}; toClose →  {};
      toOpen →  {}; closed →  {};
   Transition-Specifications:
      R1: <opened,toClose>; Lower(true); true => true;
      R2: <toClose,closed>; Down(true); true => true;
      R3: <toOpen,opened>; Up(true); true => true;
      R4: <closed,toOpen>; Raise(true); true => true;
   Time-Constraints:
      TCvar1: R1, Down, [0, 1], {};
      TCvar2: R4, Up, [1, 2], {};
end
```

Figure 7.05 Formal Specification of Class Gate

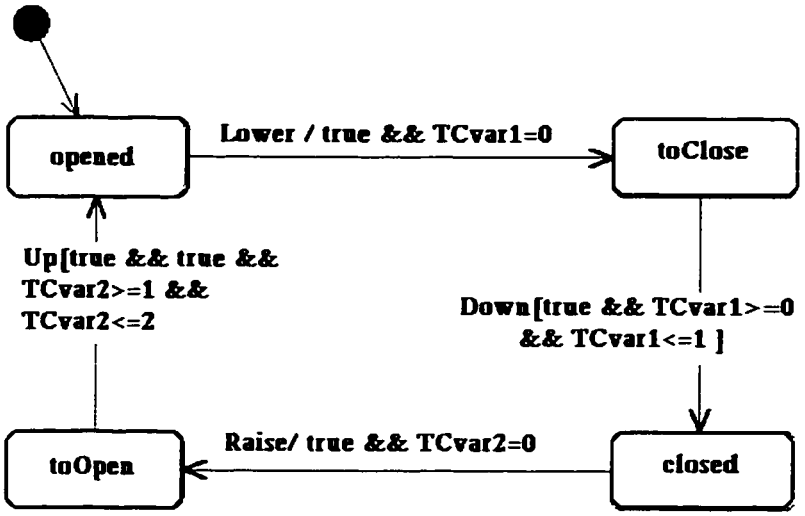

Figure 7.06: State Diagram of Class Gate

7.3 Pair Test Result of Class Gate and Controller

Between class gate and class controller, there are shared events "Lower" and "Raise". When "Lower!" event occurs in state "active", if the time constraint condition Tcvar1<=1 is true; the Controller goes into the state "monitor". The event "Lower?" occurring in gate's state "opened" will lead the object of gate from state "opened" to state "toClose". The program will create an object typed Gate_Controller. After these, the grid automaton and test case will be generated

for the object like a single object. The experiment result of the new object's state list and transition list is given in appendix 1: If we don't consider the circle transition, the state diagram of pair testing object is give below (Figure 7.07), which is same as the Mao's result (Figure 39 of P100).

7.4 The Pair Testing of Objects of Class Train and Controller

A message "Near!" will occur when a train approach the crossing, the controller receives the message, the state of train is triggered from state "idle" to state "toCross"; simultaneously, the state of controller is triggered from state "idle" to state "active". Another shared event is "Exit", the train sends message of "exit" to the controller, we create a train_controller object which represents the synchronized behaviour of train and controller, and then we use GA and TA algorithms to generate grid automatons and test covers. The result are given in appendix 2, the state diagram of pair testing object of train and controller is given below (Figure 7.08) which is same as the Mao's result (Figure 39 of 100).

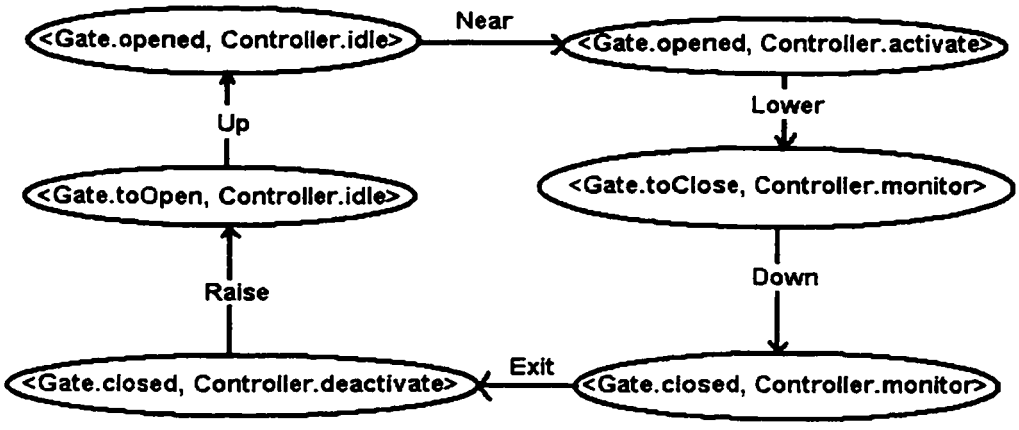

Figure 7.07 State Diagram of Pair Testing Object of Gate and Controller

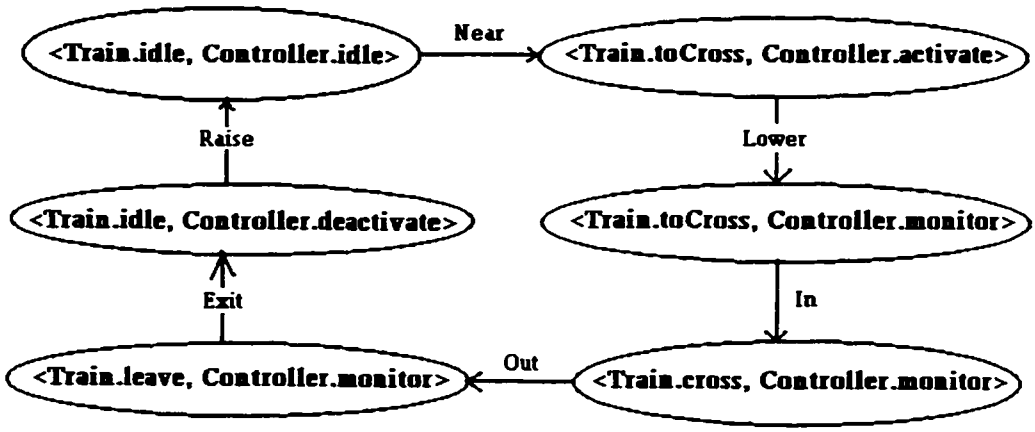

Figure 7.08 State Diagram of Pair Testing Object of Train & Controller

7.5 System Testing:

The system consists of three kinds of objects, which are train, controller, and gates. Between controller and gate, controller and train, there are shared events. We can construct system object by two ways: one is pair testing by gate and controller, in which we get an object that is the sub system of gate and controller, and then use the sub system to do pair testing with train object; the other experiment is first to do pair testing for the train and the controller, in which we get an object that is the sub system of train and controller, and then the sub system do pair testing with the gate. The result shows we get same result as Mao's result. (P103, Figure 41)

7.5.1 Experiment One: train⊗(gate⊗controller)

The experiment is to do pair test gate and controller to get a composite object first, and then the composite object does pair testing with train to get another composite object, which we call the system object. We generate grid automaton and test cases for the system object. The system object's state diagram is the same as Mao's result. (Pp 103 Figure 41) The experiment result of

states, transitions, and test cases are listed Appendix 3. The state diagram of system object is showed below, Fig 7.09, in which we use "< T_toCross, G_opened, C_activate > " to indicate the diagram " < Train.toCross, Gate_Controller. < Gate.opened, Controller.activate>> ". The other state names are indicated in same way.

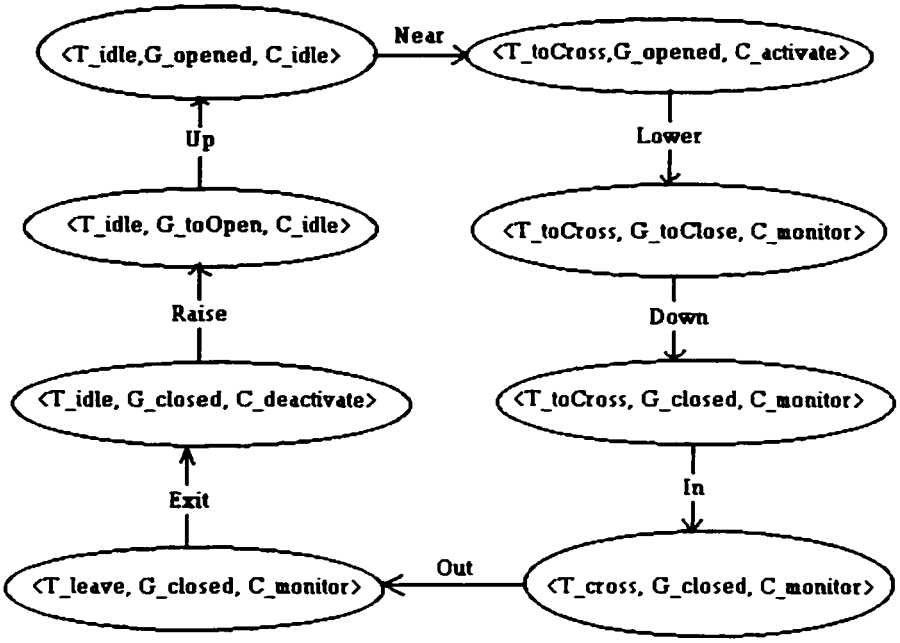

Figure 7.09 State Diagram of System Testing Object of Train, Gate, & Controller

7.5.2 Experiment Two: gate⊗(train⊗controller)

The experiment two give the system object from gate, train, and controller by first doing pair testing for train and controller to get composed object, and then doing pair testing again for the composed object with the object gate. The program creates a system object that includes information of the object of gate, controller, and gate. A grid automaton and test cases for the system object are produced by the experiment, which illustrates that same result as in the experiment one. The result is given Appendix 4. The state diagram of system is

given in Figure. 7.10, in which we use same name style as in experiment one. The result is the same as Mao's result. (Pp: 103 Figures 41)

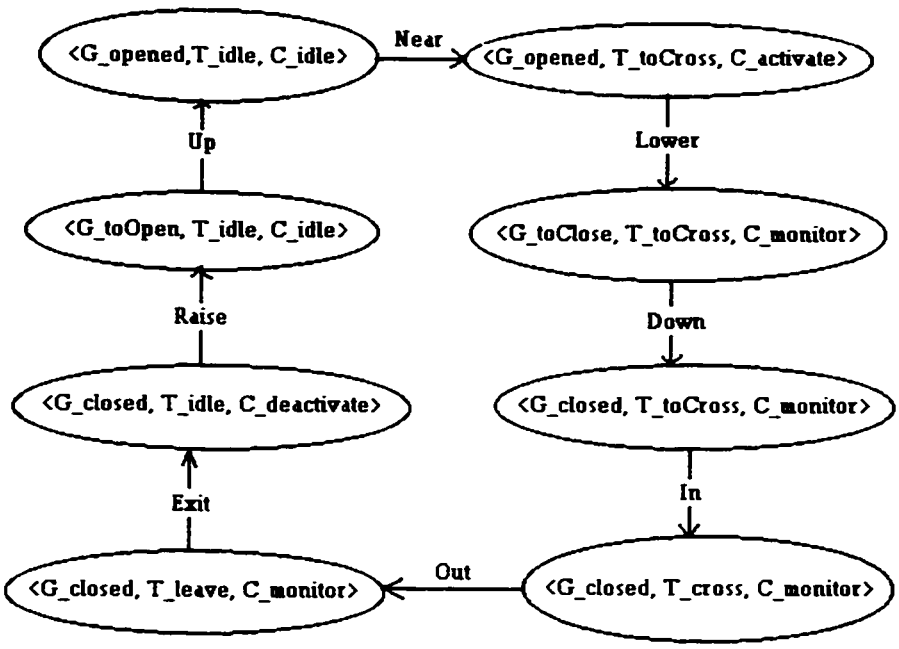

Figure 7.10: State Diagram of System Testing Object of Gate, Train, & Controller

7.6 Summary

The experiments in the major report show that pair testing and system testing which is based on pair testing and TROM are possible. It avoids the huge calculating in the pair testing of grid automatons. From the experiments, we can see the grid automaton after time constraint decomposition has huge state number comparing with states of the system TROM objects. Composing the objects' class to get the system object's TROM to get the test cases of the system is a good way, which is the experiment's result. The results in the report are same with what are reported in Mao's thesis [Mao02].

7.7 The Future Work

The specification based testing follows black box testing and bases on the specification of the class. So the testing can evaluate the design as well as the correctness of the specification.

Chapter 8 Reference

[AP98] V.S Alagar and K. Periyasamy ***Specification of Software system*** Springer-Verlag ISBN 0-387-98430-5

[PR01] Pressman, Roger S. ***Software Engineering: a Practitioner's approach*** –5th ed.p. cm-(McGraw-Hill series in computer Science) ISBN: 0-07-3655778-3

[HP85] D. Harel, A. Pnueli. ***On the Development of Reactive Systems.*** In Logic and Models of Concurrent Systems, NATO, Advanced Study Institute on Logics and Models for Verification and Specification of Concurrent Systems Spring Verlag, 1985.

[Ach95] R. Achuthan, ***A Formal Model for Object-Oriented Development of Real-Time Reactive Systems.*** PhD. thesis, Department of Computer Science, Concordia University, Montreal, Canada, October 1995

[Mao02] Mao Zheng, Concordia University, 2002 ***The Automated t Generation From Formal Specifications Of Real-Time Reactive Systems*** PhD. thesis, Department of Computer Science, Concordia University, Montreal, Canada, October 2001

[AOZ00a] V.S.Alagar, O.Ormandjieva, M.Zheng. ***Specification-Based Testing for Real-Time Reactive systems.*** In Proceedings of 34th International Conference on Technology of Object-Oriented Languages and Systems, TOOL34, pp: 25-36, IEEE Computer Society, Santa Barbara, California, USA July-August, 2000

[AZ01] V.S.Alagar, M.Zheng. ***A Rigorous Method for Testing Real-Time Reactive systems***. In Proceedings of Sixth IEEE International Conference on Engineering of Complex Computer Systems, ICECCS2000, pp:12- 24, IEEE Computer Society, Toko, Japan, September, 2000.

[AAM98] V.S.Alagar, R.Achuthan, D.Muthiayen. TROMLIB: *A Software* ***Development Environment for Real-Time Reactive Systems***. (first version 1996, revised 2001), Submitted for Publication.

[RM96] R.M.Poston, ***"Automating specification Based software Testing"***, IEEE Computer Society Press, ISBN 08186-7531-4, May 1996

[HM96] C.Heitmeyer, D.Mandrioli ***Formal Methods for Real-Time Computing***. John Wiley & Sons, 1996

[Zha00] L.Zhang. ***Implementing Real-Time Reactive Systems from Object-Oriented Design Specifications***. Master Thesis, Department of Computer Science, Concordia University, Montreial, Canada, 2000

[AC95] V.S. Alagar and A. Celer. ***Automating the Generation and Sequencing of Test Cases from Larch Specification.*** Technical Report, Concordia University, Montreial, Canada, June 1995

[Pro96] A. Protopsaltou. ***Constructing Black-box Testing Suits for Systems Specified in Larch/C++.*** Master Thesis, Department of Computer Science, Concordia University, Montreial, Canada, April 1999

[Tao96] H. Tao. Static Analyzer***: A Design Tool for TROM***. Master Thesis, Department of Computer Science, Concordia University, Montreial, Canada, August 1996

[Mut96] D.Muthiayen ***Animation and Formal Verification of Real-Time Reactive Systems in an Object-Oriented Environment***. Master Thesis, Department of Computer Science, Concordia University, Montreial, Canada, October 1996

[Nag99] D.Muthiayen ***Real-Time Reactive System Developemnt – A Formal approach based on UML and PVS***. PhD Thesis, Department of Computer Science, Concordia University, Montreial, Canada, January 2000

[Pom99] F. **Pompeo *A Formal Verification Assistant for TROMLIB environment.*** Master Thesis, Department of Computer Science, Concordia University, Montreial, Canada, November 1999

[Pop99] O. Popistas. Rose-GRC Translator: ***Mapping UML Visual Models onto Formal Specifications.*** Master Thesis, Department of Computer Science, Concordia University, Montreial, Canada, April 1999

[Sir99] V. Srinivasan. ***Graphical User Interface for TROMLIB Environment***. Master Thesis, Department of Computer Science, Concordia University, Montreial, Canada, December 1999.

[Hai99] G. Haidar. ***Reasoning System for Real-Time Reactive Systems***. Master Thesis, Department of Computer Science, Concordia University, Montreial, Canada, December 1999

Appendix 1 Result of pair testing of Gate and Controller

The class name is: Gate_Controller

State List:

<<Gate.opened, Controller.idle>, true>
<<Gate.opened, Controller.activate>, false>
<<Gate.toClose, Controller.monitor>, false>
<<Gate.closed, Controller.deactivate>, false>
<<Gate.toOpen, Controller.idle>, false>
<<Gate.closed, Controller.monitor>, false>

Transition Spec List:

CR-0 <<Gate.opened, Controller.activate>, <Gate.toClose, Controller.monitor>> , Lower;
CR-1 <<Gate.closed, Controller.deactivate>, <Gate.toOpen, Controller.idle>> , Raise;
CR-2 <<Gate.opened, Controller.idle>, <Gate.opened, Controller.activate>> , Near;
CR-3 <<Gate.toClose, Controller.monitor>, <Gate.closed, Controller.monitor>> , Down;
CR-5 <<Gate.toOpen, Controller.idle>, <Gate.opened, Controller.idle>> , Up;
CR-7 <<Gate.closed, Controller.monitor>, <Gate.closed, Controller.deactivate>> , Exit;
CR-8 <<Gate.opened, Controller.activate>, <Gate.opened, Controller.activate>> , Near;
CR-9 <<Gate.toClose, Controller.monitor>, <Gate.toClose, Controller.monitor>> , Exit;
CR-10 <<Gate.toClose, Controller.monitor>, <Gate.toClose, Controller.monitor>> , Near;
CR-11 <<Gate.closed, Controller.monitor>, <Gate.closed, Controller.monitor>> , Exit;
CR-12 <<Gate.closed, Controller.monitor>, <Gate.closed, Controller.monitor>> , Near;

Transition Spec List (after time constraint decomposed):

R_1 <<Gate.opened, Controller.activate>+0/4, <Gate.toClose, Controller.monitor>+0/4> , Lower;
R_2 <<Gate.opened, Controller.activate>+1/4, <Gate.toClose, Controller.monitor>+0/4> , Lower;
R_3 <<Gate.opened, Controller.activate>+2/4, <Gate.toClose, Controller.monitor>+0/4> , Lower;
R_4 <<Gate.opened, Controller.activate>+3/4, <Gate.toClose, Controller.monitor>+0/4> , Lower;
R_5 <<Gate.opened, Controller.activate>+4/4, <Gate.toClose, Controller.monitor>+0/4> , Lower;
R_6 <<Gate.opened, Controller.activate>+0/4, <Gate.opened, Controller.activate>+1/4> , 1/4;
R_7 <<Gate.opened, Controller.activate>+1/4, <Gate.opened, Controller.activate>+2/4> , 1/4;
R_8 <<Gate.opened, Controller.activate>+2/4, <Gate.opened, Controller.activate>+3/4> , 1/4;
R_9 <<Gate.opened, Controller.activate>+3/4, <Gate.opened, Controller.activate>+4/4> , 1/4;
R_10 <<Gate.closed, Controller.deactivate>+0/4, <Gate.toOpen, Controller.idle>+0/4> , Raise;
R_11 <<Gate.closed, Controller.deactivate>+1/4, <Gate.toOpen, Controller.idle>+0/4> , Raise;
R_12 <<Gate.closed, Controller.deactivate>+2/4, <Gate.toOpen, Controller.idle>+0/4> , Raise;
R_13 <<Gate.closed, Controller.deactivate>+3/4, <Gate.toOpen, Controller.idle>+0/4> , Raise;
R_14 <<Gate.closed, Controller.deactivate>+4/4, <Gate.toOpen, Controller.idle>+0/4> , Raise;
R_15 <<Gate.closed, Controller.deactivate>+0/4, <Gate.closed, Controller.deactivate>+1/4> , 1/4;
R_16 <<Gate.closed, Controller.deactivate>+1/4, <Gate.closed, Controller.deactivate>+2/4> , 1/4;
R_17 <<Gate.closed, Controller.deactivate>+2/4, <Gate.closed, Controller.deactivate>+3/4> , 1/4;
R_18 <<Gate.closed, Controller.deactivate>+3/4, <Gate.closed, Controller.deactivate>+4/4> , 1/4;
R_19 <<Gate.opened, Controller.idle>, <Gate.opened, Controller.activate>+0/4> , Near;
R_20 <<Gate.toClose, Controller.monitor>+0/4, <Gate.closed, Controller.monitor>> , Down;
R_21 <<Gate.toClose, Controller.monitor>+1/4, <Gate.closed, Controller.monitor>> , Down;
R_22 <<Gate.toClose, Controller.monitor>+2/4, <Gate.closed, Controller.monitor>> , Down;
R_23 <<Gate.toClose, Controller.monitor>+3/4, <Gate.closed, Controller.monitor>> , Down;
R_24 <<Gate.toClose, Controller.monitor>+4/4, <Gate.closed, Controller.monitor>> , Down;
R_25 <<Gate.toClose, Controller.monitor>+0/4, <Gate.toClose, Controller.monitor>+1/4> , 1/4;
R_26 <<Gate.toClose, Controller.monitor>+1/4, <Gate.toClose, Controller.monitor>+2/4> , 1/4;
R_27 <<Gate.toClose, Controller.monitor>+2/4, <Gate.toClose, Controller.monitor>+3/4> , 1/4;
R_28 <<Gate.toClose, Controller.monitor>+3/4, <Gate.toClose, Controller.monitor>+4/4> , 1/4;
R_29 <<Gate.toOpen, Controller.idle>+4/4, <Gate.opened, Controller.idle>> , Up;
R_30 <<Gate.toOpen, Controller.idle>+5/4, <Gate.opened, Controller.idle>> , Up;
R_31 <<Gate.toOpen, Controller.idle>+6/4, <Gate.opened, Controller.idle>> , Up;

R_32 <<Gate.toOpen, Controller.idle>+7/4, <Gate.opened, Controller.idle>> , Up;
R_33 <<Gate.toOpen, Controller.idle>+8/4, <Gate.opened, Controller.idle>> , Up;
R_34 <<Gate.toOpen, Controller.idle>+0/4, <Gate.toOpen, Controller.idle>+1/4> , 1/4;
R_35 <<Gate.toOpen, Controller.idle>+1/4, <Gate.toOpen, Controller.idle>+2/4> , 1/4;
R_36 <<Gate.toOpen, Controller.idle>+2/4, <Gate.toOpen, Controller.idle>+3/4> , 1/4;
R_37 <<Gate.toOpen, Controller.idle>+3/4, <Gate.toOpen, Controller.idle>+4/4> , 1/4;
R_38 <<Gate.toOpen, Controller.idle>+4/4, <Gate.toOpen, Controller.idle>+5/4> , 1/4;
R_39 <<Gate.toOpen, Controller.idle>+5/4, <Gate.toOpen, Controller.idle>+6/4> , 1/4;
R_40 <<Gate.toOpen, Controller.idle>+6/4, <Gate.toOpen, Controller.idle>+7/4> , 1/4;
R_41 <<Gate.toOpen, Controller.idle>+7/4, <Gate.toOpen, Controller.idle>+8/4> , 1/4;
R_42 <<Gate.closed, Controller.monitor>, <Gate.closed, Controller.deactivate>+0/4> , Exit;
R_43 <<Gate.opened, Controller.activate>+0/4, <Gate.opened, Controller.activate>+0/4> , Near;
R_44 <<Gate.opened, Controller.activate>+1/4, <Gate.opened, Controller.activate>+1/4> , Near;
R_45 <<Gate.opened, Controller.activate>+2/4, <Gate.opened, Controller.activate>+2/4> , Near;
R_46 <<Gate.opened, Controller.activate>+3/4, <Gate.opened, Controller.activate>+3/4> , Near;
R_47 <<Gate.opened, Controller.activate>+4/4, <Gate.opened, Controller.activate>+4/4> , Near;
R_48 <<Gate.toClose, Controller.monitor>+0/4, <Gate.toClose, Controller.monitor>+0/4> , Exit;
R_49 <<Gate.toClose, Controller.monitor>+1/4, <Gate.toClose, Controller.monitor>+1/4> , Exit;
R_50 <<Gate.toClose, Controller.monitor>+2/4, <Gate.toClose, Controller.monitor>+2/4> , Exit;
R_51 <<Gate.toClose, Controller.monitor>+3/4, <Gate.toClose, Controller.monitor>+3/4> , Exit;
R_52 <<Gate.toClose, Controller.monitor>+4/4, <Gate.toClose, Controller.monitor>+4/4> , Exit;
R_53 <<Gate.toClose, Controller.monitor>+0/4, <Gate.toClose, Controller.monitor>+0/4> , Near;
R_54 <<Gate.toClose, Controller.monitor>+1/4, <Gate.toClose, Controller.monitor>+1/4> , Near;
R_55 <<Gate.toClose, Controller.monitor>+2/4, <Gate.toClose, Controller.monitor>+2/4> , Near;
R_56 <<Gate.toClose, Controller.monitor>+3/4, <Gate.toClose, Controller.monitor>+3/4> , Near;
R_57 <<Gate.toClose, Controller.monitor>+4/4, <Gate.toClose, Controller.monitor>+4/4> , Near;
R_58 <<Gate.closed, Controller.monitor>, <Gate.closed, Controller.monitor>> , Exit;
R_59 <<Gate.closed, Controller.monitor>, <Gate.closed, Controller.monitor>> , Near;

Test Cases:

Class Gate_Controller

State Coverage Cases

<<Gate.opened, Controller.activate>+0/4> : Near
<<Gate.toClose, Controller.monitor>+0/4> : Near, Lower
<<Gate.opened, Controller.activate>+1/4> : Near, 1/4
<<Gate.closed, Controller.monitor>> : Near, Lower, Down
<<Gate.toClose, Controller.monitor>+1/4> : Near, Lower, 1/4
<<Gate.opened, Controller.activate>+2/4> : Near, 1/4, 1/4
<<Gate.closed, Controller.deactivate>+0/4> : Near, Lower, Down, Exit
<<Gate.toClose, Controller.monitor>+2/4> : Near, Lower, 1/4, 1/4
<<Gate.opened, Controller.activate>+3/4> : Near, 1/4, 1/4, 1/4
<<Gate.toOpen, Controller.idle>+0/4> : Near, Lower, Down, Exit, Raise
<<Gate.closed, Controller.deactivate>+1/4> : Near, Lower, Down, Exit, 1/4
<<Gate.toClose, Controller.monitor>+3/4> : Near, Lower, 1/4, 1/4, 1/4
<<Gate.opened, Controller.activate>+4/4> : Near, 1/4, 1/4, 1/4, 1/4
<<Gate.toOpen, Controller.idle>+1/4> : Near, Lower, Down, Exit, Raise, 1/4
<<Gate.closed, Controller.deactivate>+2/4> : Near, Lower, Down, Exit, 1/4, 1/4
<<Gate.toClose, Controller.monitor>+4/4> : Near, Lower, 1/4, 1/4, 1/4, 1/4
<<Gate.toOpen, Controller.idle>+2/4> : Near, Lower, Down, Exit, Raise, 1/4, 1/4
<<Gate.closed, Controller.deactivate>+3/4> : Near, Lower, Down, Exit, 1/4, 1/4, 1/4
<<Gate.toOpen, Controller.idle>+3/4> : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4
<<Gate.closed, Controller.deactivate>+4/4> : Near, Lower, Down, Exit, 1/4, 1/4, 1/4, 1/4
<<Gate.toOpen, Controller.idle>+4/4> : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4

<<Gate.toOpen, Controller.idle>+5/4> : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, 1/4
<<Gate.toOpen, Controller.idle>+6/4> : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4
<<Gate.toOpen, Controller.idle>+7/4> : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4
<<Gate.toOpen, Controller.idle>+8/4> : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4

This is the end of State_Cover test cases.

Transition Coverage Cases

(<<Gate.opened, Controller.activate>+1/4> , <<Gate.toClose, Controller.monitor>+0/4>) Lower : Near, 1/4, Lower
(<<Gate.opened, Controller.activate>+2/4> , <<Gate.toClose, Controller.monitor>+0/4>) Lower : Near, 1/4, 1/4, Lower
(<<Gate.opened, Controller.activate>+3/4> , <<Gate.toClose, Controller.monitor>+0/4>) Lower : Near, 1/4, 1/4, 1/4, Lower
(<<Gate.opened, Controller.activate>+4/4> , <<Gate.toClose, Controller.monitor>+0/4>) Lower : Near, 1/4, 1/4, 1/4, 1/4, Lower
(<<Gate.closed, Controller.deactivate>+1/4> , <<Gate.toOpen, Controller.idle>+0/4>) Raise : Near, Lower, Down, Exit, 1/4, Raise
(<<Gate.closed, Controller.deactivate>+2/4> , <<Gate.toOpen, Controller.idle>+0/4>) Raise : Near, Lower, Down, Exit, 1/4, 1/4, Raise
(<<Gate.closed, Controller.deactivate>+3/4> , <<Gate.toOpen, Controller.idle>+0/4>) Raise : Near, Lower, Down, Exit, 1/4, 1/4, 1/4, Raise
(<<Gate.closed, Controller.deactivate>+4/4> , <<Gate.toOpen, Controller.idle>+0/4>) Raise : Near, Lower, Down, Exit, 1/4, 1/4, 1/4, 1/4, Raise
(<<Gate.toClose, Controller.monitor>+1/4> , <<Gate.closed, Controller.monitor>>) Down : Near, Lower, 1/4, Down
(<<Gate.toClose, Controller.monitor>+2/4> , <<Gate.closed, Controller.monitor>>) Down : Near, Lower, 1/4, 1/4, Down
(<<Gate.toClose, Controller.monitor>+3/4> , <<Gate.closed, Controller.monitor>>) Down : Near, Lower, 1/4, 1/4, 1/4, Down
(<<Gate.toClose, Controller.monitor>+4/4> , <<Gate.closed, Controller.monitor>>) Down : Near, Lower, 1/4, 1/4, 1/4, 1/4, Down
(<<Gate.toOpen, Controller.idle>+4/4> , <<Gate.opened, Controller.idle>>) Up : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, Up
(<<Gate.toOpen, Controller.idle>+5/4> , <<Gate.opened, Controller.idle>>) Up : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, 1/4, Up
(<<Gate.toOpen, Controller.idle>+6/4> , <<Gate.opened, Controller.idle>>) Up : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, Up
(<<Gate.toOpen, Controller.idle>+7/4> , <<Gate.opened, Controller.idle>>) Up : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, Up
(<<Gate.toOpen, Controller.idle>+8/4> , <<Gate.opened, Controller.idle>>) Up : Near, Lower, Down, Exit, Raise, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, 1/4, Up
(<<Gate.opened, Controller.activate>+0/4> , <<Gate.opened, Controller.activate>+0/4>) Near : Near, Near
(<<Gate.opened, Controller.activate>+1/4> , <<Gate.opened, Controller.activate>+1/4>) Near : Near, 1/4, Near
(<<Gate.opened, Controller.activate>+2/4> , <<Gate.opened, Controller.activate>+2/4>) Near : Near, 1/4, 1/4, Near
(<<Gate.opened, Controller.activate>+3/4> , <<Gate.opened, Controller.activate>+3/4>) Near : Near, 1/4, 1/4, 1/4, Near
(<<Gate.opened, Controller.activate>+4/4> , <<Gate.opened, Controller.activate>+4/4>) Near : Near, 1/4, 1/4, 1/4, 1/4, Near
(<<Gate.toClose, Controller.monitor>+0/4> , <<Gate.toClose, Controller.monitor>+0/4>) Exit : Near, Lower, Exit

(<<Gate.toClose, Controller.monitor>+1/4> , <<Gate.toClose, Controller.monitor>+1/4>) Exit : Near, Lower, 1/4, Exit
(<<Gate.toClose, Controller.monitor>+2/4> , <<Gate.toClose, Controller.monitor>+2/4>) Exit : Near, Lower, 1/4, 1/4, Exit
(<<Gate.toClose, Controller.monitor>+3/4> , <<Gate.toClose, Controller.monitor>+3/4>) Exit : Near, Lower, 1/4, 1/4, 1/4, Exit
(<<Gate.toClose, Controller.monitor>+4/4> , <<Gate.toClose, Controller.monitor>+4/4>) Exit : Near, Lower, 1/4, 1/4, 1/4, 1/4, Exit
(<<Gate.toClose, Controller.monitor>+0/4> , <<Gate.toClose, Controller.monitor>+0/4>) Near : Near, Lower, Near
(<<Gate.toClose, Controller.monitor>+1/4> , <<Gate.toClose, Controller.monitor>+1/4>) Near : Near, Lower, 1/4, Near
(<<Gate.toClose, Controller.monitor>+2/4> , <<Gate.toClose, Controller.monitor>+2/4>) Near : Near, Lower, 1/4, 1/4, Near
(<<Gate.toClose, Controller.monitor>+3/4> , <<Gate.toClose, Controller.monitor>+3/4>) Near : Near, Lower, 1/4, 1/4, 1/4, Near
(<<Gate.toClose, Controller.monitor>+4/4> , <<Gate.toClose, Controller.monitor>+4/4>) Near : Near, Lower, 1/4, 1/4, 1/4, 1/4, Near
(<<Gate.closed, Controller.monitor>> , <<Gate.closed, Controller.monitor>>) Exit : Near, Lower, Down, Exit
(<<Gate.closed, Controller.monitor>> , <<Gate.closed, Controller.monitor>>) Near : Near, Lower, Down, Near

This is the end of Transition_Cover test cases.

Appendix 2 Result of pair testing of Train and Controller

The class name is : Train_Controller

State List:

<<Train.idle, Controller.idle>, true>
<<Train.toCross, Controller.activate>, false>
<<Train.leave, Controller.monitor>, false>
<<Train.idle, Controller.deactivate>, false>
<<Train.toCross, Controller.monitor>, false>
<<Train.cross, Controller.monitor>, false>

Transition Spec List:

CR-0 <<Train.idle, Controller.idle>, <Train.toCross, Controller.activate>> , Near;
CR-1 <<Train.leave, Controller.monitor>, <Train.idle, Controller.deactivate>> , Exit;
CR-3 <<Train.toCross, Controller.activate>, <Train.toCross, Controller.monitor>> , Lower;
CR-4 <<Train.idle, Controller.deactivate>, <Train.idle, Controller.idle>> , Raise;
CR-5 <<Train.toCross, Controller.monitor>, <Train.cross, Controller.monitor>> , In;
CR-6 <<Train.cross, Controller.monitor>, <Train.leave, Controller.monitor>> , Out;
CR-7 <<Train.toCross, Controller.activate>, <Train.toCross, Controller.activate>> , Near;
CR-8 <<Train.leave, Controller.monitor>, <Train.leave, Controller.monitor>> , Exit;
CR-9 <<Train.leave, Controller.monitor>, <Train.leave, Controller.monitor>> , Near;
CR-10 <<Train.toCross, Controller.monitor>, <Train.toCross, Controller.monitor>> , Exit;
CR-11 <<Train.toCross, Controller.monitor>, <Train.toCross, Controller.monitor>> , Near;
CR-12 <<Train.cross, Controller.monitor>, <Train.cross, Controller.monitor>> , Exit;
CR-13 <<Train.cross, Controller.monitor>, <Train.cross, Controller.monitor>> , Near;

Transition Spec List(after time constraint decomposed):

R_1 <<Train.idle, Controller.idle>, <Train.toCross, Controller.activate>+0/4> , Near;
R_2 <<Train.toCross, Controller.activate>+0/4, <Train.toCross, Controller.activate>+1/4> , 1/4;
R_3 <<Train.toCross, Controller.activate>+1/4, <Train.toCross, Controller.activate>+2/4> , 1/4;
R_4 <<Train.toCross, Controller.activate>+2/4, <Train.toCross, Controller.activate>+3/4> , 1/4;
R_5 <<Train.toCross, Controller.activate>+3/4, <Train.toCross, Controller.activate>+4/4> , 1/4;
R_6 <<Train.leave, Controller.monitor>+0/4, <Train.idle, Controller.deactivate>+0/4> , Exit;

R_7 <<Train.leave, Controller.monitor>+1/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_8 <<Train.leave, Controller.monitor>+2/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_9 <<Train.leave, Controller.monitor>+3/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_10 <<Train.leave, Controller.monitor>+4/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_11 <<Train.leave, Controller.monitor>+5/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_12 <<Train.leave, Controller.monitor>+6/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_13 <<Train.leave, Controller.monitor>+7/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_14 <<Train.leave, Controller.monitor>+8/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_15 <<Train.leave, Controller.monitor>+9/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_16 <<Train.leave, Controller.monitor>+10/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_17 <<Train.leave, Controller.monitor>+11/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_18 <<Train.leave, Controller.monitor>+12/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_19 <<Train.leave, Controller.monitor>+13/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_20 <<Train.leave, Controller.monitor>+14/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_21 <<Train.leave, Controller.monitor>+15/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_22 <<Train.leave, Controller.monitor>+16/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_23 <<Train.leave, Controller.monitor>+17/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_24 <<Train.leave, Controller.monitor>+18/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_25 <<Train.leave, Controller.monitor>+19/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_26 <<Train.leave, Controller.monitor>+20/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_27 <<Train.leave, Controller.monitor>+21/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_28 <<Train.leave, Controller.monitor>+22/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_29 <<Train.leave, Controller.monitor>+23/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_30 <<Train.leave, Controller.monitor>+24/4, <Train.idle, Controller.deactivate>+0/4> , Exit;
R_31 <<Train.leave, Controller.monitor>+0/4, <Train.leave, Controller.monitor>+1/4> , 1/4;
R_32 <<Train.leave, Controller.monitor>+1/4, <Train.leave, Controller.monitor>+2/4> , 1/4;
R_33 <<Train.leave, Controller.monitor>+2/4, <Train.leave, Controller.monitor>+3/4> , 1/4;
R_34 <<Train.leave, Controller.monitor>+3/4, <Train.leave, Controller.monitor>+4/4> , 1/4;
R_35 <<Train.leave, Controller.monitor>+4/4, <Train.leave, Controller.monitor>+5/4> , 1/4;
R_36 <<Train.leave, Controller.monitor>+5/4, <Train.leave, Controller.monitor>+6/4> , 1/4;
R_37 <<Train.leave, Controller.monitor>+6/4, <Train.leave, Controller.monitor>+7/4> , 1/4;
R_38 <<Train.leave, Controller.monitor>+7/4, <Train.leave, Controller.monitor>+8/4> , 1/4;
R_39 <<Train.leave, Controller.monitor>+8/4, <Train.leave, Controller.monitor>+9/4> , 1/4;
R_40 <<Train.leave, Controller.monitor>+9/4, <Train.leave, Controller.monitor>+10/4> , 1/4;
R_41 <<Train.leave, Controller.monitor>+10/4, <Train.leave, Controller.monitor>+11/4> , 1/4;
R_42 <<Train.leave, Controller.monitor>+11/4, <Train.leave, Controller.monitor>+12/4> , 1/4;
R_43 <<Train.leave, Controller.monitor>+12/4, <Train.leave, Controller.monitor>+13/4> , 1/4;
R_44 <<Train.leave, Controller.monitor>+13/4, <Train.leave, Controller.monitor>+14/4> , 1/4;
R_45 <<Train.leave, Controller.monitor>+14/4, <Train.leave, Controller.monitor>+15/4> , 1/4;
R_46 <<Train.leave, Controller.monitor>+15/4, <Train.leave, Controller.monitor>+16/4> , 1/4;
R_47 <<Train.leave, Controller.monitor>+16/4, <Train.leave, Controller.monitor>+17/4> , 1/4;
R_48 <<Train.leave, Controller.monitor>+17/4, <Train.leave, Controller.monitor>+18/4> , 1/4;
R_49 <<Train.leave, Controller.monitor>+18/4, <Train.leave, Controller.monitor>+19/4> , 1/4;
R_50 <<Train.leave, Controller.monitor>+19/4, <Train.leave, Controller.monitor>+20/4> , 1/4;
R_51 <<Train.leave, Controller.monitor>+20/4, <Train.leave, Controller.monitor>+21/4> , 1/4;
R_52 <<Train.leave, Controller.monitor>+21/4, <Train.leave, Controller.monitor>+22/4> , 1/4;
R_53 <<Train.leave, Controller.monitor>+22/4, <Train.leave, Controller.monitor>+23/4> , 1/4;
R_54 <<Train.leave, Controller.monitor>+23/4, <Train.leave, Controller.monitor>+24/4> , 1/4;
R_55 <<Train.toCross, Controller.activate>+0/4, <Train.toCross, Controller.monitor>+0/4> , Lower;
R_56 <<Train.toCross, Controller.activate>+1/4, <Train.toCross, Controller.monitor>+0/4> , Lower;
R_57 <<Train.toCross, Controller.activate>+2/4, <Train.toCross, Controller.monitor>+0/4> , Lower;
R_58 <<Train.toCross, Controller.activate>+3/4, <Train.toCross, Controller.monitor>+0/4> , Lower;
R_59 <<Train.toCross, Controller.activate>+4/4, <Train.toCross, Controller.monitor>+0/4> , Lower;
R_60 <<Train.idle, Controller.deactivate>+0/4, <Train.idle, Controller.idle>> , Raise;
R_61 <<Train.idle, Controller.deactivate>+1/4, <Train.idle, Controller.idle>> , Raise;
R_62 <<Train.idle, Controller.deactivate>+2/4, <Train.idle, Controller.idle>> , Raise;

R_63 <<Train.idle, Controller.deactivate>+3/4, <Train.idle, Controller.idle>> , Raise;
R_64 <<Train.idle, Controller.deactivate>+4/4, <Train.idle, Controller.idle>> , Raise;
R_65 <<Train.idle, Controller.deactivate>+0/4, <Train.idle, Controller.deactivate>+1/4> , 1/4;
R_66 <<Train.idle, Controller.deactivate>+1/4, <Train.idle, Controller.deactivate>+2/4> , 1/4;
R_67 <<Train.idle, Controller.deactivate>+2/4, <Train.idle, Controller.deactivate>+3/4> , 1/4;
R_68 <<Train.idle, Controller.deactivate>+3/4, <Train.idle, Controller.deactivate>+4/4> , 1/4;
R_69 <<Train.toCross, Controller.monitor>+8/4, <Train.cross, Controller.monitor>> , In;
R_70 <<Train.toCross, Controller.monitor>+9/4, <Train.cross, Controller.monitor>> , In;
R_71 <<Train.toCross, Controller.monitor>+10/4, <Train.cross, Controller.monitor>> , In;
R_72 <<Train.toCross, Controller.monitor>+11/4, <Train.cross, Controller.monitor>> , In;
R_73 <<Train.toCross, Controller.monitor>+12/4, <Train.cross, Controller.monitor>> , In;
R_74 <<Train.toCross, Controller.monitor>+13/4, <Train.cross, Controller.monitor>> , In;
R_75 <<Train.toCross, Controller.monitor>+14/4, <Train.cross, Controller.monitor>> , In;
R_76 <<Train.toCross, Controller.monitor>+15/4, <Train.cross, Controller.monitor>> , In;
R_77 <<Train.toCross, Controller.monitor>+16/4, <Train.cross, Controller.monitor>> , In;
R_78 <<Train.toCross, Controller.monitor>+0/4, <Train.toCross, Controller.monitor>+1/4> , 1/4;
R_79 <<Train.toCross, Controller.monitor>+1/4, <Train.toCross, Controller.monitor>+2/4> , 1/4;
R_80 <<Train.toCross, Controller.monitor>+2/4, <Train.toCross, Controller.monitor>+3/4> , 1/4;
R_81 <<Train.toCross, Controller.monitor>+3/4, <Train.toCross, Controller.monitor>+4/4> , 1/4;
R_82 <<Train.toCross, Controller.monitor>+4/4, <Train.toCross, Controller.monitor>+5/4> , 1/4;
R_83 <<Train.toCross, Controller.monitor>+5/4, <Train.toCross, Controller.monitor>+6/4> , 1/4;
R_84 <<Train.toCross, Controller.monitor>+6/4, <Train.toCross, Controller.monitor>+7/4> , 1/4;
R_85 <<Train.toCross, Controller.monitor>+7/4, <Train.toCross, Controller.monitor>+8/4> , 1/4;
R_86 <<Train.toCross, Controller.monitor>+8/4, <Train.toCross, Controller.monitor>+9/4> , 1/4;
R_87 <<Train.toCross, Controller.monitor>+9/4, <Train.toCross, Controller.monitor>+10/4> , 1/4;
R_88 <<Train.toCross, Controller.monitor>+10/4, <Train.toCross, Controller.monitor>+11/4> , 1/4;
R_89 <<Train.toCross, Controller.monitor>+11/4, <Train.toCross, Controller.monitor>+12/4> , 1/4;
R_90 <<Train.toCross, Controller.monitor>+12/4, <Train.toCross, Controller.monitor>+13/4> , 1/4;
R_91 <<Train.toCross, Controller.monitor>+13/4, <Train.toCross, Controller.monitor>+14/4> , 1/4;
R_92 <<Train.toCross, Controller.monitor>+14/4, <Train.toCross, Controller.monitor>+15/4> , 1/4;
R_93 <<Train.toCross, Controller.monitor>+15/4, <Train.toCross, Controller.monitor>+16/4> , 1/4;
R_94 <<Train.cross, Controller.monitor>, <Train.leave, Controller.monitor>+0/4> , Out;
R_95 <<Train.toCross, Controller.activate>+0/4, <Train.toCross, Controller.activate>+0/4> , Near;
R_96 <<Train.toCross, Controller.activate>+1/4, <Train.toCross, Controller.activate>+1/4> , Near;
R_97 <<Train.toCross, Controller.activate>+2/4, <Train.toCross, Controller.activate>+2/4> , Near;
R_98 <<Train.toCross, Controller.activate>+3/4, <Train.toCross, Controller.activate>+3/4> , Near;
R_99 <<Train.toCross, Controller.activate>+4/4, <Train.toCross, Controller.activate>+4/4> , Near;
R_100 <<Train.leave, Controller.monitor>+0/4, <Train.leave, Controller.monitor>+0/4> , Exit;
R_101 <<Train.leave, Controller.monitor>+1/4, <Train.leave, Controller.monitor>+1/4> , Exit;
R_102 <<Train.leave, Controller.monitor>+2/4, <Train.leave, Controller.monitor>+2/4> , Exit;
R_103 <<Train.leave, Controller.monitor>+3/4, <Train.leave, Controller.monitor>+3/4> , Exit;
R_104 <<Train.leave, Controller.monitor>+4/4, <Train.leave, Controller.monitor>+4/4> , Exit;
R_105 <<Train.leave, Controller.monitor>+5/4, <Train.leave, Controller.monitor>+5/4> , Exit;
R_106 <<Train.leave, Controller.monitor>+6/4, <Train.leave, Controller.monitor>+6/4> , Exit;
R_107 <<Train.leave, Controller.monitor>+7/4, <Train.leave, Controller.monitor>+7/4> , Exit;
R_108 <<Train.leave, Controller.monitor>+8/4, <Train.leave, Controller.monitor>+8/4> , Exit;
R_109 <<Train.leave, Controller.monitor>+9/4, <Train.leave, Controller.monitor>+9/4> , Exit;
R_110 <<Train.leave, Controller.monitor>+10/4, <Train.leave, Controller.monitor>+10/4> , Exit;
R_111 <<Train.leave, Controller.monitor>+11/4, <Train.leave, Controller.monitor>+11/4> , Exit;
R_112 <<Train.leave, Controller.monitor>+12/4, <Train.leave, Controller.monitor>+12/4> , Exit;
R_113 <<Train.leave, Controller.monitor>+13/4, <Train.leave, Controller.monitor>+13/4> , Exit;
R_114 <<Train.leave, Controller.monitor>+14/4, <Train.leave, Controller.monitor>+14/4> , Exit;
R_115 <<Train.leave, Controller.monitor>+15/4, <Train.leave, Controller.monitor>+15/4> , Exit;
R_116 <<Train.leave, Controller.monitor>+16/4, <Train.leave, Controller.monitor>+16/4> , Exit;
R_117 <<Train.leave, Controller.monitor>+17/4, <Train.leave, Controller.monitor>+17/4> , Exit;
R_118 <<Train.leave, Controller.monitor>+18/4, <Train.leave, Controller.monitor>+18/4> , Exit;

R_119 <<Train.leave, Controller.monitor>+19/4, <Train.leave, Controller.monitor>+19/4> , Exit;
R_120 <<Train.leave, Controller.monitor>+20/4, <Train.leave, Controller.monitor>+20/4> , Exit;
R_121 <<Train.leave, Controller.monitor>+21/4, <Train.leave, Controller.monitor>+21/4> , Exit;
R_122 <<Train.leave, Controller.monitor>+22/4, <Train.leave, Controller.monitor>+22/4> , Exit;
R_123 <<Train.leave, Controller.monitor>+23/4, <Train.leave, Controller.monitor>+23/4> , Exit;
R_124 <<Train.leave, Controller.monitor>+24/4, <Train.leave, Controller.monitor>+24/4> , Exit;
R_125 <<Train.leave, Controller.monitor>+0/4, <Train.leave, Controller.monitor>+0/4> , Near;
R_126 <<Train.leave, Controller.monitor>+1/4, <Train.leave, Controller.monitor>+1/4> , Near;
R_127 <<Train.leave, Controller.monitor>+2/4, <Train.leave, Controller.monitor>+2/4> , Near;
R_128 <<Train.leave, Controller.monitor>+3/4, <Train.leave, Controller.monitor>+3/4> , Near;
R_129 <<Train.leave, Controller.monitor>+4/4, <Train.leave, Controller.monitor>+4/4> , Near;
R_130 <<Train.leave, Controller.monitor>+5/4, <Train.leave, Controller.monitor>+5/4> , Near;
R_131 <<Train.leave, Controller.monitor>+6/4, <Train.leave, Controller.monitor>+6/4> , Near;
R_132 <<Train.leave, Controller.monitor>+7/4, <Train.leave, Controller.monitor>+7/4> , Near;
R_133 <<Train.leave, Controller.monitor>+8/4, <Train.leave, Controller.monitor>+8/4> , Near;
R_134 <<Train.leave, Controller.monitor>+9/4, <Train.leave, Controller.monitor>+9/4> , Near;
R_135 <<Train.leave, Controller.monitor>+10/4, <Train.leave, Controller.monitor>+10/4> , Near;
R_136 <<Train.leave, Controller.monitor>+11/4, <Train.leave, Controller.monitor>+11/4> , Near;
R_137 <<Train.leave, Controller.monitor>+12/4, <Train.leave, Controller.monitor>+12/4> , Near;
R_138 <<Train.leave, Controller.monitor>+13/4, <Train.leave, Controller.monitor>+13/4> , Near;
R_139 <<Train.leave, Controller.monitor>+14/4, <Train.leave, Controller.monitor>+14/4> , Near;
R_140 <<Train.leave, Controller.monitor>+15/4, <Train.leave, Controller.monitor>+15/4> , Near;
R_141 <<Train.leave, Controller.monitor>+16/4, <Train.leave, Controller.monitor>+16/4> , Near;
R_142 <<Train.leave, Controller.monitor>+17/4, <Train.leave, Controller.monitor>+17/4> , Near;
R_143 <<Train.leave, Controller.monitor>+18/4, <Train.leave, Controller.monitor>+18/4> , Near;
R_144 <<Train.leave, Controller.monitor>+19/4, <Train.leave, Controller.monitor>+19/4> , Near;
R_145 <<Train.leave, Controller.monitor>+20/4, <Train.leave, Controller.monitor>+20/4> , Near;
R_146 <<Train.leave, Controller.monitor>+21/4, <Train.leave, Controller.monitor>+21/4> , Near;
R_147 <<Train.leave, Controller.monitor>+22/4, <Train.leave, Controller.monitor>+22/4> , Near;
R_148 <<Train.leave, Controller.monitor>+23/4, <Train.leave, Controller.monitor>+23/4> , Near;
R_149 <<Train.leave, Controller.monitor>+24/4, <Train.leave, Controller.monitor>+24/4> , Near;
R_150 <<Train.toCross, Controller.monitor>+8/4, <Train.toCross, Controller.monitor>+8/4> , Exit;
R_151 <<Train.toCross, Controller.monitor>+9/4, <Train.toCross, Controller.monitor>+9/4> , Exit;
R_152 <<Train.toCross, Controller.monitor>+10/4, <Train.toCross, Controller.monitor>+10/4> , Exit;
R_153 <<Train.toCross, Controller.monitor>+11/4, <Train.toCross, Controller.monitor>+11/4> , Exit;
R_154 <<Train.toCross, Controller.monitor>+12/4, <Train.toCross, Controller.monitor>+12/4> , Exit;
R_155 <<Train.toCross, Controller.monitor>+13/4, <Train.toCross, Controller.monitor>+13/4> , Exit;
R_156 <<Train.toCross, Controller.monitor>+14/4, <Train.toCross, Controller.monitor>+14/4> , Exit;
R_157 <<Train.toCross, Controller.monitor>+15/4, <Train.toCross, Controller.monitor>+15/4> , Exit;
R_158 <<Train.toCross, Controller.monitor>+16/4, <Train.toCross, Controller.monitor>+16/4> , Exit;
R_159 <<Train.toCross, Controller.monitor>+8/4, <Train.toCross, Controller.monitor>+8/4> , Near;
R_160 <<Train.toCross, Controller.monitor>+9/4, <Train.toCross, Controller.monitor>+9/4> , Near;
R_161 <<Train.toCross, Controller.monitor>+10/4, <Train.toCross, Controller.monitor>+10/4> , Near;
R_162 <<Train.toCross, Controller.monitor>+11/4, <Train.toCross, Controller.monitor>+11/4> , Near;
R_163 <<Train.toCross, Controller.monitor>+12/4, <Train.toCross, Controller.monitor>+12/4> , Near;
R_164 <<Train.toCross, Controller.monitor>+13/4, <Train.toCross, Controller.monitor>+13/4> , Near;
R_165 <<Train.toCross, Controller.monitor>+14/4, <Train.toCross, Controller.monitor>+14/4> , Near;
R_166 <<Train.toCross, Controller.monitor>+15/4, <Train.toCross, Controller.monitor>+15/4> , Near;
R_167 <<Train.toCross, Controller.monitor>+16/4, <Train.toCross, Controller.monitor>+16/4> , Near;
R_168 <<Train.cross, Controller.monitor>, <Train.cross, Controller.monitor>> , Exit;
R_169 <<Train.cross, Controller.monitor>, <Train.cross, Controller.monitor>> , Near;

Appendix 3 Result of System Testing of train⊗ (gate ⊗ controller)

The class name is: Train_Gate_Controller

State List:

<<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>, true>
<<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>, false>
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>, false>
<<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>, false>
<<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>, false>
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>, false>
<<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>, false>
<<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>, false>

Transition Spec List:

CR-0 <<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>> , Near;
CR-1 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>> , Exit;
CR-3 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>> , Lower;
CR-4 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>> , Raise;
CR-6 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>> , Down;
CR-7 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>, <Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> , Up;
CR-8 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
CR-9 <<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>> , Out;
CR-10 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>> , Near;
CR-11 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>> , Exit;
CR-12 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>> , Near;
CR-13 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>> , Exit;
CR-14 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>> , Near;
CR-15 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>> , Exit;
CR-16 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>> , Near;
CR-17 <<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , Exit;
CR-18 <<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , Near;

Transition Spec List:

R_1 <<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+0/6> , Near;
R_2 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+0/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+1/6> , 1/6;

R_3 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+1/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+2/6> , 1/6;
R_4 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+2/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+3/6> , 1/6;
R_5 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+3/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+4/6> , 1/6;
R_6 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+4/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+5/6> , 1/6;
R_7 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+5/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+6/6> , 1/6;
R_8 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_9 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_10 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_11 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_12 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_13 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_14 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_15 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_16 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_17 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_18 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_19 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_20 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_21 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_22 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_23 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_24 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_25 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_26 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_27 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_28 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_29 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_30 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;

R_31 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_32 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_33 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+25/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_34 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+26/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_35 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+27/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_36 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+28/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_37 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+29/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_38 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+30/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_39 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+31/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_40 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+32/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_41 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+33/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_42 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+34/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_43 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+35/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_44 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+36/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> , Exit;
R_45 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6> , 1/6;
R_46 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6> , 1/6;
R_47 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6> , 1/6;
R_48 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6> , 1/6;
R_49 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6> , 1/6;
R_50 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6> , 1/6;
R_51 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6> , 1/6;
R_52 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6> , 1/6;
R_53 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6> , 1/6;
R_54 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6> , 1/6;
R_55 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6> , 1/6;
R_56 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> , 1/6;
R_57 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> , 1/6;
R_58 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> , 1/6;

R_59 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> , 1/6;
R_60 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> , 1/6;
R_61 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> , 1/6;
R_62 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> , 1/6;
R_63 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> , 1/6;
R_64 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> , 1/6;
R_65 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> , 1/6;
R_66 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> , 1/6;
R_67 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> , 1/6;
R_68 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> , 1/6;
R_69 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+25/6> , 1/6;
R_70 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+25/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+26/6> , 1/6;
R_71 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+26/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+27/6> , 1/6;
R_72 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+27/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+28/6> , 1/6;
R_73 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+28/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+29/6> , 1/6;
R_74 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+29/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+30/6> , 1/6;
R_75 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+30/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+31/6> , 1/6;
R_76 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+31/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+32/6> , 1/6;
R_77 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+32/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+33/6> , 1/6;
R_78 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+33/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+34/6> , 1/6;
R_79 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+34/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+35/6> , 1/6;
R_80 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+35/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+36/6> , 1/6;
R_81 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+0/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Lower;
R_82 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+1/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Lower;
R_83 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+2/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Lower;
R_84 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+3/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Lower;
R_85 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+4/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Lower;
R_86 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+5/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Lower;

R_87 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+6/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Lower;
R_88 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> , Raise;
R_89 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+1/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> , Raise;
R_90 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+2/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> , Raise;
R_91 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+3/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> , Raise;
R_92 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+4/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> , Raise;
R_93 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+5/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> , Raise;
R_94 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+6/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> , Raise;
R_95 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+1/6> , 1/6;
R_96 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+1/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+2/6> , 1/6;
R_97 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+2/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+3/6> , 1/6;
R_98 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+3/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+4/6> , 1/6;
R_99 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+4/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+5/6> , 1/6;
R_100 <<Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+5/6, <Train.idle, Gate_Controller.<Gate.closed, Controller.deactivate>>+6/6> , 1/6;
R_101 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Down;
R_102 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Down;
R_103 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Down;
R_104 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Down;
R_105 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Down;
R_106 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Down;
R_107 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+6/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Down;
R_108 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6> , 1/6;
R_109 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6> , 1/6;
R_110 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6> , 1/6;
R_111 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6> , 1/6;
R_112 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6> , 1/6;
R_113 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+6/6> , 1/6;
R_114 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+6/6, <Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> , Up;

R_115 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+7/6, <Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> , Up;
R_116 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+8/6, <Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> , Up;
R_117 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+9/6, <Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> , Up;
R_118 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+10/6, <Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> , Up;
R_119 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+11/6, <Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> , Up;
R_120 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+12/6, <Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> , Up;
R_121 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+1/6> , 1/6;
R_122 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+1/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+2/6> , 1/6;
R_123 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+2/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+3/6> , 1/6;
R_124 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+3/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+4/6> , 1/6;
R_125 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+4/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+5/6> , 1/6;
R_126 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+5/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+6/6> , 1/6;
R_127 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+6/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+7/6> , 1/6;
R_128 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+7/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+8/6> , 1/6;
R_129 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+8/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+9/6> , 1/6;
R_130 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+9/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+10/6> , 1/6;
R_131 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+10/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+11/6> , 1/6;
R_132 <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+11/6, <Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+12/6> , 1/6;
R_133 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_134 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_135 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_136 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_137 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_138 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_139 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_140 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_141 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_142 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;

R_143 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_144 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_145 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , In;
R_146 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6> , 1/6;
R_147 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6> , 1/6;
R_148 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6> , 1/6;
R_149 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6> , 1/6;
R_150 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6> , 1/6;
R_151 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6> , 1/6;
R_152 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6> , 1/6;
R_153 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6> , 1/6;
R_154 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6> , 1/6;
R_155 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6> , 1/6;
R_156 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6> , 1/6;
R_157 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> , 1/6;
R_158 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> , 1/6;
R_159 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> , 1/6;
R_160 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> , 1/6;
R_161 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> , 1/6;
R_162 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> , 1/6;
R_163 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> , 1/6;
R_164 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> , 1/6;
R_165 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> , 1/6;
R_166 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> , 1/6;
R_167 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> , 1/6;
R_168 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> , 1/6;
R_169 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> , 1/6;
R_170 <<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Out;

R_171 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+0/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+0/6> , Near;
R_172 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+1/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+1/6> , Near;
R_173 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+2/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+2/6> , Near;
R_174 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+3/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+3/6> , Near;
R_175 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+4/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+4/6> , Near;
R_176 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+5/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+5/6> , Near;
R_177 <<Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+6/6, <Train.toCross, Gate_Controller.<Gate.opened, Controller.activate>>+6/6> , Near;
R_178 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Exit;
R_179 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6> , Exit;
R_180 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6> , Exit;
R_181 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6> , Exit;
R_182 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6> , Exit;
R_183 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6> , Exit;
R_184 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6> , Exit;
R_185 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6> , Exit;
R_186 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6> , Exit;
R_187 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6> , Exit;
R_188 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6> , Exit;
R_189 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6> , Exit;
R_190 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> , Exit;
R_191 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> , Exit;
R_192 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> , Exit;
R_193 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> , Exit;
R_194 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> , Exit;
R_195 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> , Exit;
R_196 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> , Exit;
R_197 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> , Exit;
R_198 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> , Exit;

R_199 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> , Exit;
R_200 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> , Exit;
R_201 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> , Exit;
R_202 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> , Exit;
R_203 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+25/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+25/6> , Exit;
R_204 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+26/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+26/6> , Exit;
R_205 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+27/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+27/6> , Exit;
R_206 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+28/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+28/6> , Exit;
R_207 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+29/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+29/6> , Exit;
R_208 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+30/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+30/6> , Exit;
R_209 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+31/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+31/6> , Exit;
R_210 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+32/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+32/6> , Exit;
R_211 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+33/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+33/6> , Exit;
R_212 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+34/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+34/6> , Exit;
R_213 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+35/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+35/6> , Exit;
R_214 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+36/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+36/6> , Exit;
R_215 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> , Near;
R_216 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6> , Near;
R_217 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6> , Near;
R_218 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6> , Near;
R_219 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6> , Near;
R_220 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6> , Near;
R_221 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6> , Near;
R_222 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6> , Near;
R_223 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6> , Near;
R_224 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6> , Near;
R_225 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6> , Near;
R_226 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6> , Near;

R_227 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> , Near;
R_228 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> , Near;
R_229 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> , Near;
R_230 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> , Near;
R_231 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> , Near;
R_232 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> , Near;
R_233 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> , Near;
R_234 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> , Near;
R_235 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> , Near;
R_236 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> , Near;
R_237 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> , Near;
R_238 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> , Near;
R_239 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> , Near;
R_240 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+25/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+25/6> , Near;
R_241 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+26/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+26/6> , Near;
R_242 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+27/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+27/6> , Near;
R_243 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+28/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+28/6> , Near;
R_244 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+29/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+29/6> , Near;
R_245 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+30/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+30/6> , Near;
R_246 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+31/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+31/6> , Near;
R_247 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+32/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+32/6> , Near;
R_248 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+33/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+33/6> , Near;
R_249 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+34/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+34/6> , Near;
R_250 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+35/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+35/6> , Near;
R_251 <<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+36/6, <Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+36/6> , Near;
R_252 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Exit;
R_253 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6> , Exit;
R_254 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6> , Exit;

R_255 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6> , Exit;
R_256 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6> , Exit;
R_257 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6> , Exit;
R_258 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+6/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+6/6> , Exit;
R_259 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> , Near;
R_260 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6> , Near;
R_261 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6> , Near;
R_262 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6> , Near;
R_263 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6> , Near;
R_264 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6> , Near;
R_265 <<Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+6/6, <Train.toCross, Gate_Controller.<Gate.toClose, Controller.monitor>>+6/6> , Near;
R_266 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> , Exit;
R_267 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> , Exit;
R_268 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> , Exit;
R_269 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> , Exit;
R_270 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> , Exit;
R_271 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> , Exit;
R_272 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> , Exit;
R_273 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> , Exit;
R_274 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> , Exit;
R_275 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> , Exit;
R_276 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> , Exit;
R_277 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> , Exit;
R_278 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> , Exit;
R_279 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> , Near;
R_280 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> , Near;
R_281 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> , Near;
R_282 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> , Near;

R_283 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> , Near;
R_284 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> , Near;
R_285 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> , Near;
R_286 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> , Near;
R_287 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> , Near;
R_288 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> , Near;
R_289 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> , Near;
R_290 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> , Near;
R_291 <<Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6, <Train.toCross, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> , Near;
R_292 <<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , Exit;
R_293 <<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>, <Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> , Near;

```
Test Cases:

Class Train_Gate_Controller

State Coverage Cases
<<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+0/6> : Near
<<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+1/6> : Near, 1/6
<<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+0/6> : Near, Lower
<<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+2/6> : Near, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+0/6> : Near, Lower, Down
<<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+1/6> : Near, Lower, 1/6
<<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+3/6> : Near, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+1/6> : Near, Lower, Down, 1/6
<<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+2/6> : Near, Lower, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+4/6> : Near, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+2/6> : Near, Lower, Down, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+3/6> : Near, Lower, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+5/6> : Near, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+3/6> : Near, Lower, Down, 1/6, 1/6, 1/6
```

```
<<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+4/6> : Near, Lower, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+6/6> : Near, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+4/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+5/6> : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+5/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+6/6> : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+6/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+7/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+8/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+9/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+10/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+11/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+12/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+13/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+0/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+14/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+1/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+15/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise
```

```
<<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+1/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+2/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+16/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+1/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6
<<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+2/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+3/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+17/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+2/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+3/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+4/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+18/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+3/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+4/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6,
1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+5/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+19/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+4/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+5/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6,
1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+6/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
```

<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+20/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+5/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+6/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+7/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+21/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+6/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+8/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+22/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+7/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+9/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+23/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+8/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+10/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+24/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+9/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+11/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6

<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+10/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+12/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+11/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+13/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6
<<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+12/6> :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+14/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+15/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+16/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+17/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+18/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+19/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+20/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+21/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+22/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+23/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+24/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+25/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+26/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+27/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+28/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+29/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+30/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+31/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+32/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+33/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+34/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,

```
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+35/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Train.leave, Gate_Controller.<Gate.closed, Controller.monitor>>+36/6>
: Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

This is the end of State_Cover test cases.

Transition Coverage Cases

( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+1/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+2/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6,
Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+3/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6,
1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+4/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6,
1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+5/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6,
1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+6/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+7/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+8/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
```

```
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+9/6> , <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+0/6> ) Exit :  Near, Lower, Down, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+10/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+11/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+12/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+13/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+14/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+15/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+16/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+17/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+18/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
```

```
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+19/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+20/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+21/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+22/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+23/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+24/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+25/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+26/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+27/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
```

```
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+28/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+29/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+30/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+31/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+32/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+33/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+34/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+35/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
```

```
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+36/6> , <<Train.idle,
Gate_Controller.<Gate.closed, Controller.deactivate>>+0/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+1/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> ) Lower :
Near, 1/6, Lower
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+2/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> ) Lower :
Near, 1/6, 1/6, Lower
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+3/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> ) Lower :
Near, 1/6, 1/6, 1/6, Lower
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+4/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> ) Lower :
Near, 1/6, 1/6, 1/6, 1/6, Lower
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+5/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> ) Lower :
Near, 1/6, 1/6, 1/6, 1/6, 1/6, Lower
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+6/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> ) Lower :
Near, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Lower
( <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+1/6> , <<Train.idle,
Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> ) Raise :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, Exit, 1/6, Raise
( <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+2/6> , <<Train.idle,
Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> ) Raise :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, Exit, 1/6, 1/6, Raise
( <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+3/6> , <<Train.idle,
Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> ) Raise :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, Exit, 1/6, 1/6, 1/6, Raise
( <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+4/6> , <<Train.idle,
Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> ) Raise :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6, Raise
( <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+5/6> , <<Train.idle,
Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> ) Raise :  Near,
```

```
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6, 1/6, Raise
( <<Train.idle, Gate_Controller.<Gate.closed,
Controller.deactivate>>+6/6> , <<Train.idle,
Gate_Controller.<Gate.toOpen, Controller.idle>>+0/6> ) Raise :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Raise
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+1/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> ) Down :  Near,
Lower, 1/6, Down
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+2/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> ) Down :  Near,
Lower, 1/6, 1/6, Down
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+3/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> ) Down :  Near,
Lower, 1/6, 1/6, 1/6, Down
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+4/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> ) Down :  Near,
Lower, 1/6, 1/6, 1/6, 1/6, Down
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+5/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> ) Down :  Near,
Lower, 1/6, 1/6, 1/6, 1/6, 1/6, Down
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+6/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> ) Down :  Near,
Lower, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Down
( <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+6/6> ,
<<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> ) Up :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Up
( <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+7/6> ,
<<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> ) Up :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Up
( <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+8/6> ,
<<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> ) Up :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Up
( <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+9/6> ,
<<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> ) Up :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, Up
( <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+10/6> ,
<<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> ) Up :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Up
( <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+11/6> ,
<<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> ) Up :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
```

```
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Up
( <<Train.idle, Gate_Controller.<Gate.toOpen, Controller.idle>>+12/6> ,
<<Train.idle, Gate_Controller.<Gate.opened, Controller.idle>>> ) Up :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Up
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+13/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+14/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+15/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+16/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+17/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+18/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+19/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+20/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+21/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+22/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
```

```
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+23/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+24/6> , <<Train.cross,
Gate_Controller.<Gate.closed, Controller.monitor>>> ) In :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+0/6> , <<Train.toCross,
Gate_Controller.<Gate.opened, Controller.activate>>+0/6> ) Near :
Near, Near
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+1/6> , <<Train.toCross,
Gate_Controller.<Gate.opened, Controller.activate>>+1/6> ) Near :
Near, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+2/6> , <<Train.toCross,
Gate_Controller.<Gate.opened, Controller.activate>>+2/6> ) Near :
Near, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+3/6> , <<Train.toCross,
Gate_Controller.<Gate.opened, Controller.activate>>+3/6> ) Near :
Near, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+4/6> , <<Train.toCross,
Gate_Controller.<Gate.opened, Controller.activate>>+4/6> ) Near :
Near, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+5/6> , <<Train.toCross,
Gate_Controller.<Gate.opened, Controller.activate>>+5/6> ) Near :
Near, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.opened,
Controller.activate>>+6/6> , <<Train.toCross,
Gate_Controller.<Gate.opened, Controller.activate>>+6/6> ) Near :
Near, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+0/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+1/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+1/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+2/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+2/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+3/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+3/6> ) Exit :  Near,
```

```
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+4/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+4/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+5/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+5/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+6/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+6/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+7/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+7/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+8/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+8/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+9/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+9/6> ) Exit :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+10/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+10/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+11/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+11/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+12/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+13/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Exit
```

```
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+14/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+15/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+16/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+17/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+18/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+19/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+20/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+21/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+22/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+23/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> ) Exit :
```

```
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+24/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+25/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+25/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+26/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+26/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+27/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+27/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+28/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+28/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+29/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+29/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+30/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+30/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+31/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+31/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
```

```
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+32/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+32/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+33/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+33/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+34/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+34/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+35/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+35/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+36/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+36/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+0/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+0/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+1/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+1/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+2/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+2/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+3/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+3/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, Near
```

```
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+4/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+4/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+5/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+5/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+6/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+6/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+7/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+7/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+8/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+8/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+9/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+9/6> ) Near :  Near,
Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+10/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+10/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+11/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+11/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+12/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+13/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+14/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> ) Near :
```

```
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+15/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+16/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+17/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+18/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+19/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+20/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+21/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+22/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+23/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
```

```
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+24/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+25/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+25/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+26/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+26/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+27/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+27/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+28/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+28/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+29/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+29/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+30/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+30/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+31/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+31/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
```

```
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+32/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+32/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+33/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+33/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+34/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+34/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+35/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+35/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.leave, Gate_Controller.<Gate.closed,
Controller.monitor>>+36/6> , <<Train.leave,
Gate_Controller.<Gate.closed, Controller.monitor>>+36/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+0/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> ) Exit :
Near, Lower, Exit
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+1/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6> ) Exit :
Near, Lower, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+2/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6> ) Exit :
Near, Lower, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+3/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6> ) Exit :
Near, Lower, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+4/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6> ) Exit :
Near, Lower, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+5/6> , <<Train.toCross,
```

```
Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6> ) Exit :
Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+6/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+6/6> ) Exit :
Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+0/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+0/6> ) Near :
Near, Lower, Near
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+1/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+1/6> ) Near :
Near, Lower, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+2/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+2/6> ) Near :
Near, Lower, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+3/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+3/6> ) Near :
Near, Lower, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+4/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+4/6> ) Near :
Near, Lower, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+5/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+5/6> ) Near :
Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.toClose,
Controller.monitor>>+6/6> , <<Train.toCross,
Gate_Controller.<Gate.toClose, Controller.monitor>>+6/6> ) Near :
Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+12/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+13/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+14/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+15/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+16/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> ) Exit :
```

```
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+17/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+18/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+19/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+20/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+21/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+22/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+23/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+24/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> ) Exit :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Exit
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+12/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+12/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+13/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+13/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+14/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+14/6> ) Near :
```

```
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+15/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+15/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+16/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+16/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+17/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+17/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+18/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+18/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+19/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+19/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+20/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+20/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+21/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+21/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+22/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+22/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+23/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+23/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
( <<Train.toCross, Gate_Controller.<Gate.closed,
Controller.monitor>>+24/6> , <<Train.toCross,
Gate_Controller.<Gate.closed, Controller.monitor>>+24/6> ) Near :
Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
Near
( <<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> ,
<<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> )
Exit :  Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, In, Exit
```

```
( <<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> ,
<<Train.cross, Gate_Controller.<Gate.closed, Controller.monitor>>> )
Near :  Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6,
1/6, 1/6, 1/6, In, Near

This is the end of Transition_Cover test cases.
```

Appendix 4 Result of System Testing of gate ⊗(train ⊗ controller)

The class name is : Gate_Train_Controller

State List:

<<Gate.opened, Train_Controller.<Train.idle, Controller.idle>>, true>
<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>, false>
<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>, false>
<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>, false>
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>, false>
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>, false>
<<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>, false>
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>, false>

Transition Spec List:

CR-0 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>> , Lower;
CR-1 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>> , Raise;
CR-2 <<Gate.opened, Train_Controller.<Train.idle, Controller.idle>>, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>> , Near;
CR-3 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>> , Down;
CR-5 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>, <Gate.opened, Train_Controller.<Train.idle, Controller.idle>>> , Up;
CR-7 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
CR-8 <<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>> , Out;
CR-9 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>> , Exit;
CR-10 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>> , Near;
CR-11 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>> , Exit;
CR-12 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>> , Near;
CR-13 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>> , Exit;
CR-14 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>> , Near;
CR-15 <<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , Exit;
CR-16 <<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , Near;
CR-17 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>> , Exit;
CR-18 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>> , Near;

Transition Spec List:

R_1 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+0/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Lower;

R_2 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+1/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Lower;

R_3 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+2/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Lower;

R_4 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+3/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Lower;

R_5 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+4/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Lower;

R_6 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+5/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Lower;

R_7 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+6/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Lower;

R_8 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+0/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+1/6> , 1/6;

R_9 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+1/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+2/6> , 1/6;

R_10 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+2/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+3/6> , 1/6;

R_11 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+3/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+4/6> , 1/6;

R_12 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+4/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+5/6> , 1/6;

R_13 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+5/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+6/6> , 1/6;

R_14 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6> , Raise;

R_15 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+1/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6> , Raise;

R_16 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+2/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6> , Raise;

R_17 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+3/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6> , Raise;

R_18 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+4/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6> , Raise;

R_19 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+5/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6> , Raise;

R_20 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+6/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6> , Raise;

R_21 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+1/6> , 1/6;

R_22 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+1/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+2/6> , 1/6;

R_23 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+2/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+3/6> , 1/6;

R_24 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+3/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+4/6> , 1/6;

R_25 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+4/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+5/6> , 1/6;

R_26 <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+5/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+6/6> , 1/6;

R_27 <<Gate.opened, Train_Controller.<Train.idle, Controller.idle>>, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+0/6> , Near;

R_28 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Down;
R_29 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Down;
R_30 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Down;
R_31 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Down;
R_32 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Down;
R_33 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Down;
R_34 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Down;
R_35 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> , 1/6;
R_36 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> , 1/6;
R_37 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> , 1/6;
R_38 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> , 1/6;
R_39 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> , 1/6;
R_40 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> , 1/6;
R_41 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+6/6, <Gate.opened, Train_Controller.<Train.idle, Controller.idle>>> , Up;
R_42 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+7/6, <Gate.opened, Train_Controller.<Train.idle, Controller.idle>>> , Up;
R_43 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+8/6, <Gate.opened, Train_Controller.<Train.idle, Controller.idle>>> , Up;
R_44 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+9/6, <Gate.opened, Train_Controller.<Train.idle, Controller.idle>>> , Up;
R_45 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+10/6, <Gate.opened, Train_Controller.<Train.idle, Controller.idle>>> , Up;
R_46 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+11/6, <Gate.opened, Train_Controller.<Train.idle, Controller.idle>>> , Up;
R_47 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+12/6, <Gate.opened, Train_Controller.<Train.idle, Controller.idle>>> , Up;
R_48 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+1/6> , 1/6;
R_49 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+1/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+2/6> , 1/6;
R_50 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+2/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+3/6> , 1/6;
R_51 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+3/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+4/6> , 1/6;
R_52 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+4/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+5/6> , 1/6;
R_53 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+5/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+6/6> , 1/6;
R_54 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+6/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+7/6> , 1/6;
R_55 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+7/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+8/6> , 1/6;

R_56 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+8/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+9/6> , 1/6;
R_57 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+9/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+10/6> , 1/6;
R_58 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+10/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+11/6> , 1/6;
R_59 <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+11/6, <Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+12/6> , 1/6;
R_60 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_61 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_62 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_63 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_64 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_65 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_66 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_67 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_68 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_69 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_70 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_71 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_72 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , In;
R_73 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> , 1/6;
R_74 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+1/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> , 1/6;
R_75 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+2/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> , 1/6;
R_76 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+3/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> , 1/6;
R_77 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+4/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> , 1/6;
R_78 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+5/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> , 1/6;
R_79 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+6/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+7/6> , 1/6;
R_80 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+7/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+8/6> , 1/6;
R_81 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+8/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+9/6> , 1/6;
R_82 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+9/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+10/6> , 1/6;
R_83 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+10/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+11/6> , 1/6;

R_84 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+11/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6> , 1/6;
R_85 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6> , 1/6;
R_86 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6> , 1/6;
R_87 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6> , 1/6;
R_88 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6> , 1/6;
R_89 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6> , 1/6;
R_90 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6> , 1/6;
R_91 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6> , 1/6;
R_92 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6> , 1/6;
R_93 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6> , 1/6;
R_94 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6> , 1/6;
R_95 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6> , 1/6;
R_96 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6> , 1/6;
R_97 <<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6> , Out;
R_98 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6> , 1/6;
R_99 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6> , 1/6;
R_100 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6> , 1/6;
R_101 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6> , 1/6;
R_102 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6> , 1/6;
R_103 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6> , 1/6;
R_104 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6> , 1/6;
R_105 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6> , 1/6;
R_106 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6> , 1/6;
R_107 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6> , 1/6;
R_108 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6> , 1/6;
R_109 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6> , 1/6;
R_110 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6> , 1/6;
R_111 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6> , 1/6;

R_112 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6> , 1/6;
R_113 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6> , 1/6;
R_114 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6> , 1/6;
R_115 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6> , 1/6;
R_116 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6> , 1/6;
R_117 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6> , 1/6;
R_118 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6> , 1/6;
R_119 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6> , 1/6;
R_120 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6> , 1/6;
R_121 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6> , 1/6;
R_122 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6> , 1/6;
R_123 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6> , 1/6;
R_124 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6> , 1/6;
R_125 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6> , 1/6;
R_126 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6> , 1/6;
R_127 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6> , 1/6;
R_128 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6> , 1/6;
R_129 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6> , 1/6;
R_130 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6> , 1/6;
R_131 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6> , 1/6;
R_132 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6> , 1/6;
R_133 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6> , 1/6;
R_134 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_135 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_136 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_137 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_138 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_139 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;

R_140 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_141 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_142 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_143 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_144 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_145 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_146 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_147 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_148 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_149 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_150 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_151 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_152 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_153 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_154 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_155 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_156 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_157 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_158 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_159 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_160 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_161 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_162 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_163 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_164 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_165 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_166 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_167 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;

R_168 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_169 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_170 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6, <Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> , Exit;
R_171 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+0/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+0/6> , Near;
R_172 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+1/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+1/6> , Near;
R_173 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+2/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+2/6> , Near;
R_174 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+3/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+3/6> , Near;
R_175 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+4/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+4/6> , Near;
R_176 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+5/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+5/6> , Near;
R_177 <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+6/6, <Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+6/6> , Near;
R_178 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Exit;
R_179 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> , Exit;
R_180 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> , Exit;
R_181 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> , Exit;
R_182 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> , Exit;
R_183 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> , Exit;
R_184 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> , Exit;
R_185 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> , Near;
R_186 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> , Near;
R_187 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> , Near;
R_188 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> , Near;
R_189 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> , Near;
R_190 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> , Near;
R_191 <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6, <Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> , Near;
R_192 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6> , Exit;
R_193 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6> , Exit;
R_194 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6> , Exit;
R_195 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6> , Exit;

R_196 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6> , Exit;
R_197 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6> , Exit;
R_198 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6> , Exit;
R_199 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6> , Exit;
R_200 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6> , Exit;
R_201 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6> , Exit;
R_202 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6> , Exit;
R_203 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6> , Exit;
R_204 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6> , Exit;
R_205 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6> , Near;
R_206 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6> , Near;
R_207 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6> , Near;
R_208 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6> , Near;
R_209 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6> , Near;
R_210 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6> , Near;
R_211 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6> , Near;
R_212 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6> , Near;
R_213 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6> , Near;
R_214 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6> , Near;
R_215 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6> , Near;
R_216 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6> , Near;
R_217 <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6, <Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6> , Near;
R_218 <<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , Exit;
R_219 <<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>, <Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , Near;
R_220 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6> , Exit;
R_221 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6> , Exit;
R_222 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6> , Exit;
R_223 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6> , Exit;

R_224 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6> , Exit;
R_225 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6> , Exit;
R_226 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6> , Exit;
R_227 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6> , Exit;
R_228 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6> , Exit;
R_229 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6> , Exit;
R_230 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6> , Exit;
R_231 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6> , Exit;
R_232 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6> , Exit;
R_233 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6> , Exit;
R_234 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6> , Exit;
R_235 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6> , Exit;
R_236 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6> , Exit;
R_237 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6> , Exit;
R_238 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6> , Exit;
R_239 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6> , Exit;
R_240 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6> , Exit;
R_241 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6> , Exit;
R_242 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6> , Exit;
R_243 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6> , Exit;
R_244 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6> , Exit;
R_245 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6> , Exit;
R_246 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6> , Exit;
R_247 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6> , Exit;
R_248 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6> , Exit;
R_249 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6> , Exit;
R_250 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6> , Exit;
R_251 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6> , Exit;

R_252 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6> , Exit;
R_253 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6> , Exit;
R_254 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6> , Exit;
R_255 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6> , Exit;
R_256 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6> , Exit;
R_257 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6> , Near;
R_258 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6> , Near;
R_259 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6> , Near;
R_260 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6> , Near;
R_261 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6> , Near;
R_262 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6> , Near;
R_263 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6> , Near;
R_264 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6> , Near;
R_265 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6> , Near;
R_266 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6> , Near;
R_267 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6> , Near;
R_268 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6> , Near;
R_269 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6> , Near;
R_270 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6> , Near;
R_271 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6> , Near;
R_272 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6> , Near;
R_273 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6> , Near;
R_274 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6> , Near;
R_275 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6> , Near;
R_276 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6> , Near;
R_277 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6> , Near;
R_278 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6> , Near;
R_279 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6> , Near;

R_280 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6> , Near;
R_281 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6> , Near;
R_282 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6> , Near;
R_283 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6> , Near;
R_284 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6> , Near;
R_285 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6> , Near;
R_286 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6> , Near;
R_287 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6> , Near;
R_288 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6> , Near;
R_289 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6> , Near;
R_290 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6> , Near;
R_291 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6> , Near;
R_292 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6> , Near;
R_293 <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6, <Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6> , Near;

Test Cases:

Class Gate_Train_Controller

State Coverage Cases
<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+0/6> : Near
<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> : Near, Lower
<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+1/6> : Near, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> : Near, Lower, Down
<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> : Near, Lower, 1/6
<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+2/6> : Near, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> : Near, Lower, Down, 1/6
<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> : Near, Lower, 1/6, 1/6
<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+3/6> : Near, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> : Near, Lower, Down, 1/6, 1/6
<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> : Near, Lower, 1/6, 1/6, 1/6
<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+4/6> : Near, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> : Near, Lower, Down, 1/6, 1/6, 1/6
<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> : Near, Lower, 1/6, 1/6, 1/6, 1/6
<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+5/6> : Near, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6
<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+6/6> : Near, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+7/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+8/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+9/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+10/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+11/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6
<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise
<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+1/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+1/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6
<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+2/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+2/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+3/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+3/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+4/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+4/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+5/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+5/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+6/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+6/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+7/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+8/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+9/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+10/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+11/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+12/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6
<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6> : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6

This is the end of State_Cover test cases.

Transition Coverage Cases

(<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+1/6> , <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6>) Raise : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, Raise
(<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+2/6> , <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6>) Raise : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, Raise
(<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+3/6> , <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6>) Raise : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6, Raise
(<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+4/6> , <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6>) Raise : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6, Raise
(<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+5/6> , <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6>) Raise : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6, 1/6, Raise
(<<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+6/6> , <<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+0/6>) Raise : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Raise
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6>) Down : Near, Lower, 1/6, Down
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6>) Down : Near, Lower, 1/6, 1/6, Down
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6>) Down : Near, Lower, 1/6, 1/6, 1/6, Down
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6>) Down : Near, Lower, 1/6, 1/6, 1/6, 1/6, Down
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6>) Down : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, Down

(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+0/6>) Down : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Down
(<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+6/6> , <<Gate.opened, Train_Controller.<Train.idle, Controller.idle>>>) Up : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Up
(<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+7/6> , <<Gate.opened, Train_Controller.<Train.idle, Controller.idle>>>) Up : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Up
(<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+8/6> , <<Gate.opened, Train_Controller.<Train.idle, Controller.idle>>>) Up : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Up
(<<Gate.toOpen, Train_Controller.<Train.idle, Controller.idle>>+9/6> , <<Gate.opened, Train_Controller.<Train.idle, Controller.idle>>>) Up : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit, Raise, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Up
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6> , <<Gate.closed, Train_Controller.<Train.idle, Controller.deactivate>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+0/6> , <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+0/6>) Near : Near, Near
(<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+1/6> , <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+1/6>) Near : Near, 1/6, Near
(<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+2/6> , <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+2/6>) Near : Near, 1/6, 1/6, Near
(<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+3/6> , <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+3/6>) Near : Near, 1/6, 1/6, 1/6, Near

(<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+4/6> . <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+4/6>) Near : Near, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+5/6> . <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+5/6>) Near : Near, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+6/6> . <<Gate.opened, Train_Controller.<Train.toCross, Controller.activate>>+6/6>) Near : Near, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6>) Exit : Near, Lower, Exit
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6>) Exit : Near, Lower, 1/6, Exit
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6>) Exit : Near, Lower, 1/6, 1/6, Exit
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6>) Exit : Near, Lower, 1/6, 1/6, 1/6, Exit
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6>) Exit : Near, Lower, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6>) Exit : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6>) Exit : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+0/6>) Near : Near, Lower, Near
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+1/6>) Near : Near, Lower, 1/6, Near
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+2/6>) Near : Near, Lower, 1/6, 1/6, Near
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+3/6>) Near : Near, Lower, 1/6, 1/6, 1/6, Near
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+4/6>) Near : Near, Lower, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+5/6>) Near : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6> . <<Gate.toClose, Train_Controller.<Train.toCross, Controller.monitor>>+6/6>) Near : Near, Lower, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6> . <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6> . <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6> . <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6> . <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6> . <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+12/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+13/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+14/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+15/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+16/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+17/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+18/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+19/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+20/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+21/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+22/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+23/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6> , <<Gate.closed, Train_Controller.<Train.toCross, Controller.monitor>>+24/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , <<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Exit
(<<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>> , <<Gate.closed, Train_Controller.<Train.cross, Controller.monitor>>>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Near
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6>) Exit : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Exit

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+0/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+1/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+2/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+3/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+4/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+5/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+6/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+7/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+8/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+9/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+10/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+11/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+12/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+13/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+14/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+15/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+16/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+17/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+18/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+19/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+20/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+21/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+22/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+23/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+24/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+25/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+26/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+27/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+28/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+29/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+30/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+31/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+32/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+33/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6> . <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+34/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+35/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near
(<<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6> , <<Gate.closed, Train_Controller.<Train.leave, Controller.monitor>>+36/6>) Near : Near, Lower, Down, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, In, Out, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, 1/6, Near

This is the end of Transition_Cover test cases.