Chapter 5

RTSR Software Tool

5.1 RTSR Overview

Based on the algorithms for the requirement-based regression test reduction using dependence analysis described in Chapter 3 and Chapter 4, a tool called Regression Test Suite Reduction (RTSR) has been developed as a part of the Test Suite Reduction/Generation software tool (TSRG) [17], which is implemented in Sun Solaris Sparc 5.8 using C++ and Java 2 Platform.

RTSR is built to reduce the number of regression test cases in a given test suite by employing interaction patterns computed for each test case. For each elementary modification, during traversal of a test case, three interaction patterns are computed: (1) affecting interaction pattern, (2) affected interaction pattern, and (3) side-effect interaction pattern. If the same interaction pattern of a certain type is computed for two different test cases for an elementary modification, these test cases are considered equivalent, wrt the elementary modification and the interaction pattern. A test case is included in the reduced regression test suite if at least one of its interaction patterns does not exist for any of the test cases in the reduced regression test suite. Figure 5.1 shows the structure of the RTSR tool.

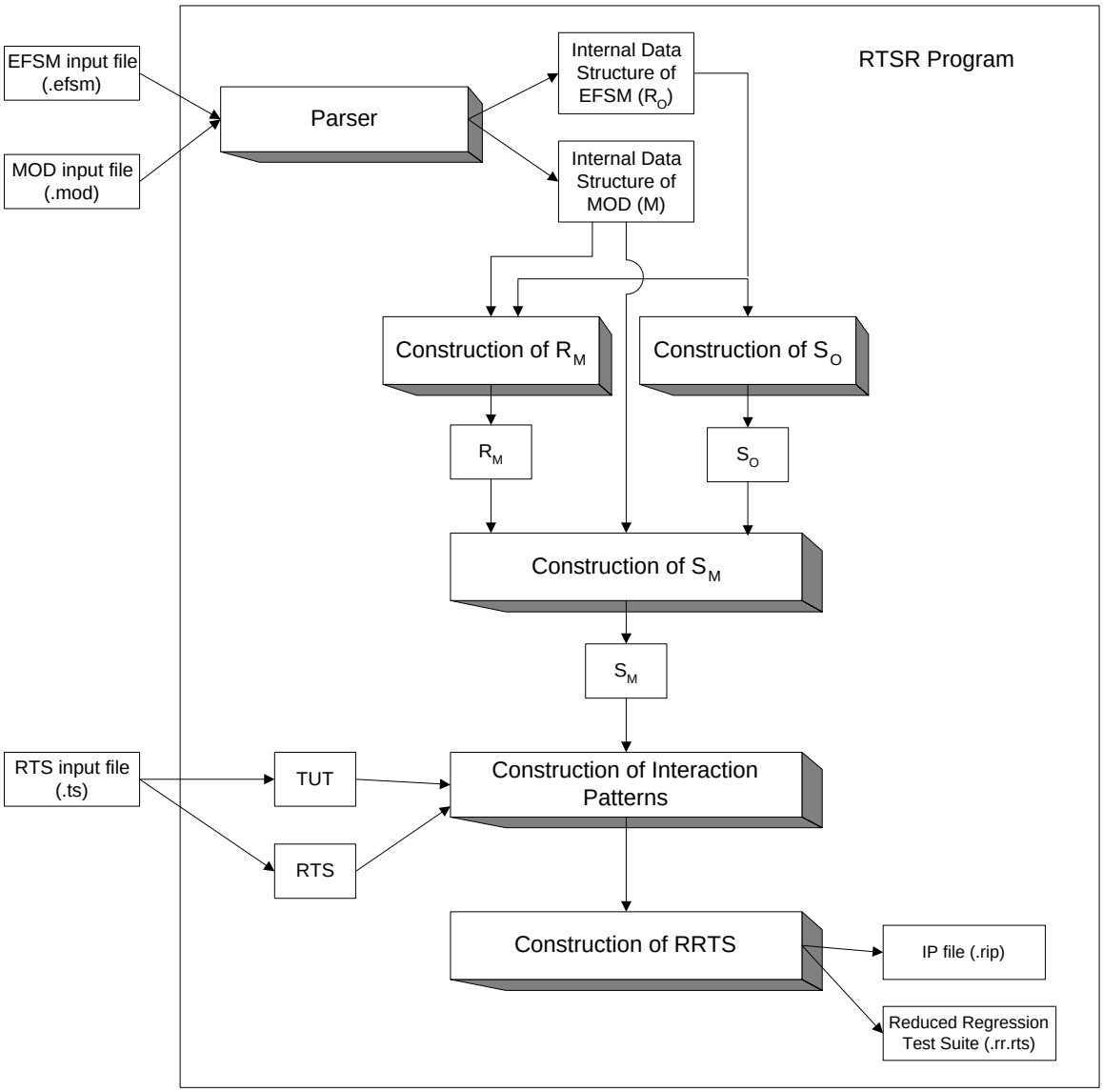

Figure 5.1 Structure of RTSR

The RTSR performs the following tasks:

Phase 1: Construction of SDG S_O and modified EFSM R_M

Phase 2: Construction of modified SDG S_M

Phase 3: Generation of Interaction Patterns

Phase 4: Construction of Reduced Regression Test Suite *RRTS*

The details of the implementations of these phases are presented in Section 5.4. Section 5.2 gives the input file formats of RTSR, and Section 5.3 gives the output file formats of RTSR.

It must be noted that the EFSM parser was developed by Tuong Nguyen to yield internal data structures EFSM R_O and MOD M [28]. The construction of SDG S_O was implemented by Gao Yan based on the algorithm for generating static dependence graph from an EFSM in [9].

5.2 Input File Formats

As shown in Figure 5.1, RTSR requires three input files: EFSM input file, MOD input file, and RTS input file. The EFSM input file is a file that represents an EFSM model, which is given the ".efsm" extension; the MOD input file is a file that represents a set of elementary modifications, which is given the ".mod" extension; and the RTS input file is a file that represents a test suite, which is given the ".ts" extension.

In this thesis, the EFSM input file is defined formally below using the Backus-Naur Form (BNF) [28]. Note that although some constructs of SDL such as set, reset, procedure are included in this definition, they are not considered in our work.

Table 5.1 BNF Definition of an EFSM Input File

<efsm> ::=
    efsmId
    numStates startStateIndex exitStateIndex
    <transitions>
<transitions> ::=

<transition> | <transitions> <transition>

<transition> ::=

transition transitionId

sourceStateIndex destinationStateIndex

<requirement>

<requirement> ::=

[<input>]

[<enablingPredicate>]

/

[<actions>]

<actions> ::=

<action> | <actions> <action>

<action> ::=

<output> | <assignment> | <set> | <reset> | <procedureCall>

<input> ::=

inputId **(** [<parameters>] **)**

<output> ::=

outputId **(** [<parameters>] **)**

<enablingPredicate> ::=

<variableIds> [**/*** BooleanExpression ***/**]

<assignment> ::=

<variableId> := <expression>

<set> ::=

set (constant , timerId **)**

<reset> ::=

reset (timerId **)**

procedureCall ::=

procedure (procedureId **(** <variableIds> [; <variableIds>] **)) {** <pbrDefs> **}**

<parameters> ::=

<parameter> {, <parameter>}*

<parameter> ::=

<variableId> | **constant**

<variableIds> ::=

<variableId> {, <variableId>}*

<pbrDefs> ::=

<pbrDef> | <pbrDefs> <pbrDef>

<pbrDef> ::=

<variableId> := <expression> ;

<expression> ::=

function (<variableIds> **)** | **constant**

<variableId> ::= id

The ".efsm" file for the EFSM of the simplified ATM system in Figure 2.1 is given in Appendix A.6.

The MOD input file is a set of elementary modifications where each elementary modification is represented by a modification type and the modified transition. A

modification type is either an addition or a deletion. The MOD input file is defined formally below in BNF:

Table 5.2 BNF Definition of a MOD Input File

<mod> ::=
    <modTransitions>

<modTransitions> ::=
    <modTransition> | <modTransitions> <modTransition>

<modTransition> ::=
    mtransition mType transitionId
    /* mType (addition = 0, deletion = 1, unknown = 2) */
    sourceStateIndex destinationStateIndex
    <requirement>

<requirement> ::=
    [<input>]
    [<enablingPredicate>]
    /
    [<actions>]

<actions> ::=
    <action> | <actions> <action>

<action> ::=
    <output> | <assignment> | <set> | <reset> | <procedureCall>

<input> ::=
    inputId **(** [<parameters>] **)**

<output> ::=
    outputId **(** [<parameters>] **)**

<enablingPredicate> ::=

<variableIds> [**/*** booleanExpression ***/**]

<assignment> ::=

<variableId> := <expression>

<set> ::=

set (constant , timerId **)**

<reset> ::=

reset (timerId **)**

procedureCall ::=

procedure (procedureId **(** <variableIds> [; <variableIds>] **)) {** <pbrDefs> **}**

<parameters> ::=

<parameter> {, <parameter>}*

<parameter> ::=

<variableId> | **constant**

<variableIds> ::=

<variableId> {**,** <variableId>}*

<pbrDefs> ::=

<pbrDef> | <pbrDefs> <pbrDef>

<pbrDef> ::=

<variableId> **:=** <expression> **;**

<expression> ::=

function (<variableIds> **)** | **constant**

<variableId> ::=

id

An example ".mod" file for the EFSM of the simplified ATM system in Figure 2.1 is given in Appendix A.7.

The RTS input file consists of a set of transitions under test, each of which is the transition related to an elementary modification and a collection of regression test cases. A test case is a complete sequence of transitions that starts at the start state and ends at the exit state of the EFSM. The RTS input file is defined formally below in BNF [28]:

Table 5.3 BNF Definition of a TS Input File

<ts> ::=

efsmId <tuts> <tests>

<tuts> ::=

<tut> | <tuts> <tut>

<tut> ::=

transitionId

<tests> ::=

<test> | <tests> <test>

<test> ::=

test testId <transitionSeq>

<transitionSeq> ::=

transitionId | <transitionSeq> transitioned

An example ".ts" file for the EFSM of the simplified ATM system in Figure 2.1 is given in Appendix A.8.

5.3 Output File Formats

RTSR generates two output files: a Reduced RTS output file and an IP output file. The Reduced RTS file is a file that represents the reduced regression test suite where redundant test cases have been eliminated. A Reduced RTS file is given the ".rr.ts" extension and has the same format as the original RTS file.

The IP file is a file that represents a set of interaction patterns wrt each *tut*. Each interaction pattern indicates a group of equivalent test cases that result in it. The test cases are referred to by their test id. The extension of the IP output file is ".rip". The IP output file is defined formally below in BNF:

Table 5.4 BNF Definition of an IP Output File

<ip> ::=

efsmId <tut> <ips>

<tut> ::=

transitionId

<ips> ::=

<ip> | <ips> <ip>

<ip> ::=

<ipType> ipId [<testIds>] <nodes>

<ipType> ::=

ip_affecting | ip_affected | ip_sideEffect

<testIds> ::=

testId | <testIds> testId

<nodes> ::=

<node> | <nodes> <node>

<node> ::=

node nodeIndex <label> [<adjacencySet>]

<label> ::=

transitionId

<adjacencySet> ::=

<reverseSet> | <nonreverseSet>

<reverseSet> ::=

<reverse> | <reverseSet> <reverse>

<reverse> ::=

inc sourceIndex <dependencyType>

<nonreverseSet> ::=

<nonreverse> | <nonreverseSet> <nonreverse>

<nonreverse> ::=

out destinationIndex <dependencyType>

<dependencyTye> ::=

dat | ctl | activation | affectingGhostDat | affectedGhostDat | ghostActivation

It is noted that an IP output file distinguishes three types of interaction patterns according to the ipType, i.e. “ip_affecting”, “ip_affected”, and “ip_sideEffect” that denote affecting interaction pattern, affected interaction pattern, and side-effect interaction pattern, respectively. The prefix of ipId “R” denotes that the three types of interaction patterns are generated for regression testing. An example “.rip” file for the EFSM of the simplified ATM system in Figure 2.1 is given in Appendix A.9.

5.4 RTSR Tool

RTSR uses EFSM model dependence analysis to reduce regression test suites. We assume that interactions between EFSM transitions are represented as EFSM dependencies between transitions. If the same interaction pattern of a certain type is computed for two different test cases for an elementary modification, these test cases are considered equivalent, wrt the elementary modification and the interaction pattern. A test case is included in the reduced test suite if at least one of its interaction patterns does not exist for any of the test cases in the reduced test suite [19]. RTSR can be broken down into four phases as mentioned in Section 5.1.

Phase 1: Construction of SDG S_O and Modified EFSM R_M

Given two input files, EFSM file and MOD file, RTSR concatenates the EFSM file and MOD file, then analyzes the concatenated file by lexical parser, and forms the EFSM and MOD internal data structures R_O and M, respectively. The SDG S_O of the original EFSM R_O can be built using EFSM internal data structure. The modified EFSM R_M can be built using R_O and M, which are described in Chapter 3. For example, from the EFSM input file given in Appendix A.6 and the MOD input file given in Appendix A.7, we can construct S_O and R_M. Parts of S_O and R_M are shown in Table 5.5 and Table 5.6, respectively. Table 5.5 represents the dependencies existing from transition T1 to transition T2. Table 5.6 represents the modified EFSM with added transition T9 between state S_2 and S_3.

Table 5.5 An Example of Dependencies from T1 to T2

EFSM id: ATM_System Original Static Dependency Graph (SDG):

Source node index: 0, Label: T1

Destination node index: 1, Label: T2

Number of edges: 3

List of edges: DType(Data=0,Control=1,CUse=2,PUse=3)

(Variable, DType, OOrder in def, OOrder in use)

(pin,0,2,3)

(attempts,0,3,2)

(attempts,0,3,5)

Table 5.6 An Example Internal Data Structure of Modified EFSM with Added Transition T9

EFSM id: ATM_System

Label: T9, Internal index: 8

Source state: 2

Destination state: 3

List of variables & occurrences: OType(Def=0,CUse=1,PUse=2)

(OType,Var,TLabel,OOrder)

(1,b,T9,1)

Number of components: 2

List of components:

AType(INPUT=0,OUTPUT=1,ASSIGN=2,SET=3,RESET=4,PRED=5,PROC=6)

Index,Id,AType,List(OType,Var,TLabel,OOrder)

0,Balance,0

1,Print,1,(1,b,T9,1)

Phase 2: Construction of Modified SDG S_M

In this phase, we have S_O, R_M, and M, and can construct the SDG S_M of modified EFSM. The detailed algorithm for constructing S_M is described in Chapter 3. A part of S_M which represents the dependencies existing from transition T1 to transition T9 of the EFSM of the simplified ATM system in Figure 2.1 is shown in Table 5.7.

Table 5.7: An Example of Internal Data Structure of S_M

Source node index: 0, Label: T1
Destination node index: 8, Label: T9
MType(ADD=0,DEL=1,REP=2,MUK=3): 3
Related edge: N/A
Number of edges: 1
List of edges: DType(Data=0,Control=1,CUse=2,PUse=3,
AFFNGDA=4,AFFEDDA=5,AD=6,AFFNGGAD=7,AFFEDGAD=8,
GAD=9,AFFNGCO=10,AFFEDCO=11,DUK=12)
(Variable, DType, OOrder in def, OOrder in use)
(b,4,1,1)

Phase 3: Generation of Interaction Patterns

In this phase, we have S_M. In order to generate interaction patterns, an RTS input file is required. As stated previously, an RTS input file, ".ts" consists of a set of elementary modifications, which is represented by a set of transitions under test (*TUT*) and a collection of regression test cases (*RTS*) where each regression test case (*rts*) is a transition sequence starting at the start state and ending at the exit state of the corresponding EFSM. RTSR extracts *TUT* and *RTS*. For each *tut*, RTSR obtains RTS_{tut}

which is the subset of *RTS* associated with a *tut.* For each *rts* in RTS_{tut}, three interaction patterns are computed: affecting interaction pattern, affected interaction pattern, and side-effect interaction pattern. The detailed algorithm for generating interaction patterns is described in Chapter 4.

Phase 4: Construction of Reduced Regression Test Suite *RRTS*

In this phase, a Reduced RTS output file, ".rr.ts" and an IP output file, ".rip" are constructed. For each elementary modification, we use Pattern1Set, pattern2Set, and Pattern3Set to store three types of interaction patterns. From Phase 3, for each test case, we obtained three interaction patterns, namely Pattern1, Pattern2, and Pattern3, which will be used to reduce the original test suite. If there exists an interaction pattern that is not in the Pattern1Set, Pattern2Set, or Pattern3Set, we insert this test case into the reduced regression test suite. RTSR reports the reduced regression test suite in an output file with ".rr.ts" extension. For each elementary modification, RTSR also identifies sets of equivalent test cases wrt a certain type of interaction pattern, and report it in an IP output file with ".rip" extension. For example, consider the addition of transition T9 to the EFSM of the simplified ATM system (Figure 2.1). The *tut* is T9 that represents an elementary modification of adding transition T9. Suppose the regression test suite contains the following two tests:

Test_1 = *T1, T4, T9, T7, T5, T7, T9, T7, T8*, and

Test_2 = *T1, T2, T4, T9, T7, T5, T7, T9, T7, T8*,

A part of *RRTS* for Test_1 and Test_2 is shown in Table 5.8. A part of IP output file shown in Table 5.9 represents the case that for transition under test T9, Test_1 and Test_2 are equivalent wrt the affecting interaction pattern.

Table 5.8: An Example of Reduced Regression Test Suite in *RRTS* File (.rr.ts file)

ATM_System
T9
test Test_1 *T1, T4, T9, T7, T5, T7, T9, T7, T8*

Table 5.9: An Example of Affecting Interaction Pattern for T9 in IP Output File (.rip file)

ATM_System
T9
ip_affecting RT9_0 Test_1 Test_2
node 0 T1
node 1 T4
inc 0 dat
node 2 T5
inc 1 ctl
inc 0 dat
node 3 T9
inc 2 dat
inc 1 ctl
inc 0 dat

5.5 Application of RTSR to an Example

We have applied RTSR developed in this thesis to the simplified ATM system of Figure 2.1. The requirements of the simplified ATM system are described in English in

Appendix A.1 [28]. The EFSM model for the simplified ATM system is presented in Appendix A.2.

Consider adding a balance inquiry transaction to the simplified ATM system, and deleting the deposit transaction from the simplified ATM system. The added balance inquiry transaction is represented by transition T9 and the deleted deposit transaction is represented by transition T6. The modified EFSM model of the simplified ATM system with added balance transaction and deleted deposit transaction is shown in Appendix A.3. From the modified EFSM model, the regression test suite is derived and represented in Appendix A.4.

RTSR accepts three inputs files: an EFSM input file (shown in Appendix A.6), a MOD input file (shown in Appendix A.7), and an RTS input file (shown in Appendix A.4). The RTS input file is constructed manually according to the IPO_2-df-Chains coverage criteria [37].

After applying RTSR, the interaction patterns for T9 and T6dummy with the equivalent test cases wrt a certain interaction pattern are shown in Table 5.10 and Table 5.11, respectively.

Table 5.10 The Interaction Patterns and the Equivalent Test Cases wrt a Certain Interaction Pattern for T9

Number of Interaction Patterns	Interaction Pattern	Equivalent Test Cases wrt a Certain Interaction Pattern (specified by test ids)
1	Affecting Interaction pattern #1	Test_2, Test_3, Test_5, Test_6
2	Affecting Interaction pattern #2	Test_7, Test_10, Test_13, Test_46, Test_49, Test_61, Test_64, Test_76, Test_79, Test_91, Test_92
3	Affecting Interaction Pattern #3	Test_8, Test_9, Test_11, Test_12, Test_14, Test_15

4	Affecting Interaction Pattern #4	Test_17, Test_18, Test_20, Test_21, Test_22, Test_23, Test_24, Test_25, Test_26, Test_27, Test_28, Test_29, Test_30, Test_32, Test_33, Test_35, Test_36, Test_37, Test_38, Test_39, Test_40, Test_41, Test_42, Test_43, Test_44, Test_45
5	Affecting Interaction Pattern #5	Test_47, Test_48, Test_50, Test_51, Test_52, Test_53, Test_54, Test_55, Test_56, Test_57, Test_58, Test_59, Test_60, Test_62, Test_63, Test_65, Test_66, Test_67, Test_68, Test_69, Test_70, Test_71, Test_72, Test_73, Test_74, Test_75, Test_77, Test_78, Test_80, Test_81, Test_82, Test_83, Test_84, Test_85, Test_86, Test_87, Test_88, Test_89, Test_90
6	Affected Interaction Pattern #1	Test_2, Test_3, Test_5, Test_6, Test_7, Test_8, Test_9, Test_10, Test_11, Test_12, Test_13, Test_14, Test_15, Test_17, Test_18, Test_20, Test_21, Test_22, Test_23, Test_24, Test_25, Test_26, Test_27, Test_28, Test_29, Test_30, Test_32, Test_33, Test_35, Test_36, Test_37, Test_38, Test_39, Test_40, Test_41, Test_42, Test_43, Test_44, Test_45, Test_46, Test_47, Test_48, Test_49, Test_50, Test_51, Test_52, Test_53, Test_54, Test_55, Test_56, Test_57, Test_58, Test_59, Test_60, Test_61, Test_62, Test_63, Test_64, Test_65, Test_66, Test_67, Test_68, Test_69, Test_70, Test_71, Test_72, Test_73, Test_74, Test_75, Test_76, Test_77, Test_78, Test_79, Test_80, Test_81, Test_82, Test_83, Test_84, Test_85, Test_86, Test_87, Test_88, Test_89, Test_90, Test_91, Test_92
7	Side-effect Interaction Pattern #1	Test_7, Test_8, Test_9, Test_10, Test_11, Test_12, Test_13, Test_14, Test_15, Test_46, Test_47, Test_48, Test_49, Test_50, Test_51, Test_52, Test_53, Test_54, Test_55, Test_56, Test_57, Test_58, Test_59, Test_60, Test_61, Test_62, Test_63, Test_64, Test_65, Test_66, Test_67, Test_68, Test_69, Test_70, Test_71, Test_72, Test_73, Test_74, Test_75, Test_76, Test_77, Test_78, Test_79, Test_80, Test_81, Test_82, Test_83, Test_84, Test_85, Test_86, Test_87, Test_88, Test_89, Test_90, Test_91, Test_92

Table 5.11 The Interaction Patterns and the Equivalent Test Cases wrt a Certain Interaction Pattern for T6dummy

Number of Interaction Patterns	Interaction Pattern	Equivalent Test Cases wrt a Certain Interaction Pattern (specified by test ids)
1	Affecting Interaction Pattern #1	Test_3, Test_12
2	Affecting Interaction Pattern #2	Test_4, Test_5, Test_7, Test_8, Test_13, Test_14, Test_31, Test_32, Test_37, Test_38, Test_46, Test_47, Test_52, Test_53, Test_76, Test_77, Test_82, Test_83, Test_92
3	Affecting Interaction pattern #3	Test_6, Test_9, Test_15
4	Affecting Interaction pattern #4	Test_18, Test_19, Test_20, Test_21, Test_24, Test_25, Test_26, Test_27, Test_28, Test_29, Test_30, Test_63, Test_64, Test_65, Test_66, Test_69, Test_70, Test_71, Test_72, Test_73, Test_74, Test_75
5	Affecting Interaction pattern #5	Test_33, Test_34, Test_35, Test_36, Test_39, Test_40, Test_41, Test_42, Test_43, Test_44, Test_45, Test_48, Test_49, Test_50, Test_51, Test_54, Test_55, Test_56, Test_57, Test_58, Test_59, Test_60, Test_78, Test_79, Test_80, Test_81, Test_84, Test_85, Test_86, Test_87, Test_88, Test_89, Test_90
6	Affected Interaction Pattern #1	Test_3, Test_12, Test_18, Test_24, Test_63, Test_69, Test_92
7	Affected Interaction Pattern #2	Test_4, Test_13,
8	Affected Interaction Pattern #3	Test_5, Test_14
9	Affected Interaction Pattern #4	Test_6, Test_8, Test_9, Test_15
10	Affected Interaction Pattern #5	Test_7
11	Affected Interaction Pattern #6	Test_19, Test_31, Test_34, Test_64, Test_76, Test_79
12	Affected Interaction Pattern #7	Test_20, Test_28, Test_29, Test_32, Test_35, Test_37, Test_38, Test_43, Test_44, Test_65, Test_73, Test_74, Test_77, Test_80, Test_82, Test_83, Test_88, Test_89
13	Affected Interaction Pattern #8	Test_21, Test_25, Test_26, Test_27, Test_30, Test_33, Test_36, Test_39, Test_40, Test_41, Test_42, Test_45, Test_47, Test_48, Test_50, Test_51, Test_52, Test_53, Test_54, Test_55,

		Test_56, Test_57, Test_58, Test_59, Test_60, Test_66, Test_70, Test_71, Test_72, Test_75, Test_78, Test_81, Test_84, Test_85, Test_86, Test_87, Test_90
14	Affected Interaction Pattern #9	Test_46, Test_49

The results of regression test suite reduction for T9 and T6dummy are shown in Table 5.12.

Let

RTS denote the number of regression test cases in a test suite,

RRTS denote the number of test cases in a reduced regression test suite, and

% denote the percentage of reduction

Table 5.12 Regression Test Suite Reduction of the Simplified ATM System

TUT	#RTS	#RRTS	%
T9	86	5	94
T6dummy	79	11	86

The results of the above example show that RTSR can be used to reduce the size of the regression test suite successfully and significantly. For example, for the simplified ATM system wrt the elementary modifications T6 and T9, a reduction of 86% to 94% is achieved.

Our example also shows that when applying regression test suite reduction based on the EFSM dependence analysis, the size of the reduced regression test suite is bounded by the number of possible interaction patterns associated with modifications.

For example, in the simplified ATM system, the possible number of affecting interaction pattern, affected interaction pattern, and side-effect interaction pattern wrt T9 are shown in Table 5.13, and the possible number of affecting interaction pattern,

affected interaction pattern, and side-effect interaction pattern wrt T6dummy are shown in Table 5.14.

Table 5.13 Possible Number of Interaction Patterns wrt T9

Number of Affecting Interaction Patterns	Number of Affected Interaction Patterns	Number of Side-effect Interaction Patterns
5	1	1

Table 5.14 Possible Number of Interaction Patterns wrt T6dummy

Number of Affecting Interaction Patterns	Number of Affected Interaction Patterns	Number of Side-effect Interaction Patterns
5	9	0

Interaction patterns are used to reduce the original test suite. For each elementary modification, three interaction patterns are computed during traversal of a test case. A test case is included in the reduced test suite if at least one of its interaction patterns does not exist for any of the tests in the reduced test suite. Therefore, the minimum size of the reduced test suite wrt T9 is equal to 5, and the minimum size of the reduced test suite wrt T6dummy is equal to 9. In the other words, the number of possible interaction patterns designates the minimum size of the reduced regression test suite regardless of the test strategy used in the regression test suite generation.

In the next chapter, we present our conclusions, with a summary of contributions and directions for future research.