Test Case Generation from a Z Specification of the Landing Gear System

Maximiliano Cristiá
CIFASIS and UNR, Rosario, Argentina
cristia@cifasis-conicet.gov.ar

In this technical report we present the results of a case study on the application of a model-based testing method (MBT) to a real-world problem from the aviation industry. The requirements were proposed by engineers working for the European aviation industry and comprise the landing gear system (LGS) of an aircraft. We developed a complete Z specification of the control software of the LGS. Then, we automatically generated abstract test cases by applying FASTEST (a tool implementing the Test Template Framework, which is a MBT method). These test cases cover all the functional and real-time scenarios described in the requirements. The manual work required to generate them is minimum.

Contents

1 Introduction **2**
1.1 Introduction to Model-Based Testing 3
1.1.1 Software Testing 3
1.1.2 Functional Correctness and Formal Specifications 4
1.1.3 Model-Based Testing 5

2 Introduction to the Z Notation **6**
2.1 The Requirements 6
2.2 The Form of a Z Specification 6
2.3 Basic Types 7
2.4 The State Space 7
2.5 Opening the First Savings Account 8
2.6 State Invariants 11

3 Introduction to the Test Template Framework **12**
3.1 The Valid Input Space of a Z Operation 12
3.2 Applying Testing Tactics 13
3.3 Building a Tree of Test Specifications 16
3.4 Pruning Inconsistent Test Specifications 17
3.5 Deriving Abstract Test Cases from Test Specifications 17
3.6 Brief Discussion of the TTF 18
3.6.1 Advantages of the TTF 18
3.6.2 The Form of Test Cases in the TTF 18
3.6.3 Test Oracles in the TTF 19
3.6.4 State Invariants in the TTF 19

4 The Z Specification of the Landing Gear System **20**
4.1 Basic Types . . . 20
4.2 State of the Controlling Software . . . 21
4.3 Initial states . . . 27
4.4 Assumptions and Limitations of the Z Specification . . . 29
4.5 Operations Concerning Sensor Readings and its Anomalies . . . 30
4.5.1 Gears in Extended Position . . . 31
4.5.2 Gears in Retracted Position . . . 32
4.5.3 Shock Absorbers . . . 33
4.5.4 Doors Open . . . 34
4.5.5 Doors Closed . . . 35
4.5.6 Hydraulic Circuit . . . 36
4.5.7 Analogical Switch . . . 37
4.6 Operations Concerning Normal Mode . . . 38
4.6.1 Interaction with the cockpit . . . 39
4.6.2 Retraction sequence . . . 40
4.6.3 Outgoing sequence . . . 48
4.7 Time Advance . . . 53
4.8 Operations Concerning Health Monitoring . . . 53
4.8.1 Anomalies Related to the Analogical Switch . . . 53
4.8.2 Anomalies Related to the Hydraulic Circuit . . . 55
4.8.3 Anomalies Related to Doors Motion . . . 56
4.8.4 Anomalies Related to Gears Motion . . . 57

5 Test Case Generation from the Z Specification **62**

6 Conclusions **62**

A Test Conditions **64**

B Abstract Test Cases **108**

1 Introduction

This answer to the case study shows how (abstract) test cases can be (almost) automatically generated from a Z specification of the landing gear system (LGS). In this way, the cost of writing the specification is paid-off by using the specification not only to produce the implementation but also to test it. Test case generation from a formal specification is within the scope of model-based testing (MBT). There are many different MBT methods to generate test cases [20, 26, 5]. Some of them use Z as the modeling language but there are methods for other formal (and semi-formal) notations, including some within the scope of ABZ 2014.

In this technical report a particular MBT method known as the Test Template Framework is used. The TTF uses Z as the modeling language and is particularly oriented towards *functional unit testing* [37]. Recently tool support for the TTF was provided by Fastest [11]. Even more recently Fastest started to use $\{log\}$ (pronounced 'setlog') as a test case generator [13]. $\{log\}$ is a Constraint Logic Programming (CLP) language that embodies the fundamental forms of set designation and a number of primitive operations for set management. Fastest automates most of the steps of a TTF testing campaing upt to the production of abstract test cases. Hence, the Z specification of the LGS was loaded in Fastest to automatically generate test cases.

Test cases produced by Fastest are abstract test cases in the sense that they are pieces of Z text. In other words, the variables and constants participating in one of these test cases are defined at the Z level. Hence, these test cases cannot be provided to the implementation of the LGS. However, they can be semi-automatically refined to the implementation by writing so-called refinement rules in a simple language that allows engineers to connect abstract test cases with the implementation technology [12]. Test case refinement is out of the scope of this document and currently is not fully implemented in Fastest.

The Z specification presented here specifies only the controlling software [6, Sect. 4]. It was not formally verified basically due to resource availability. However, it would be desirable to prove that it verifies some state invariants and, more importantly, that it verifies the expected properties proposed by the designers of the LGS [6, Sect. 5]. Had the authors have the time for such verification they would have used the Z/EVES theorem prover [33]. However, to be able to prove some of this properties it would be necessary to specify some domain properties described in the requirements document [6, Sect. 3] and perhaps the extensions to the Z notation proposed by Evans [16] would also be necessary.

This technical report introduces many concepts (such as MBT, the Z notation, the TTF, etc.) to make it self-contained. It also includes the complete list of the satisfiable test conditions (Appendix A) and abstract test cases (Appendix B) automatically generated by Fastest.

1.1 Introduction to Model-Based Testing

Software construction has proved to be more complex than expected. Most often software projects run beyond budget, are delivered late and having many errors. Only an insignificant portion of the products of the software industry are sold with warranty. There is a number of reasons for this state of the practice, but companies usually complain about the costs of software verification as the cause of not doing it thoroughly [8, page 20] [3, page 88] [30, page 157] [27] [32, table ES-1 at page ES-5]. Reducing the costs of verification would imply more projects within budget and less errors. One of the most promising strategies for reducing the costs of verification is making it as automatic as possible. On the other hand, the software industry relies almost exclusively on testing to perform the functional verification of its products. Currently, testing is essentially a manual activity that automates only the most trivial tasks [28, 32].

1.1.1 Software Testing

Software testing can be defined as the dynamic verification of a program by running it on a finite set of carefully chosen test cases, from the usually infinite input domain, and comparing the actual behavior with respect to the expected one [15, 38]. We want to remark the following:

- Testing implies running the program as opposed to, say, static analysis performed on the source code.
- The set of test cases on which the program will be executed is finite and usually very small, compared with the size of the input domain.
- These test cases must be selected, i.e there are some criteria or rules that must be followed in order to chose test cases. It would be wrong a selection process guided by the mood of the engineer.

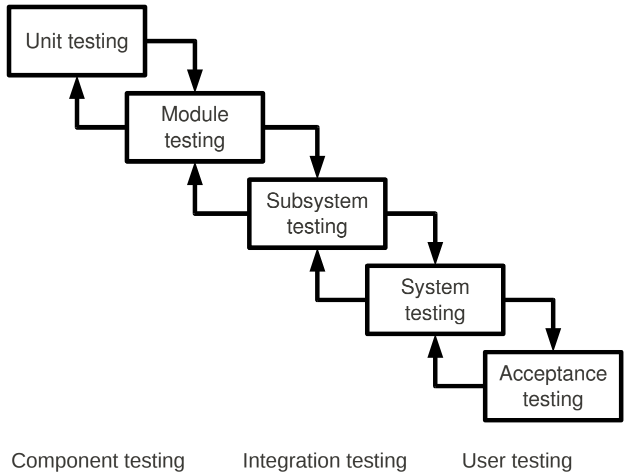

Figure 1: Steps of the testing process

- The output produced by the program for each test case must be compared with the expected output. If both agree then the program is correct on that test case; otherwise some error has been found. The artefact that helps to decide the presence of an error is called oracle.

Many qualities of a program can be tested. For example, performance, portability, usability, security and so on. Although they are all important, functional correctness is perhaps the one on which industry pays more attention. In many contexts, for instance, performing poorly is bad but performing wrongly is worse.

Traditionally, the testing process has been divided into five steps as shown in Figure 1. The idea is to start testing small portions of the system under test (SUT) called units—usually they are subroutines, procedures or functions—in such a way that once they have passed all the tests, they are progressively assembled together. As new units are integrated, the resulting modules are tested. Sometimes it is possible to independently test subsystems of the SUT. Finally, the full system is tested by users. In this way, errors are discovered as earlier as possible.

Fastest focuses on improving a particular unit testing method and providing tool support for the selection of functional test cases for it, as we will shortly see.

1.1.2 Functional Correctness and Formal Specifications

The last item above suggests that there must be some way of determining what the expected output of a program is. In other words, there should be a way of determining whether the program is functionally correct or not. The classical definition of functional correctness is: a program is functionally correct if it behaves according to its functional specification [19, page 17]. This means that two documents or descriptions are needed to perform functional verification: the program itself and its functional specification. In turn, this implies that functional testing is possible only if a specification of the program or SUT is present. The functional specification is sometimes used as the oracle because it is, in fact, the definition of correctness for its implementation.

Furthermore, if automation of the testing process is the goal, then some kind of formal specification is mandatory because otherwise mechanical analysis of the specification becomes unfeasible, turning testing automation unrealistic. A specification is formal if it is written in a formal notation or language

[19, page 167]. Formal notations or formalisms for specifying software systems are known as formal methods and have a long and well-established tradition within the Software Engineering community [7, 22].

Fastest focuses on functional testing based on a formal functional specification of the SUT.

1.1.3 Model-Based Testing

When testing and formal specifications are combined we enter into the scope of Model-Based Testing (MBT). MBT is a well-known technique aimed at testing software systems analyzing a formal model or specification of the SUT [38, 21]. That is, MBT approaches generate test cases from the formal specification of the SUT. The fundamental hypothesis behind MBT is that, as a program is correct if it satisfies its specification, then the specification is an excellent source of test cases.

One of the possible processes of testing a system through a MBT method is depicted in Figure 2. The first step is to analyze the model of the SUT looking for abstract test cases. Usually, MBT methods divides this step into two activities: firstly, test specifications are generated, and, secondly, abstract test cases are derived from them. Although the form of test specifications depends on the particular MBT method, they can be thought as sets of abstract test cases. Test cases produced during the "Generation" step are abstract in the sense that they are written in the same language of the model, making them, in most of the MBT methods, not executable. In effect, during the "Refinement" step these abstract test cases are made executable by a process that can be called *refinement*, *concretization* or *reification*. Note that this not necessarily means that the SUT has been refined from the model; it only says that abstract test cases must be refined. Once test cases have been refined they have to be executed by running the program on each of them. In doing so, the program produces some output for each test case. At this point, some way of using the model as an oracle, to decide whether a given test case has found an error or not, is needed. There are two possibilities depending on the MBT method and the formal notation being used:

1. When the model is analyzed during the "Generation" step, each abstract test case is bound to its corresponding expected result. Later, these expected results are refined along the same lines of test cases. Finally, the actual output of the program is compared with the result of refining the expected results.

2. The output produced by the SUT for each test case is abstracted at the level of the specification. Then, each abstract test case and its corresponding abstract(ed) output are replaced in the specification. If the specification reduces to *true* then no error was found; if it reduces to *false* then an error was found.

MBT has been applied to models written in different formal notations such as Z [37], Finite State Machines (FSM) and their extensions [20], B [26], algebraic specifications [5], and so on. However, most of the work has focused on the "Generation" step from some variant of FSM for system testing [21, 29]. One of the greatest advantages of working with FSM lays in the degree of automation that can be achieved by many MBT methods. On the other hand, FSM pose a strong limit on the kind of systems that can be specified.

Fastest provides support for the "Generation" step from Z specifications as a way to widen the class of systems that can be specified.

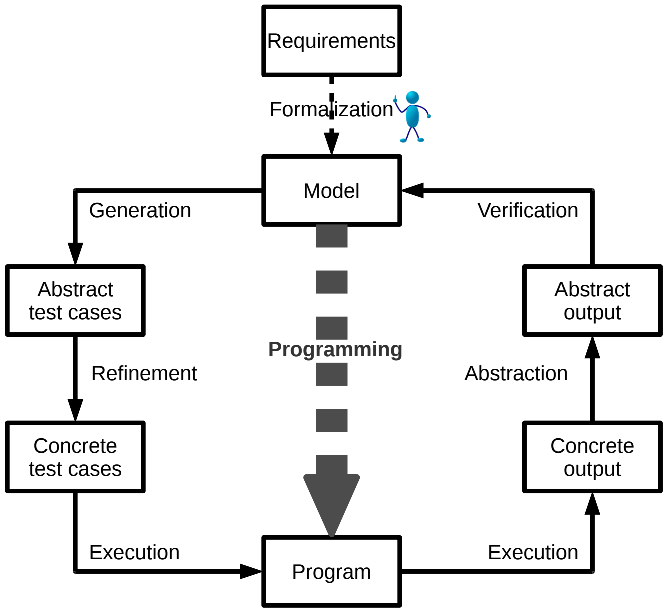

Figure 2: A general description of the MBT process

2 Introduction to the Z Notation

Here we introduce the Z notation by means of an example. It is assumed that the reader is fluent in predicate logic and discrete mathematics. Z is introduced just to the point needed to read the rest of this document; for deeper presentations consult any textbook on Z. The Z notation is a formal method based on first-order logic and Zermelo-Fraenkel set theory that has been extensively studied and applied to a range of software systems [22, 2]. There are two slightly different versions of the language. The first to appear is known as the Spivey version [35], and the second one is referred as Standard Z because it is the result of a standardization process carried out by ISO [23]. We will use the second one.

2.1 The Requirements

Think in the savings accounts of a bank. Each account is identified by a so-called account number. Clients can share an account and each client can own many accounts—some of which might be shared with other clients, and some not. The bank requires to keep record of just the balance of each account, and the ID and name of each client. Any person can open an account in the bank becoming its first owner. Owners can add and remove other owners and can withdraw money from their accounts and check the balance of their accounts. Any person can deposit money in any account.

2.2 The Form of a Z Specification

The Z language can be used in different ways but there is a de-facto usage style. Any Z specification takes the form of a state machine—not necessarily a finite one. This machine is defined by giving its state space and the transitions between those states. The state space is given by declaring a tuple of typed state variables. A transition, called operation in Z, is defined by specifying its signature, its preconditions and its postconditions. The signature of an operation includes input, state and output variables. Each operation can change the state of the machine. State change is described by giving the relation between before-state and after-state variables.

2.3 Basic Types

As we have said, each savings account is identified by an account number. We need a way to name these account numbers. Since account numbers are used just as identifiers we can abstract them away, not caring about their internal structure. Z provides so-called basic or given types for these cases. The Z syntax for introducing a basic type is:

$$[ACCNUM]$$

In this way, it is possible to declare variables of type *ACCNUM* and it is possible to build more complex types involving it—for instance the type of all sets of account numbers is $\mathbb{P} ACCNUM$. Along the same lines, we introduce basic types for the ID's of clients and their names:

$$[UID, NAME]$$

We represent the money that clients can deposit and withdraw and the balance of savings accounts as natural numbers. We think that specifying them as real numbers does not add any significant detail to the model, but makes it truly complicated since Z does not provides a native type for real numbers. If the decimal positions are really needed, then we can think that each natural number used in the specification is the result of multiplying the corresponding real number by a convenient power of 10—for instance, all amounts of money are multiplied by 100. The type for the integer numbers, $\mathbb{Z}$, is built-in in Z. The notation also includes the set of natural numbers, $\mathbb{N}$. Then, we define:

$$MONEY == \mathbb{N}$$

$$BALANCE == \mathbb{N}$$

In other words, we introduce two synonymous for the set of natural numbers so the specification is more readable.

2.4 The State Space

The state space is defined as follows:

Bank
$$clients : UID \pfun NAME$$
$$balances : ACCNUM \pfun BALANCE$$
$$owners : UID \leftrightarrow ACCNUM$$

This construction is called schema; each schema has a name that can be used in other schemas. In particular this is a state schema because it only declares state variables. In effect, it declares three state variables by giving their names and types. Each state of the system corresponds to a particular valuation of these three variables. The type constructor $\pfun$ defines partial functions[1]. Then, *clients* is a partial

[1] It must be noted that the Z type system is not as strong as the type systems of other formalisms, such as Coq [9]. So we will be as formal as is usual in the Z community regarding its type system.

function from *UID* onto *NAME*. It makes sense to define such a function because each person has a unique *UID* but not a unique name; and it makes sense to make *clients* partial because not every person is a client of the bank all the time. The same is valid for *balances*: there is a functional relationship between account numbers and balances, and not all the account numbers are used all the time in the bank. The symbol $\leftrightarrow$ defines binary relations. It is correct to define *owners* as a relation, and not as a function, because a given client may own more than one account and each savings account may be owned by many clients.

Now, we can define the initial state of the system as follows:

InitBank

$Bank$

$clients = \emptyset$
$balances = \emptyset$
$owners = \emptyset$

InitBank is another schema. The upper part is the declaration part and the lower part is the predicate part—this one is optional and is absent from *Bank*. In the declaration part we can declare variables or use schema inclusion. The latter means that we can write the name of another schema instead of declaring their variables. This allows us to reuse schemas. In this case the predicate part says that each variable is equal to the empty set. It is important to remark that the $=$ symbol is logical equality and not an imperative assignment—Z has no notion of control flow. In Z, relations and functions are sets of ordered pairs. Being sets they can be compared with the empty set. The symbol $\emptyset$ is polymorphic in Z: it is the same for all types.

Since in Z each variable has a type, all the expressions are typed and then it is possible to implement a type-checker for the language [33, 17].

2.5 Opening the First Savings Account

Now we can start defining each operation of the system. In order to keep this introduction short we specify just one operation and in doing so we introduce some more Z concepts. The operation describes how the first savings account is opened for a given person requesting it.

NewClientOk

$\Delta Bank$
$u? : UID$
$name? : NAME$
$n? : ACCNUM$

$u? \notin \operatorname{dom} clients$
$n? \notin \operatorname{dom} balances$
$clients' = clients \cup \{u? \mapsto name?\}$
$balances' = balances \cup \{n? \mapsto 0\}$
$owners' = owners \cup \{u? \mapsto n?\}$

The expression $\Delta Bank$ in the declaration part is a shorthand for including the schemas *Bank* and *Bank*$'$. We already know what it means including *Bank*. *Bank*$'$ is equal to *Bank* but all of its variables are decorated with a prime. Therefore, $Bank'$ declares $clients'$, $balances'$ and $owners'$ of the same types than those in $Bank$. When a state variable is decorated with the prime it is assumed to be an after-state variable. The net effect of including $\Delta Bank$ is, then, the declaration of three before-state variables and three after-state variables. A Δ expression is included in every operation schema that produces a state change.

Variables decorated with a question mark, like $u?$, are assumed to be input variables. Then, $u?$ represents the ID of the person willing to open a savings account in the bank, and $name?$ is his/her name. To simplify the specification a little bit we assume that a bank's clerk provides the account number, $n?$, when the operation is called—instead of the system generating it.

Note that the predicate part consists of five atomic predicates. When two or more predicates are in different rows they are assumed to be a conjunction. In other words, for instance:

$$u? \notin \operatorname{dom} clients$$
$$n? \notin \operatorname{dom} balances$$

is equivalent to:

$$u? \notin \operatorname{dom} clients \land n? \notin \operatorname{dom} balances$$

Z uses the standard symbols of discrete mathematics and set theory so we think it will not be difficult for the reader to understand each predicate. Remember that functions and relations are sets of ordered pairs so they can participate in set expressions. For instance, $balances \cup \{n? \mapsto 0\}$ adds an ordered pair to $clients$. Again, the expression $balances' = balances \cup \{n? \mapsto 0\}$ is actually a predicate saying that $balances'$ is equal to $balances \cup \{n? \mapsto 0\}$, and not that the latter is assigned to $balances'$. In other words, this predicate says that the value of $balances$ in the after-state is equal to the value of $balances$ in the before-state plus the ordered pair $n? \mapsto 0$.

Note that operations are defined by giving their preconditions and postconditions. In *NewClientOk* the preconditions are:

$$u? \notin \operatorname{dom} clients$$
$$n? \notin \operatorname{dom} balances$$

while its postconditions are:

$$clients' = clients \cup \{u? \mapsto name?\}$$
$$balances' = balances \cup \{n? \mapsto 0\}$$
$$owners' = owners \cup \{u? \mapsto n?\}$$

Therefore, *NewClientOK* does not say what the system shall do when $u? \notin \operatorname{dom} clients \land n? \notin \operatorname{dom} balances$ does not hold. The bank says that nothing has to be done when either the person requesting the account is already a client or when the account number chosen by the clerk is already in use. Then, we define a new schema for the first case:

$$ClientAlreadyExists == [\Xi Bank;\ u? : UID | u? \in \operatorname{dom} clients]$$

This is another way of writing schemas, called horizontal form. It has the same meaning than:

$$
\begin{array}{|l}
\hline
\textit{ClientAlreadyExists} \\
\Xi Bank \\
u? : UID \\
\hline
u? \in \operatorname{dom} clients \\
\hline
\end{array}
$$

The expression $\Xi Bank$ is a shorthand for:

$$
\begin{array}{|l}
\hline
\Xi Bank \\
\Delta Bank \\
\hline
clients' = clients \\
balances' = balances \\
owners' = owners \\
\hline
\end{array}
$$

If a Ξ expression is included in an operation schema, it means that the operation will not produce a state change because all the primed state variables are equal to their unprimed counterparts. When a schema whose predicate part is not empty is included in another schema, the net effect is twofold: (a) the declaration part of the former is included in the declaration part of the latter; and (b) the predicate of the former is conjoined to the predicate of the latter. Hence, *ClientAlreadyExists* could have been written as follows:

$$
\begin{array}{|l}
\hline
\textit{ClientAlreadyExists} \\
clients, clients' : UID \rightarrow\!\!\!\!\!\!+\; NAME \\
balances, balances' : ACCNUM \rightarrow\!\!\!\!\!\!+\; BALANCE \\
owners, owners' : UID \leftrightarrow ACCNUM \\
u? : UID \\
\hline
u? \in \operatorname{dom} clients \\
clients' = clients \\
balances' = balances \\
owners' = owners \\
\hline
\end{array}
$$

We define the following schema for the negation of the remaining precondition:

$$
\begin{array}{l}
AccountAlreadyExists == \\
\qquad [\Xi Bank;\ n? : ACCNUM \mid n? \in \operatorname{dom} balances]
\end{array}
$$

Usually, schemas like *NewClientOk* are said to specify the successful cases or situations, while schemas like *ClientAlreadyExists* and *AccountAlreadyExists* specify the erroneous cases. Finally, we assemble the three schemas to define the total operation—i.e. an operation whose precondition is equivalent to *true*—for a person opening his/her first savings account in the bank:

$$
\begin{array}{l}
NewClient == \\
\qquad NewClientOk \\
\qquad \lor ClientAlreadyExists \\
\qquad \lor AccountAlreadyExists
\end{array}
$$

NewClient is defined by a so-called schema expression. Schema expressions are expressions involving schema names and logical connectives. They can be very complex but we will not need all this complexity in this thesis. Let A be the schema defined as $[D_A | P_A]$ where D_A is the declaration part and P_A is its predicate. Similarly, let B the schema defined by $[D_B | P_B]$. Then, the schema C defined by $A \circledast B$, where $\circledast$ is any of $\wedge$, $\vee$ and $\Rightarrow$, is the schema $[D_A;\ D_B | P_A \circledast P_B]$. In other words, the declaration parts of the schemas involved in a schema expression are joined together and the predicates are connected with the same connectors used in the expression—if there is some clash in the declaration parts it must be resolved by the user. In symbols:

$$A == [D_A | P_A]$$

$$B == [D_B | P_B]$$

$$C == A \circledast B, \text{ where } \circledast \text{ is any of } \wedge, \vee, \Rightarrow \text{ then}$$

$$C == [D_A;\ D_B | P_A \circledast P_B]$$

Essentially, this is all the reader needs to know about Z to understand the rest of this thesis. Actually, Fastest, the tool we have developed, does not support the whole language, but it does support a fully expressive subset, as we discuss in Chapter **??**. Therefore, we now introduce the rest of the savings account specification but including informal comments only when some new Z feature is introduced.

2.6 State Invariants

A predicate is said to be a state invariant if it holds in every state of the system. The usual Z style includes state invariants in the state schema. For example, the state schema for the savings account system would have been:

```
┌─ Bank ──────────────────────────────
│ clients : UID ⇸ NAME
│ balances : ACCNUM ⇸ BALANCE
│ owners : UID ↔ ACCNUM
├──────────
│ dom clients = dom owners
│ dom balances = ran owners
│ ran balances ⊆ ℕ
└─────────────────────────────────────
```

instead of the one we have defined at the beginning of this section. Note that, in this way the state invariant is conjoined to the predicate part of every schema where $\Delta Bank$ or $\Xi Bank$ are included. This is a simple technique that guarantees that every operation will preserve the state invariant.

However, for reasons that we are going to explain in Section 3.6.4, we deal with state invariants in a different fashion. We first define a schema stating the invariant:

```
┌─ BankInv ───────────────────────────
│ Bank
├──────────
│ dom clients = dom owners
│ dom balances = ran owners
│ ran balances ⊆ ℕ
└─────────────────────────────────────
```

and then we require a proof obligation stating that each operation preserves it. For example:

Theorem NewClientInv
$BankInv \wedge NewClient \Rightarrow BankInv'$

Discharging such proof obligations is a responsibility of those who write the specification. This way of writing invariants is similar to other formal methods such as TLA+ [25] and B [1].

3 Introduction to the Test Template Framework

As we have said, the TTF is a particular method for the "Generation" step of the MBT process (Figure 2), specially well suited for unit testing from Z specifications. Each operation within the specification is analysed to derive or generate abstract test cases. This analysis consists of the following steps:

1. Consider the VIS of each Z operation
2. Apply one or more testing tactics in order to partition the input space
3. Build a tree of test specifications
4. Prune inconsistent test specifications
5. Find one abstract test case from each remaining test specification

Before executing the first step, engineers have to select those schemas that are operations. In effect, not all schemas representing operations have to be selected because some of them are used in the definition of others. For example, in the specification of the savings account system *ClientAlreadyExists* and *AccountAlreadyExists*, among others, will not be selected because by selecting *NewClient* test cases for them would also be generated.

3.1 The Valid Input Space of a Z Operation

The VIS of a Z operation is derived from the its Input Space (IS). The IS of an operation is the schema declaring all the state and input variables declared in the operation. For example, the IS of *NewClient* is:

$$\begin{aligned} &NewClient_{IS} == \\ &\quad [clients : UID \rightarrowtail NAME;\ owners : UID \leftrightarrow ACCNUM; \\ &\quad balances : ACCNUM \rightarrowtail BALANCE;\ u? : UID; \\ &\quad name? : NAME;\ n? : ACCNUM] \end{aligned}$$

The VIS is the schema that restricts the IS to verify the precondition of the operation:

$$Op_{VIS} == [Op_{IS} | \mathrm{pre}\ Op]$$

Informally, the precondition of an operation is that part of its predicate that does not contain output nor primed-state variables. Z provides the pre operation which takes a schema and returns its precondition. The VIS of a total operation is equal to its IS, since, by definition, its precondition is equivalent to *true*. *NewClient* is a total operation, therefore, we have:

$$NewClient_{VIS} == NewClient_{IS}$$

3.2 Applying Testing Tactics

The key aspect of the TTF is to partition the VIS of each operation into equivalence classes by applying one or more testing tactics. These equivalence classes are called *test classes*, *test objectives*, *test templates* or *test specifications*; we will use the latter. In other words a test specification S of some operation Op is a set such that $S \subseteq Op_{VIS}$. Test specifications obtained in this way can be further subdivided into more test specifications by applying other testing tactics. The net effect of this technique is a progressive partition of the VIS into more restrictive test specifications. This procedure can continue until the engineer is satisfied with the possible accuracy of the test specifications with respect to their ability to uncover errors in the implementation. Once the engineering is done with partitioning, she/he has to take one abstract test case from each resulting test specification.

Although, theoretically, testing tactics should produce a partition, in practice this is not always the case. Producing a partition is relevant because in some way it guarantees both full coverage of the VIS and non repetition of test cases. Given a VIS S a partition for S is a family of test specifications $\{S_i\}_{i \in I}$ for some set of indexes I, such that:

$$\bigcup_{i \in I} S_i = S \tag{1}$$

$$S_i \cap S_j = \emptyset \text{ for all } i,j \in I \text{ and } i \neq j \tag{2}$$

Therefore, by taking an abstract test cases from every S_i, S is fully covered and no two test cases test the same. On the contrary, if $\{S_i\}_{i \in I}$ is not a partition then either something is not tested or two or more test cases will test the same. In this sense, the TTF relies on the uniformity hypothesis [18], which can be stated as follows:

> *Uniformity hypothesis.* Let S_i be a test specification of some partition for some program P. Let t_1 and t_2 be two elements of S_i. The uniformity hypothesis says, then, that P passes t_1 if and only if P passes t_2.

In other words, the hypothesis says that each test specification is an equivalence class with respect to the way the program behaves for any element of it. Although this hypothesis cannot be proved many MBT methods rely on it [21]. From a safety perspective, if a testing tactic does not produce a partition of a test specification, then it should at least verify equation (1).

Therefore, testing tactics are the tools that testers have to partition the VIS of each operation. A tactic indicates how the current test specification must be partitioned by giving a set of predicates that are used to define each partition. Each of these predicates is called characteristic predicate; i.e. it characterizes a test specification. Each testing tactic partitions a test specification in a different way aiming at producing test cases to test different aspects of a program.

The original authors of the TTF proposed some testing tactics [36, 37] and we propose more (see Chapter **??**). In this section we will apply two of the original tactics to the *NewClient* operation. The first one we will apply, called DNF, was proposed even before the TTF [14], and in general it does not produce a partition . It says:

1. Write the predicate of the operation in DNF.

   Writing a predicate in DNF means writing it as a disjunction of conjunctions of atomic predicates or negations of atomic predicates.

2. Take the precondition of each resulting disjunct.

$$
\begin{array}{ll}
S = \emptyset, T = \emptyset & S \neq \emptyset, T \neq \emptyset, S \subset T \\
S = \emptyset, T \neq \emptyset & S \neq \emptyset, T \neq \emptyset, T \subset S \\
S \neq \emptyset, T = \emptyset & S \neq \emptyset, T \neq \emptyset, T = S \\
S \neq \emptyset, T \neq \emptyset, S \cap T = \emptyset & S \neq \emptyset, T \neq \emptyset, S \cap T \neq \emptyset, \neg (S \subseteq T), \\
 & \quad \neg (T \subseteq S), S \neq T
\end{array}
$$

Figure 3: Standard partition for $S \cup T$, $S \cap T$ and $S \setminus T$

3. Take these predicates as the characteristic predicates of the partition.

Let's apply it on *NewClient* before explaining it informally. The first step, in this example, is easy because the operation is already in DNF. Then, we get the following test specifications:

$$
\begin{array}{l}
NewClient_1^{DNF} == \\
\quad [NewClient_{VIS} | u? \notin \operatorname{dom} clients \land n? \notin \operatorname{dom} balances]
\end{array}
$$

$$NewClient_2^{DNF} == [NewClient_{VIS} | u? \in \operatorname{dom} clients]$$

$$NewClient_3^{DNF} == [NewClient_{VIS} | n? \in \operatorname{dom} balances]$$

First, note that test specifications are written in Z. This is a virtue of the TTF since it keeps all the main artifacts within the same notation. Second, note how test specifications are linked to the VIS by schema inclusion. Third, observe that this is not a partition of the VIS because, for example, the following abstract test case satisfies $NewClient_2^{DNF}$ and $NewClient_3^{DNF}$:

$$
\begin{array}{l}
clients = \{uid0 \mapsto name0\}, \\
balances = \{accnum0 \mapsto 1\}, \\
owners = \{uid0 \mapsto name0\}, \\
u? = uid0, \\
n? = accnum0
\end{array}
$$

However, DNF is a very good tactic because it generates test specifications that will test the main functional alternatives of the implementation. Nevertheless, it does not produce test cases to test the implementation of complex mathematical operators, as noted Stocks and Carrington in their seminal papers. In effect, most likely a Z specification will have mathematical operators—such us $\cup$ and $\oplus$—that do not have a basic representation in most programming languages; i.e. they need a non trivial implementation.

For this cases, the TTF proposes a powerful testing tactic called SP. This tactic is parametrized by a mathematical operator. Then, the characteristic predicates indicated by SP to partition a test specification depends on a mathematical operator. Figure 3 shows the characteristic predicates proposed for the standard partition of $\cup$, $\cap$ or $\setminus$ [37]. Note that other partitions can be proposed and used. For example, a partition containing only the first two characteristic predicates plus $S \neq \emptyset, T \neq \emptyset$ is possible—although it will tend to uncover less errors than the one in Figure 3.

Hence, engineers must analyze the specification of an operation looking for mathematical operators that they think it is likely that will have a complex implementation that would lead to errors in the program. Then, they must propose a standard partition for them—if none has been defined or they think the one defined does not suit their needs—or use an existing one. Finally, they have to partition one or more of the current test specification by conjoining the characteristic predicates of the standard partition. If the same mathematical operator appears more than once in the operation then they have to decide on which expression they are going to apply the tactic. If they want to apply the tactic to another instance of the operator or to other operators, this can be considered as the application of another tactic, so they can repeat the process after applying the first SP.

We will apply SP to $\cup$ in $clients \cup \{u? \mapsto name?\}$ in order to partition just $NewClient_1^{DNF}$. That is, we will not partition $NewClient_2^{DNF}$ and $NewClient_3^{DNF}$ by SP, because, in this case, we assess that partitioning them in this way will not lead to a better coverage. This is so because if a test case meets the conditions of $NewClient_2^{DNF}$ or $NewClient_3^{DNF}$, it will unlikely make the program to execute the code where $\cup$ is implemented. Besides, in this first application of SP we will not produce test specifications aimed at testing the correct implementation of $balances \cup \{n? \mapsto 0\}$ and $owners \cup \{u? \mapsto n?\}$. We will not do that in this thesis to keep the example manageable, although in practice it should be done.

In summary, applying SP to $\cup$ in $clients \cup \{u? \mapsto name?\}$ to partition $NewClient_1^{DNF}$ yields the following new test specifications:

$$NewClient_1^{SP} == [NewClient_1^{DNF} | clients = \emptyset \land \{u? \mapsto name?\} = \emptyset]$$

$$NewClient_2^{SP} == [NewClient_1^{DNF} | clients = \emptyset \land \{u? \mapsto name?\} \neq \emptyset]$$

$$NewClient_3^{SP} == [NewClient_1^{DNF} | clients \neq \emptyset \land \{u? \mapsto name?\} = \emptyset]$$

$$NewClient_4^{SP} == [NewClient_1^{DNF} | clients \neq \emptyset \land \{u? \mapsto name?\} \neq \emptyset \land clients \cap \{u? \mapsto name?\} = \emptyset]$$

$$NewClient_5^{SP} == [NewClient_1^{DNF} | clients \neq \emptyset \land \{u? \mapsto name?\} \neq \emptyset \land clients \subset \{u? \mapsto name?\}]$$

$$NewClient_6^{SP} == [NewClient_1^{DNF} | clients \neq \emptyset \land \{u? \mapsto name?\} \neq \emptyset \land \{u? \mapsto name?\} \subset clients]$$

$$NewClient_7^{SP} == [NewClient_1^{DNF} | clients \neq \emptyset \land \{u? \mapsto name?\} \neq \emptyset \land clients = \{u? \mapsto name?\}]$$

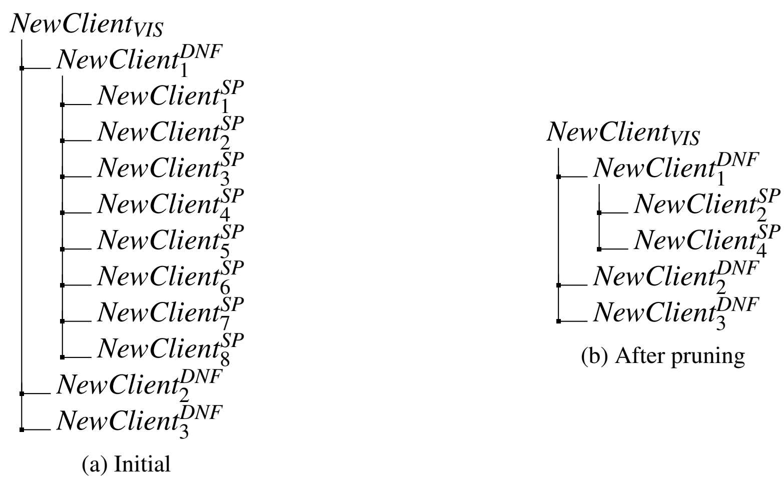

Figure 4: Initial and pruned testing trees of *NewClient*

$$
\begin{array}{l}
NewClient_8^{SP} == \\
\quad [NewClient_1^{DNF} \mid \\
\qquad clients \neq \emptyset \\
\qquad \land \{u? \mapsto name?\} \neq \emptyset \\
\qquad \land clients \cap \{u? \mapsto name?\} \neq \emptyset \\
\qquad \land \neg (clients \subseteq \{u? \mapsto name?\}) \\
\qquad \land \neg (\{u? \mapsto name?\} \subseteq clients)]
\end{array}
$$

Note that, again, test specifications are linked to each other by schema inclusion. Also observe that applying SP is no more that substituting the "formal" operands appearing in the partition definition by the "real" operands appearing in the selected expression.

As we have said, we will stop applying testing tactics in this example to keep it small. However, in a real project some more testing tactics can and should be applied to this operation. We will show the application of other tactics to other operations of the savings accounts system in Chapter **??**.

3.3 Building a Tree of Test Specifications

According to the TTF, the test specifications of a given operation must be organized in a so called testing tree. The testing tree has the VIS at the root, the test specifications generated after applying the first testing tactic form the first level and so forth. The testing tree of *NewClient* is shown in Figure 4(a).

Testing trees are important because the TTF prescribes deriving abstract test cases only from their leaves. This is so because each leaf conjoins the predicate of the test specifications above it up to the root, thus making no sense to derive abstract test cases from the internal nodes. For instance, if $NewClient_1^{DNF}$

is unfolded in $NewClient_4^{SP}$, the result is as follows:

$$
\begin{array}{l}
NewClient_4^{SP} == \\
\quad [NewClient_{VIS} | \\
\quad\quad u? \notin \operatorname{dom} clients \\
\quad\quad \land n? \notin \operatorname{dom} balances \\
\quad\quad \land clients \neq \emptyset \\
\quad\quad \land \{u? \mapsto name?\} \neq \emptyset \\
\quad\quad \land clients \cap \{u? \mapsto name?\} = \emptyset]
\end{array}
$$

Therefore, a test case that satisfies $NewClient_4^{SP}$ will also satisfy $NewClient_1^{DNF}$.

These trees can be automatically obtained from the test specifications since children include a reference to their parent node by schema inclusion, as can be seen in the test specifications shown above.

3.4 Pruning Inconsistent Test Specifications

Some test specifications might be empty because their predicates are unsatisfiable. In these cases it is impossible to find abstract test cases. Hence, inconsistent test specifications must be pruned from the testing trees. For instance, $NewClient_1^{SP}$ is inconsistent because $\{u? \mapsto name?\}$ cannot be empty. Another example is $NewClient_7^{SP}$, because *clients* cannot be equal to $\{u? \mapsto name?\}$ since $u? \notin \operatorname{dom} clients$ also holds. In our example, the testing tree resulting after pruning is depicted in Figure 4(b).

3.5 Deriving Abstract Test Cases from Test Specifications

Finally, the engineer has to choose at least one element satisfying each of the remaining leaves of the testing tree. These are the abstract test cases. For example, the following horizontal schemas represent abstract test cases of the corresponding test specifications:

$$
\begin{array}{l}
NewClient_1^{TC} == \\
\quad [NewClient_2^{SP} | \\
\quad\quad balances = \emptyset \\
\quad\quad \land u? = uid0 \\
\quad\quad \land clients = \emptyset \\
\quad\quad \land n? = accnum0 \\
\quad\quad \land name? = name0 \\
\quad\quad \land owners = \emptyset]
\end{array}
$$

$$
\begin{array}{l}
NewClient_2^{TC} == \\
\quad [NewClient_4^{SP} | \\
\quad\quad balances = \emptyset \\
\quad\quad \land u? = uid0 \\
\quad\quad \land clients = \{(uid1, name0)\} \\
\quad\quad \land n? = accnum0 \\
\quad\quad \land name? = name0 \\
\quad\quad \land owners = \emptyset] \\
NewClient_3^{TC} == \\
\quad [NewClient_2^{DNF} | \\
\quad\quad balances = \emptyset \\
\quad\quad \land u? = uid0 \\
\quad\quad \land clients = \{(uid0, name0)\} \\
\quad\quad \land n? = accnum0 \\
\quad\quad \land name? = name0 \\
\quad\quad \land owners = \emptyset] \\
NewClient_4^{TC} == \\
\quad [NewClient_3^{DNF} | \\
\quad\quad balances = \{(accnum0, 0)\} \\
\quad\quad \land u? = uid0 \\
\quad\quad \land clients = \emptyset \\
\quad\quad \land n? = accnum0 \\
\quad\quad \land name? = name0 \\
\quad\quad \land owners = \emptyset]
\end{array}
$$

Note that, once more, abstract test cases are also written in Z and how they are linked to test specifications by schema inclusion.

Fastest eliminates unsatisfiable test specification and finds abstract test cases by using the $\{log\}$ tool (pronounced 'setlog') [31, 13]. Both activities are almos automatic for users.

3.6 Brief Discussion of the TTF

We do not pretend here to compare the TTF with other approaches because it has been done when it was first published [37] and more recently [21]. We just want to highlight some issues that are related to the chances of automating it or issues that deviates from the mainstream MBT methods.

3.6.1 Advantages of the TTF

We find the TTF particularly appealing for Z users since it keeps all the key elements—operations, test specifications, abstract test cases and others—within the Z notation. Besides, it naturally provides traceability between all these elements by using schema inclusion. Furthermore, users can define new testing tactics that best fit their needs when the standard ones fall short.

3.6.2 The Form of Test Cases in the TTF

As can be seen, within the TTF an abstract test case is a conjunction of equalities between VIS variables and constant values, rather than a sequence of operations leading to the desired state, as is suggested by other approaches [34, 4, 14]. These sequences are useful when the SUT has to be put in a particular state so a test case can be run from there. Since the TTF does not produce such sequences, we have proposed a method that does two things at the same time [12]: (a) refines a TTF abstract test case into an executable program; and (b) sets the initial state of the SUT according to the abstract test case. The only prerequisite is the availability of the source code of the SUT. However, we think that the presentation of this method is outside the scope of this thesis because it concentrates on the "Generation" step of Figure 2. By saying this we want to remark that not representing test cases as sequences of operations does not necessarily mean a weakness of the TTF, it only suggests that other approaches for the execution of test cases should be investigated.

3.6.3 Test Oracles in the TTF

We consider necessary to explain how the TTF deals with test oracles, since this is different from other MBT approaches. A test oracle is a means by which it is possible to determine whether a test case has found an error in the implementation or not. Two of the advantages of the MBT methods is that oracles are rather easy to generate and then, in turn, it allows to automatically determine the presence of an error. The TTF is no exception in this regard, although it deals with test oracles in a rather different way.

Given that the ultimate goal of test oracles is to determine the presence of an error, they are useful once the SUT has been executed on a test case. In other words, it is not mandatory to generate test oracles during the "Generation" step of Figure 2, as long as they are available when the final decision about the presence of an error has to be made. Precisely, test oracles in the TTF can be calculated when that decision is about to be made. Following Figure 2, within the TTF an abstract test case is refined or concretized to become a concrete test case on which the program can be exercised. When the program is executed on such a test case, it produces some concrete output (messages on the screen, return values, exceptions, files, etc.). However, since this output is not necessarily at the level of the specification, it cannot be compared to it. Therefore, the output is abstracted to the level of the specification. In doing so, each output and after-state variable is bound to a constant value as with abstract test cases. Once this process is finished, the abstract test case and the corresponding abstract output are substituted in the specification of the corresponding Z operation. This turns the predicate of the operation into a constant formula, since both abstract test cases and abstract outputs are constants. Therefore, these predicates can be symbolically evaluated. Clearly, if one of this predicates reduces to *true* no error was found—because the output produced by the program corresponds to the expected output for the input that was provided—; but if it reduces to *false* there is an error in the program.

Hence, in the TTF test cases do not include their oracles, as we have seen when the TTF was introduced. In fact, an abstract test case only defines the values for each input and state variable. Given that Fastest generates abstract test cases as prescribed in the TTF and that this thesis concerns only to the "Generation" step, the results of the case studies reported in this thesis do not include test oracles.

3.6.4 State Invariants in the TTF

As we have shown in Section 2.6, we prefer writing state invariants in a separated schema and not in the predicate part of the state schema. In this way, when the TTF is applied to an operation, state invariants are not considered. In other words, state invariants are not analysed in the process of test case generation. This *would* imply that the code implementing it will not be tested.

However, if our approach for recording state invariants is followed there should be a proof for each operation guaranteeing that the latter satisfies the former. This means that each operation was specified in such a way as to verify the state invariant. Therefore, when the operation is analysed by the TTF, it will generate test cases that will test code implementing sufficient functionality to make the program verify the state invariant—because it implements its specification and the specification satisfies the invariant. More formally: if operation O satisfies invariant I—i.e. $I \wedge O \Rightarrow I'$—and we "prove" by testing that program P implements O—i.e. $P \Rightarrow O$—, then we can prove that P satisfies I—i.e. $I \wedge P \Rightarrow I'$. In summary, there is no need in considering state invariants during the "Generation" step as long as the corresponding proof obligations have been discharged.

There is another reason for writing state invariants as proof obligations rather than as part of the state schema. If state invariants are written inside the state schema they tend to produce *implicit preconditions* [35, page 130] [24, Section 7.6]. That is, preconditions that are not explicitly written by the specifier but which are implicit in the specification. Making implicit preconditions explicit requires solving an existential quantification. Given that the first step of the TTF is to define the VIS of each operation, and this, in turn, is defined in terms of the preconditions of the operation, then we need a simple way of getting the preconditions of the operation. Therefore, if there are implicit preconditions it is not possible to guarantee always to find all the preconditions of a given operation—but only its explicit ones. If state invariants are written as proof obligations all the preconditions must be explicit and, thus, easy to find. Hence we advocate for writing state invariants as proof obligations.

Both the TTF and Fastest work fine with either form of writing state invariants. Nevertheless, both work better if our proposal is followed because including the state invariant in an operation makes it much complex but this complexity does not mean better testing, as we have analysed in the previous paragraphs.

4 The Z Specification of the Landing Gear System

4.1 Basic Types

The Z specification uses the following basic types with the meaning given below.

$$LSET ::= forward | left | right$$

$$HPOS ::= down | up$$

$$EVST ::= pressing | idle$$

$$SENS ::= s1 | s2 | s3$$

$$LIGHT ::= on | off$$

The front gear of the aircraft $\approx$ *forward*

The left gear of the aircraft $\approx$ *left*

The right gear of the aircraft $\approx$ *right*

The down position of the handle located in the cockpit $\approx$ *down*

The up position of the handle located in the cockpit $\approx$ *up*

An electro-valve is supplying hydraulic power $\approx$ *pressing*

An electro-valve is not supplying hydraulic power $\approx$ *idle*

s is a sensor identifier $\approx s \in SENS$

A light in the cockpit is on $\approx on$

A light in the cockpit is off $\approx off$

The name "front" is already used in the Z mathematical toolkit so it cannot be used in type $LSET$. It has been replaced by *forward*.

Observe that elements of $SENS$ are regarded as identifiers; they are not the actual sensors which are not represented in this model. In this model when a device reads from its sensors it receives three ordered pairs of the form $(sensor\_id, value)$. Hence, all the devices use the same set of sensor identifiers.

The elements of type ST have different meanings depending on to which variable are bound.

$$ST ::= y | n$$

When ST is used for:

- gears then y means locked and n means maneuvering
- door opening then y means open and n means not open
- door closing then y means closed (and locked) and n means not closed (and not locked)
- the hydraulic circuit (after the general electro-valve) then y means pressurized and n means not pressurized
- shock absorbers then y means ground and n means flight (or relaxed)
- the analogical switch then y means closed and n means open

During the execution of the expected scenarios in normal mode (i.e. outgoing and retraction sequences) the controlling software passes through some internal states which are represented by type $STATE$. The meaning of each state will become clear with the specification of these scenarios. States $d0, \dots, d7$ concern the outgoing sequence, and states $u0, \dots, u7$ the retraction sequence.

$$STATE ::= init | d0 | d1 | d2 | d3 | d4 | d5 | d6 | d7 | u0 | u1 | u2 | u3 | u4 | u5 | u6 | u7$$

In this model the time is discrete and starts from zero.

$$TIME == \mathbb{N}$$

4.2 State of the Controlling Software

The state of the software is given by means of several state variables grouped in some state schemas as described below. The grouping of state variables in different schemas allows operations schemas to use (access) the minimum number of variables they need. This, in turn, make it simpler test case generation.

The first state schema groups the variables that set the state of the gears of the aircraft. It includes the following variables:

The gear g is locked or not locked in extended position $\approx gExt\,g$

The gear g is locked or not locked in retracted position $\approx gRec\,g$

The valid sensors sensing whether gear g is locked or not locked in extended position $\approx sGExt\,g$

The valid sensors sensing whether gear g is locked or not locked in retracted position $\approx sGRec\,g$

Since the type of *gEtx* and *gRec* is defined in terms of type *ST* and given that these variables represents properties of the gears, then the *y* value of *ST* means locked while the *n* value means maneuvering. Recall the list of interpretations for *ST* given in page 21.

According to this, if, for instance, at time *t* the sensors sensing whether the front gear is locked or not locked in extended position deliver the following values:

$$s1 \mapsto y, s2 \mapsto n, s3 \mapsto y$$

then at time *t* the value of *sGExtfront* will be $\{s1, s3\}$. The same applies to *sGRec* and in general to all the variables that represent the set of valid sensors of the various devices of the LGS.

GearsExtending
$gExt : LSET \rightarrow ST$
$sGExt : LSET \rightarrow \mathbb{F}\, SENS$

GearsRetracting
$gRec : LSET \rightarrow ST$
$sGRec : LSET \rightarrow \mathbb{F}\, SENS$

$$Gears == GearsExtending \wedge GearsRetracting$$

gExt could be defined also as of type $\mathbb{F}\, LSET$, which would be closer to the Z style. In this way, $g \in gExt$ if and only if gear *g* is locked in extended position. However, in this particular model it seems easier (and clearer) to define *gExt* as a function. The same applies to *gRec*.

State schema *Doors* plays the same role for doors than *Gears* for gears with a similar set of state variables. Therefore, only its designations are given below. Recall the list of interpretations for *ST* given in page 21.

DoorsOpening
$dOp : LSET \rightarrow ST$
$sDOp : LSET \rightarrow \mathbb{F}\, SENS$

DoorsClosing
$dCl : LSET \rightarrow ST$
$sDCl : LSET \rightarrow \mathbb{F}\, SENS$

$$Doors == DoorsOpening \wedge DoorsClosing$$

The door *d* is in open or not open position $\approx dOp\, d$

The door *d* is locked or not locked in closed position $\approx dCl\, d$

The valid sensors sensing whether door *d* is in open or not open position $\approx sDOp\, d$

The valid sensors sensing whether door *d* is locked or not locked in closed position $\approx sDCl\, d$

The following schema groups the variables for the shock absorbers. They are quite similar to those of the preceding schemas. So, only its designations are given. Recall the list of interpretations for *ST* given in page 21.

ShockAbsorbers

$sa : LSET \rightarrow ST$
$sSA : LSET \rightarrow \mathbb{F}\, SENS$

The shock absorber s is in ground or in flight position $\approx sa\, s$

The valid sensors sensing whether shock absorber s is in ground or in flight position $\approx sSA\, s$

Given that there is only one hydraulic circuit dedicated to the LGS it is not necessary to define a function to record its state; one simple variable is enough. Then, *hc* stores the state of the hydraulic circuit. Recall the list of interpretations for *ST* given in page 21.

HydraulicCircuit

$hc : ST$
$sHC : \mathbb{F}\, SENS$

The hydraulic circuit is pressurized or not pressurized $\approx hc$

The valid sensors sensing whether the hydraulic circuit is pressurized or not pressurized $\approx sHC$

The following schema look like *HydraulicCircuit* so only its designations are given.

AnalogicalSwitch

$as : ST$
$sAS : \mathbb{F}\, SENS$

The analogical switch is open or closed $\approx as$

The valid sensors sensing whether the analogical switch is open or closed $\approx sAS$

Handle groups the state variables regarding the handle in the cockpit. It includes the position of the handle (*hPos*) and two time marks necessary for the timing constraints.

Handle

$hPos : HPOS$
$lHPCh, l20 : TIME$

The handle is up or down $\approx$ *hPos*

Last time the handle position changed $\approx$ *lHPCh*

Last time that the handle position has not changed for 20 seconds $\approx$ *l*20

Now follows some schemas grouping state variables of the different electro-valves. Each schema includes a variable to record the state of the corresponding electro-valve as well as one variable to store the last time the electro-valve was stimulated; only the general electro-valve has an extra variable to store the last time an electro-valve was stopped.

GeneralEV
$stGEV, spGEV : TIME$
$gEV : EVST$

The general electro-valve is pressing (i.e. is providing hydraulic power) or is idle (i.e. is not providing hydraulic power) $\approx$ *gEV*

Stimulation of the general electro-valve was started at time $\approx$ *stEV*

Stimulation of the general electro-valve was stopped at time $\approx$ *spEV*

DoorOpeningEV
$doEV : EVST$
$stDOEV : TIME$

The electro-valve related to door opening is pressing or is idle $\approx$ *doEV*

Stimulation of the electro-valve related to door opening was started at time $\approx$ *stDOEV*

DoorClosingEV
$dcEV : EVST$
$stDCEV : TIME$

The electro-valve related to door closing is pressing or is idle $\approx$ *dcEV*

Stimulation of the electro-valve related to door closing was started at time $\approx$ *stDCEV*

GearsExtendingEV
$geEV : EVST$
$stGEEV : TIME$

> The electro-valve related to gear extension is pressing or is idle $\approx$ *geEV*
>
> Stimulation of the electro-valve related to gear extension was started at time $\approx$ *stGEEV*

GearsRetractingEV
grEV : *EVST*
stGREV : *TIME*

> The electro-valve related to gear retraction is pressing or is idle $\approx$ *grEV*
>
> Stimulation of the electro-valve related to gear retraction was started at time $\approx$ *stGREV*

There are two more variables to record time marks.

EVst
stEV : *TIME*

EVsp
spEV : *TIME*

> Stimulation of an electro-valve was started at time $\approx$ *stEV*
>
> Stimulation of an electro-valve was stopped at time $\approx$ *spEV*

Every time the general electro-valve is stimulated two variables will be updated:

- *gEV* will be set to *pressing*; and
- *stGEV* will be set to the current time

Every time the general electro-valve is stopped two variables will be updated:

- *gEV* will be set to *idle*; and
- *spGEV* will be set to the current time

Every time an electro valve is stimulated three variables will be updated. For example, if the door opening electro-valve is stimulated:

- *doEV* will be set to *pressing*;
- *stDOEV* will be set to the current time; and
- *stEV* (from schema *EVst*) will be set to the current time

Every time an electro valve is stopped two variables will be updated. For example, if the door opening electro-valve is stimulated:

- *doEV* will be set to *idle*;
- *spEV* (from schema *EVsp*) will be set to the current time

In this way, it is possible to know:

- The elapsed time between two consecutive stimulations of any two electro-valves, through variable *stEV*.
- The elapsed time between two consecutive orders to stop the stimulation of any two electro-valves, through variable *spEV*.
- The elapsed time between two consecutive contrary orders of any two electro-valves, through variables *stEV*, *stDOEV*, *stDCEV*, *stGEEV* and *stGREV*.
- The last time the general electro-valve was stimulated or stopped, through variables *stGEV* and *spGEV*.

The state of the cockpit is represented by three simple variables each of them corresponding to the three lights that inform the pilot about the state of the LGS. The "landing gear system failure" light (*lgsfl*) is used as a synonym for "LGS mode of operation". That is when $lgsfl = on$ the LGS has failed and the pilot can activate the emergency hydraulic circuit; when $lgsfl = off$ the LGS is operating normally.

CockpitA

$lgsfl : LIGHT$

CockpitN

$gldl, gml : LIGHT$

$Cockpit == CockpitA \wedge CockpitN$

The "landing gear system failure" light $\approx lgsfl$

The "gears are locked down" light $\approx gldl$

The "gears maneuvering" light $\approx gml$

When the two main scenarios in normal mode (i.e. the outgoing and retraction sequences) are executed the software goes through a series of internal states. The following variable records that state.

StateCounter

$st : STATE$

The software internal state during either the outgoing or retraction sequence $\approx st$

Since there are several timing restrictions that the software must met, the model keeps track of time advance by means of variable *now*. This variable is incremented by 1 ms.

Time

$now : TIME$

The current time $\approx$ *now*

4.3 Initial states

The initial state of the system represents a "healthy" aircraft on ground. That is, its gears are locked down (which in turn means that they are not retracted), the doors are closed, all the sensors are operating correctly, etc. This state is set by setting the variables in each and every one of the state schemas defined above.

GearsInit
Gears

$\operatorname{ran} gExt = \{y\}$
$\operatorname{ran} gRec = \{n\}$
$\operatorname{ran} sGExt = \operatorname{ran} sGRec = \{SENS\}$

If the aircraft is on ground the doors of the LGS are closed.

DoorsInit
Doors

$\operatorname{ran} dOp = \{n\}$
$\operatorname{ran} dCl = \{y\}$
$\operatorname{ran} sDOp = \operatorname{ran} sDCl = \{SENS\}$

Obviously the shock absorbers are on ground.

ShockAbsorbersInit
ShockAbsorbers

$\operatorname{ran} sa = \{y\}$
$\operatorname{ran} sSA = \{SENS\}$

The hydraulic circuit is not pressurized.

HydraulicCircuitInit
HydraulicCircuit

$hc = n$
$sHC = SENS$

The analogical switch is open.

AnalogicalSwitchInit
AnalogicalSwitch

$as = n$
$sAS = SENS$

The handle is down so it is consistent with the state of the gears.

HandleInit

$$Handle$$

$$hPos = down$$
$$lHPCh = l20 = 0$$

All the electro-valves are not providing hydraulic power.

GeneralEVInit

$$EVst$$
$$EVsp$$
$$GeneralEV$$

$$gEV = idle$$
$$stEV = spEV = stGEV = spGEV = 0$$

DoorOpeningEVInit

$$DoorOpeningEV$$

$$doEV = idle$$
$$stDOEV = 0$$

DoorClosingEVInit

$$DoorClosingEV$$

$$dcEV = idle$$
$$stDCEV = 0$$

GearsExtendingEVInit

$$GearsExtendingEV$$

$$geEV = idle$$
$$stGEEV = 0$$

GearsRetractingEVInit

$$GearsRetractingEV$$

$$grEV = idle$$
$$stGREV = 0$$

The lights in the cockpit reflects the state of the gears and the healthy of the system.

```
┌─ CockpitInit ────────────────────────
│ Cockpit
├──────────
│ lgsfl = off
│ gldl = on
│ gml = off
└──────────────────────────────────────
```

The internal state counter is in its initial state.

```
┌─ StateCounterInit ───────────────────
│ StateCounter
├──────────
│ st = init
└──────────────────────────────────────
```

The LGS time starts at zero.

```
┌─ TimeInit ───────────────────────────
│ Time
├──────────
│ now = 0
└──────────────────────────────────────
```

4.4 Assumptions and Limitations of the Z Specification

The Z specification presented here (and the complete model described in [10]) is based on the following assumptions and limitations:

- Each Z operation is atomic and takes no time to be executed.
- If at a given point in time there is more than one operation enabled (i.e. their preconditions are true), the system nondeterministically executes one of them.
- Only one operation is executed at any given time.
- All operations are executed according to a weak fairness formula [25]. Z should be extended as Evans suggests to be able to write these formulas [16].
- The software stops working when the red light in the cockpit is turned on ($lgsfl = on$). Then, operations do not include the precondition $lgsfl = off$ because when this is no longer true the software is not working.
- The system measures the time in milliseconds; the time is considered to be discrete.
- The specification considers just one computing module [6, Sect. 2.3]. That is all the outputs produced by the software are produced by only one computing module.
- The sentence:

  > two contrary orders (closure / opening doors, extension / retraction gears) must be separated by at least 100ms.

  is interpreted as follows:

  > the start of stimulation of the electro-valves corresponding to devices that execute contrary orders must be separated by at least 100ms.

4.5 Operations Concerning Sensor Readings and its Anomalies

In this section the operations describing how the values read by sensors are stored in the system are formalized. These operations include the specification of the conditions under which an anomaly concerning the validity of sensors is detected.

Given that every device (gears, doors, shock absorbers, etc.) reads (simultaneously) from three sensors, the following operations take an input variable $v?$ of type $SENS \rightarrow ST$. Its interpretation is simple: if $s : SENS$ then $v?s$ is the value delivered by sensor s at that moment. In this sense, $v?$ can be seen as the result of an election where each sensor votes for one of two possible candidates.

There are two key operations regarding $v?$ when a device of the LGS reads its sensors. The first one is to find the set of valid sensors. This set is formed by the majority who won the election. The y value won the election if $\#(v \rhd \{y\}) > \#(v \rhd \{n\})$, otherwise the winer is n. Note that if $\#v? = 2$ and each sensor reads a different value, n wins but this is irrelevant as is going to be shown below. The next function, *valid*, calculates the set of valid sensors by taking the domain of $v \rhd \{y\}$ or $v \rhd \{n\}$ depending on who won the election.

$$
\begin{array}{l}
valid : (SENS \nrightarrow ST) \rightarrow \mathbb{F}\, SENS \\
\hline
\forall v : SENS \nrightarrow ST \bullet \\
\quad valid\, v = \\
\qquad \textbf{if}\ \#(v \rhd \{y\}) > \#(v \rhd \{n\}) \\
\qquad \textbf{then}\ \mathrm{dom}(v \rhd \{y\}) \\
\qquad \textbf{else}\ \mathrm{dom}(v \rhd \{n\})
\end{array}
$$

Note that, although $v?$ is a total function, *valid* waits a partial function. This is so because when a sensor is invalidated the following readings consider only the two remaining sensors. Therefore, *valid* is called with $v?$ restricted to the two active sensors and this is a partial function.

The second operation regarding $v?$ is to determine the net value read by the three sensors. According to [6], the net value is the value read by the majority. *value*, returns the net value in a similar way as *valid* returns the set of valid sensors.

$$
\begin{array}{l}
value : (SENS \nrightarrow ST) \rightarrow ST \\
\hline
\forall v : SENS \nrightarrow ST \bullet \\
\quad value\, v = \\
\qquad \textbf{if}\ \#(v \rhd \{y\}) > \#(v \rhd \{n\})\ \textbf{then}\ y\ \textbf{else}\ n
\end{array}
$$

Each of the following Z operations describe how a device of the LGS reads from its sensors and what the software does with these values. All these operations share a common structure:

- A first schema, whose name ends in *N*, describing the case when the three sensors return the same value. *N* here suggests normal functioning.
- A second schema, whose name ends in *Ds*, describing the case when a sensor returns a different value for the first time (and so it is discharged for ever). *Ds* here suggests degrades.
- A third schema, whose name ends in *Dd*, describing the case when two sensors are working correctly. *Dd* here suggests degraded.
- A fourth schema, whose name ends in *A*, describing the case when the two remaining sensors differ in their readings for the first time (and so a system anomaly is detected). *A* here suggests anomaly.

- A fifth schema as the disjunction of the four previous schemas that describes the full operation.

Since all the operations share a common structure and the predicates in them are very similar to each other, only the first operation is explained in detail.

4.5.1 Gears in Extended Position

The next five schemas describe the system reading the sensors that determine whether the gears are locked or not locked in extended position. Each schema receives the gear or landing set to which the reading applies, $g?$, and $v?$ (as mentioned above). *ReadGearsExtendingN* has only one preconditions: all the three sensors are valid with respect to reading $v?$. In other words this device is working in normal mode. Then, the state of the corresponding gear ($g?$) is updated according to the net value ($value(v?)$) read by the sensors. The set of valid sensors of these devices ($sGExt$) remain unchanged (recall that the initial value for all the variables representing valid sets of sensors is $SENS$, see Sect. 4.3).

ReadGearsExtendingN

$\Delta GearsExtending;\ \Xi CockpitA$
$g? : LSET$
$v? : SENS \rightarrow ST$

$valid(v?) = SENS$
$gExt' = gExt \oplus \{g? \mapsto value(v?)\}$
$sGExt' = sGExt$

In *ReadGearsExtendingDs* the device for g starts to work in degraded mode because one of the sensors reads a different value with respect to the other two ($valid(v?) \subset SENS$), when previously all of them were working properly ($sGExt\,g? = SENS$). Therefore, the state of the corresponding gear ($g?$) is updated according to the net value ($value(v?)$) read by the sensors and the set of valid sensors for g is also updated.

ReadGearsExtendingDs

$\Delta GearsExtending;\ \Xi CockpitA$
$g? : LSET$
$v? : SENS \rightarrow ST$

$sGExt\,g? = SENS$
$valid(v?) \subset SENS$
$gExt' = gExt \oplus \{g? \mapsto value(v?)\}$
$sGExt' = sGExt \oplus \{g? \mapsto valid(v?)\}$

Now that there are just two valid sensors the net value must be calculated from them. So *ReadGearsExtendingDd* is applied if the set of valid sensors is strictly included in $SENS$ and it is exactly the same set recorded by the system: $valid(sGExt\,g? \lhd v?) = sGExt\,g?$. In other words, since one of the sensors is no longer considered (although it keeps sending its readings to the system), the set of valid sensors is calculated from the set of sensors that are currently considered valid: $sGExt\,g?$. So *valid* is called with $v?$ restricted to the set of valid sensors ($sGExt\,g? \lhd v?$) to see if this is still the same set. The same applies to the way the net value is calculated: $value(sGExt\,g? \lhd v?)$.

$$
\begin{array}{l}
\text{\_\_}\mathit{ReadGearsExtendingDd}\text{\_\_\_\_\_\_\_\_} \\
\Delta \mathit{GearsExtending};\ \Xi \mathit{CockpitA} \\
g? : \mathit{LSET} \\
v? : \mathit{SENS} \rightarrow \mathit{ST} \\
\hline
\mathit{sGExt}\, g? \subset \mathit{SENS} \\
\mathit{valid}(\mathit{sGExt}\, g? \lhd v?) = \mathit{sGExt}\, g? \\
\mathit{gExt}' = \mathit{gExt} \oplus \{g? \mapsto \mathit{value}(\mathit{sGExt}\, g? \lhd v?)\} \\
\mathit{sGExt}' = \mathit{sGExt}
\end{array}
$$

If one of the two working sensors fails ($\mathit{valid}(\mathit{sGExt}\, g? \lhd v?) \neq \mathit{sGExt}\, g?$) then the system moves to a failed state by turning on the read light in the cockpit: $\mathit{lgsfl}' = \mathit{on}$. Remember that it was assumed that from this moment the software is not working anymore. By the way, note that in this case *valid* will arbitrarily return the set $\{s_i\}$ where $v?\, s_i = n$. However, this is irrelevant because this set will be different from $\mathit{sGExt}\, g?$ since it has two elements.

$$
\begin{array}{l}
\text{\_\_}\mathit{ReadGearsExtendingA}\text{\_\_\_\_\_\_\_\_} \\
\Xi \mathit{GearsExtending};\ \Delta \mathit{CockpitA} \\
g? : \mathit{LSET} \\
v? : \mathit{SENS} \rightarrow \mathit{ST} \\
\hline
\mathit{sGExt}\, g? \subset \mathit{SENS} \\
\mathit{valid}(\mathit{sGExt}\, g? \lhd v?) \neq \mathit{sGExt}\, g? \\
\mathit{lgsfl}' = \mathit{on}
\end{array}
$$

ReadGearsExtending is simply the disjunction of the preceding four schemas, thus defining the full operation of reading the sensors informing about the lock of gears in extended position.

$$
\begin{array}{l}
\mathit{ReadGearsExtending} == \\
\quad \mathit{ReadGearsExtendingN} \\
\quad \lor \mathit{ReadGearsExtendingDs} \lor \mathit{ReadGearsExtendingDd} \lor \mathit{ReadGearsExtendingA}
\end{array}
$$

As has been said above, the next schemas share the same structure and have similar predicates so no more informal explanations will be given.

4.5.2 Gears in Retracted Position

> The system reads the sensors indicating whether the gears are locked or not locked in retracted position $\approx$ *ReadGearsRetracting*

$$
\begin{array}{l}
\text{\_\_}\mathit{ReadGearsRetractingN}\text{\_\_\_\_\_\_\_\_} \\
\Delta \mathit{GearsRetracting};\ \Xi \mathit{CockpitA} \\
g? : \mathit{LSET} \\
v? : \mathit{SENS} \rightarrow \mathit{ST} \\
\hline
\mathit{valid}(v?) = \mathit{SENS} \\
\mathit{gRec}' = \mathit{gRec} \oplus \{g? \mapsto \mathit{value}(v?)\} \\
\mathit{sGRec}' = \mathit{sGRec}
\end{array}
$$

ReadGearsRetractingDs

$\Delta GearsRetracting; \Xi CockpitA$
$g? : LSET$
$v? : SENS \rightarrow ST$

$sGRec\, g? = SENS$
$valid(v?) \subset SENS$
$gRec' = gRec \oplus \{g? \mapsto value(v?)\}$
$sGRec' = sGRec \oplus \{g? \mapsto valid(v?)\}$

ReadGearsRetractingDd

$\Delta GearsRetracting; \Xi CockpitA$
$g? : LSET$
$v? : SENS \rightarrow ST$

$sGRec\, g? \subset SENS$
$valid(sGRec\, g? \lhd v?) = sGRec\, g?$
$gRec' = gRec \oplus \{g? \mapsto value(sGRec\, g? \lhd v?)\}$
$sGRec' = sGRec$

ReadGearsRetractingA

$\Xi GearsRetracting; \Delta CockpitA$
$g? : LSET$
$v? : SENS \rightarrow ST$

$sGRec\, g? \subset SENS$
$valid(sGRec\, g? \lhd v?) \neq sGRec\, g?$
$lgsfl' = on$

$ReadGearsRetracting ==$
$\quad ReadGearsRetractingN$
$\quad \lor ReadGearsRetractingDs \lor ReadGearsRetractingDd \lor ReadGearsRetractingA$

4.5.3 Shock Absorbers

The system reads the sensors indicating whether the shock absorbers are on ground or in flight $\approx$ *ReadShockAbsorbers*

ReadShockAbsorbersN

$\Delta ShockAbsorbers; \Xi CockpitA$
$g? : LSET$
$v? : SENS \rightarrow ST$

$valid(v?) = SENS$
$sa' = sa \oplus \{g? \mapsto value(v?)\}$
$sSA' = sSA$

$$
\begin{array}{l}
\textit{ReadShockAbsorbersDs} \\
\hline
\Delta ShockAbsorbers;\ \Xi CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sSA\ g? = SENS \\
valid(v?) \subset SENS \\
sa' = sa \oplus \{g? \mapsto value(v?)\} \\
sSA' = sSA \oplus \{g? \mapsto valid(v?)\} \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textit{ReadShockAbsorbersDd} \\
\hline
\Delta ShockAbsorbers;\ \Xi CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sSA\ g? \subset SENS \\
valid(sSA\ g? \lhd v?) = sSA\ g? \\
sa' = sa \oplus \{g? \mapsto value(sSA\ g? \lhd v?)\} \\
sSA' = sSA \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textit{ReadShockAbsorbersA} \\
\hline
\Xi ShockAbsorbers;\ \Delta CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sSA\ g? \subset SENS \\
valid(sSA\ g? \lhd v?) \neq sSA\ g? \\
lgsfl' = on \\
\hline
\end{array}
$$

$$
\begin{array}{l}
ReadShockAbsorbers == \\
\quad ReadShockAbsorbersN \\
\quad \lor ReadShockAbsorbersDs \lor ReadShockAbsorbersDd \lor ReadShockAbsorbersA
\end{array}
$$

4.5.4 Doors Open

The system reads the sensors indicating whether the doors are in open or not open position $\approx$ *ReadDoorsOpening*

$$
\begin{array}{l}
\textit{ReadDoorsOpeningN} \\
\hline
\Delta DoorsOpening;\ \Xi CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
valid(v?) = SENS \\
dOp' = dOp \oplus \{g? \mapsto value(v?)\} \\
sDOp' = sDOp \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{ReadDoorsOpeningDs}} \\
\Delta DoorsOpening;\ \Xi CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sDOp\, g? = SENS \\
valid(v?) \subset SENS \\
dOp' = dOp \oplus \{g? \mapsto value(v?)\} \\
sDOp' = sDOp \oplus \{g? \mapsto valid(v?)\}
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{ReadDoorsOpeningDd}} \\
\Delta DoorsOpening;\ \Xi CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sDOp\, g? \subset SENS \\
valid(sDOp\, g? \lhd v?) = sDOp\, g? \\
dOp' = dOp \oplus \{g? \mapsto value(sDOp\, g? \lhd v?)\} \\
sDOp' = sDOp
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{ReadDoorsOpeningA}} \\
\Xi DoorsOpening;\ \Delta CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sDOp\, g? \subset SENS \\
valid(sDOp\, g? \lhd v?) \neq sDOp\, g? \\
lgsfl' = on
\end{array}
$$

$$
\begin{array}{l}
ReadDoorsOpening == \\
\quad ReadDoorsOpeningN \\
\quad \lor ReadDoorsOpeningDs \lor ReadDoorsOpeningDd \lor ReadDoorsOpeningA
\end{array}
$$

4.5.5 Doors Closed

The system reads the sensors indicating whether the doors are locked or not locked in closed position $\approx$ *ReadDoorsClosing*

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{ReadDoorsClosingN}} \\
\Delta DoorsClosing;\ \Xi CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
valid(v?) = SENS \\
dCl' = dCl \oplus \{g? \mapsto value(v?)\} \\
sDCl' = sDCl
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{ReadDoorsClosingDs}} \\
\Delta DoorsClosing;\ \Xi CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sDCl\, g? = SENS \\
valid(v?) \subset SENS \\
dCl' = dCl \oplus \{g? \mapsto value(v?)\} \\
sDCl' = sDCl \oplus \{g? \mapsto valid(v?)\}
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{ReadDoorsClosingDd}} \\
\Delta DoorsClosing;\ \Xi CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sDCl\, g? \subset SENS \\
valid(sDCl\, g? \lhd v?) = sDCl\, g? \\
dCl' = dCl \oplus \{g? \mapsto value(sDCl\, g? \lhd v?)\} \\
sDCl' = sDCl
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{ReadDoorsClosingA}} \\
\Xi DoorsClosing;\ \Delta CockpitA \\
g? : LSET \\
v? : SENS \rightarrow ST \\
\hline
sDCl\, g? \subset SENS \\
valid(sDCl\, g? \lhd v?) \neq sDCl\, g? \\
lgsfl' = on
\end{array}
$$

$$
\begin{array}{l}
ReadDoorsClosing == \\
\quad ReadDoorsClosingN \\
\quad \lor ReadDoorsClosingDs \lor ReadDoorsClosingDd \lor ReadDoorsClosingA
\end{array}
$$

4.5.6 Hydraulic Circuit

The remaining two operations describe the reading of single devices (hydraulic circuit and analogical switch). Then, the parameter $g?$ used in the previous schemas is no longer needed. Furthermore, the predicates in these operations are simpler because the variables recording the state of the device and the set of valid sensors are plain variables (i.e. they are not functions as in the previous operations).

> The system reads the sensors indicating whether the hydraulic circuit is pressurized or not pressurized $\approx$ *ReadHydraulicCircuit*

$$
\begin{array}{|l}
\hline
\multicolumn{1}{l}{\mathit{ReadHydraulicCircuitN}} \\
\Delta \mathit{HydraulicCircuit};\ \Xi \mathit{CockpitA} \\
v? : \mathit{SENS} \rightarrow \mathit{ST} \\
\hline
\mathit{valid}(v?) = \mathit{SENS} \\
\mathit{hc}' = \mathit{value}(v?) \\
\mathit{sHC}' = \mathit{sHC} \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\multicolumn{1}{l}{\mathit{ReadHydraulicCircuitDs}} \\
\Delta \mathit{HydraulicCircuit};\ \Xi \mathit{CockpitA} \\
v? : \mathit{SENS} \rightarrow \mathit{ST} \\
\hline
\mathit{sHC} = \mathit{SENS} \\
\mathit{valid}(v?) \subset \mathit{SENS} \\
\mathit{hc}' = \mathit{value}(v?) \\
\mathit{sHC}' = \mathit{valid}(v?) \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\multicolumn{1}{l}{\mathit{ReadHydraulicCircuitDd}} \\
\Delta \mathit{HydraulicCircuit};\ \Xi \mathit{CockpitA} \\
v? : \mathit{SENS} \rightarrow \mathit{ST} \\
\hline
\mathit{sHC} \subset \mathit{SENS} \\
\mathit{valid}(\mathit{sHC} \lhd v?) = \mathit{sHC} \\
\mathit{hc}' = \mathit{value}(\mathit{sHC} \lhd v?) \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\multicolumn{1}{l}{\mathit{ReadHydraulicCircuitA}} \\
\Xi \mathit{HydraulicCircuit};\ \Delta \mathit{CockpitA} \\
v? : \mathit{SENS} \rightarrow \mathit{ST} \\
\hline
\mathit{sHC} \subset \mathit{SENS} \\
\mathit{valid}(\mathit{sHC} \lhd v?) \neq \mathit{sHC} \\
\mathit{lgsfl}' = \mathit{on} \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\mathit{ReadHydraulicCircuit} == \\
\quad \mathit{ReadHydraulicCircuitN} \\
\quad \lor \mathit{ReadHydraulicCircuitDs} \lor \mathit{ReadHydraulicCircuitDd} \lor \mathit{ReadHydraulicCircuitA}
\end{array}
$$

4.5.7 Analogical Switch

The system reads the sensors indicating whether the analogical switch (between the digital part and the general electro-valve) is closed or open $\approx$ *ReadAnalogicalSwitch*

ReadAnalogicalSwitchN

$$\Delta AnalogicalSwitch;\ \Xi CockpitA$$
$$v? : SENS \rightarrow ST$$

$$valid(v?) = SENS$$
$$as' = value(v?)$$
$$sAS' = sAS$$

ReadAnalogicalSwitchDs

$$\Delta AnalogicalSwitch;\ \Xi CockpitA$$
$$v? : SENS \rightarrow ST$$

$$sAS = SENS$$
$$valid(v?) \subset SENS$$
$$as' = value(v?)$$
$$sAS' = valid(v?)$$

ReadAnalogicalSwitchDd

$$\Delta AnalogicalSwitch;\ \Xi CockpitA$$
$$v? : SENS \rightarrow ST$$

$$sAS \subset SENS$$
$$valid(sAS \lhd v?) = sAS$$
$$as' = value(sAS \lhd v?)$$
$$sAS' = sAS$$

ReadAnalogicalSwitchA

$$\Xi AnalogicalSwitch;\ \Delta CockpitA$$
$$v? : SENS \rightarrow ST$$

$$sAS \subset SENS$$
$$valid(sAS \lhd v?) \neq sAS$$
$$lgsfl' = on$$

$$ReadAnalogicalSwitch == ReadAnalogicalSwitchN \lor ReadAnalogicalSwitchDs \lor ReadAnalogicalSwitchDd \lor ReadAnalogicalSwitchA$$

4.6 Operations Concerning Normal Mode

This section includes the Z operations that describe the interaction with the cockpit, the two main scenarios in normal mode (i.e. the outgoing and retracting sequences) and the counter orders that may be given during the execution of these scenarios.

The first operation represents the pilot moving the handle up or down. Each of these operations enables the corresponding main scenario.

Each of the two main scenarios is organized in eight schemas, each representing one of the steps or elementary actions described in [6, pages 14 and 15]. The third step of the retraction sequence is decomposed in two schemas.

The counter orders are decomposed in seven schemas each, because it has been considered that if the counter order arrives before the first step of the main scenario has been executed there is nothing to revert.

4.6.1 Interaction with the cockpit

In the initial state of the LGS the internal state counter, *st*, is equal to *init*; the handle, *hPos*, is in the *down* position; and the gears are locked in extended position (see Sect. 4.3). Then, the only thing the pilot can do is to move the handle to the *up* position. In this moment *st* is set to the first internal state, $u0$, of the retracting sequence. This new value for *st* enables the first step, $Up1$, of the retracting sequence. Besides, a time mark is taken when the pilot moves the handle to record the time of its last change: $lHPCh' = now$. All this is specified in *ChangeHandleDownUp*.

ChangeHandleDownUp

$\Delta Handle$; $\Delta StateCounter$; $\Xi Time$

$hPos = down$
$hPos' = up$
$lHPCh' = now$
$st' = u0$
$l20' = l20$

The symmetric operation is specified in *ChangeHandleUpDown*.

ChangeHandleUpDown

$\Delta Handle$; $\Delta StateCounter$; $\Xi Time$

$hPos = up$
$hPos' = down$
$lHPCh' = now$
$st' = d0$
$l20' = l20$

The complete specification is as follows:

$$ChangeHandle == ChangeHandleDownUp \lor ChangeHandleUpDown$$

Since anomalies are calculated in sections 4.5 and 4.8, this section includes only the interaction with the cockpit in normal mode. These interactions include turning on and off the lights indicating the position of gears. The first operation describes the conditions to turn on and off the green light which, when on, indicates that all the gears are locked down. The three gears are locked in extended position when y is the only element of the range of *gExt*. In other words, when *gExt* is applied to any element of *LSET* the result is y, which means that each and every gear is locked in the extended position.

GearsLockedDownOn

$$\Delta CockpitN; \Xi GearsExtending$$

$$\operatorname{ran} gExt = \{y\}$$
$$gldl' = on$$

GearsLockedDownOff

$$\Delta CockpitN; \Xi GearsExtending$$

$$\operatorname{ran} gExt \neq \{y\}$$
$$gldl' = off$$

$$GearsLockedDown == GearsLockedDownOn \lor GearsLockedDownOff$$

The second operation describes the conditions to turn on and off the orange light which, when on, indicates that gears are maneuvering. Note that "gears are not maneuvering" is formalized as:

$$(\operatorname{ran} gExt = \{y\} \lor \operatorname{ran} gRec = \{y\}) \land \operatorname{ran} dCl = \{y\}$$

so the negation of this predicate means "gears are maneuvering":

$$\neg ((\operatorname{ran} gExt = \{y\} \lor \operatorname{ran} gRec = \{y\}) \land \operatorname{ran} dCl = \{y\})$$
$$\equiv (\operatorname{ran} gExt \neq \{y\} \land \operatorname{ran} gRec \neq \{y\}) \lor \operatorname{ran} dCl \neq \{y\}$$

Therefore the operation is defined as follows:

GearsManeuveringOn

$$\Delta CockpitN; \Xi Gears; \Xi DoorsClosing$$

$$(\operatorname{ran} gExt \neq \{y\} \land \operatorname{ran} gRec \neq \{y\}) \lor \operatorname{ran} dCl \neq \{y\}$$
$$gml' = on$$

GearsManeuveringOff

$$\Delta CockpitN; \Xi Gears; \Xi DoorsClosing$$

$$(\operatorname{ran} gExt = \{y\} \lor \operatorname{ran} gRec = \{y\}) \land \operatorname{ran} dCl = \{y\}$$
$$gml' = off$$

$$GearsManeuvering == GearsManeuveringOn \lor GearsManeuveringOff$$

4.6.2 Retraction sequence

When the handle is moved to the *up* position the internal state of the software, *st*, is set to $u0$ which enables the first step of the retraction sequence formalized as follows:

Up1Ok

$\Delta GeneralEV$; $\Delta EVst$; $\Delta StateCounter$
$\Xi Time$; $\Xi Handle$

$st = u0$
$hPos = up$
$200 \leq now - stEV \lor stEV = 0$
$gEV' = pressing$
$stGEV' = now$
$stEV' = now$
$st' = u1$
$spGEV' = spGEV$

As can be seen, *Up*1 has three preconditions:

- The internal state is *u*0;
- The handle is in the *up* position; and
- The last time an electro-valve was stimulated was more than 200 ms before the current time or it is the first time an electro-valve is stimulated.

In turn its postconditions are simply:

- The general electro-valve is stimulated;
- The current time is saved in a state variable because later will be necessary to see if timing constraints regarding electro-valve stimulation are met; and
- The internal state is set to *u*1.

A full specification of *Up*1 must say what the software should do if $st = u0 \land hPos = up \land (200 \leq now - stEV \lor stEV = 0)$ is not true. Although this seems odd, the software may fail and try to call the routine implementing *Up*1 when the system is in an unexpected state. In this case the system must remain in the same state. Then, the following schema is defined:

Up1E

$\Xi GeneralEV$; $\Xi EVst$; $\Xi StateCounter$; $\Xi Time$; $\Xi Handle$

$\neg (st = u0 \land hPos = up \land (200 \leq now - stEV \lor stEV = 0))$

Hence, the full operation is:

$$Up1 == Up1Ok \lor Up1E$$

In *Up*2 the door opening electro-valve is stimulated, then it is necessary to check whether enough time (100 ms) has elapsed since the last stimulation of the opposite electro-valve (door closing) unless the latter was never stimulated.

Up2Ok

$\Delta DoorOpeningEV$; $\Delta EVst$; $\Delta StateCounter$
$\Xi DoorClosingEV$; $\Xi Time$; $\Xi Handle$

$st = u1$
$hPos = up$
$200 \leq now - stEV$
$100 \leq now - stDCEV \lor stDCEV = 0$
$doEV' = pressing$
$stDOEV' = now$
$stEV' = now$
$st' = u2$

Besides, after *Up*1 has finished it is possible to receive a contrary order before any of the remaining step starts. Therefore, if a contrary order arrives between the moment that *Up*1 has just finished and right before *Up*2 starts, then the latter will stop the retraction sequence and will set the state of the system in such a way as to start the outgoing sequence at the right point. This is formalized as follows:

UpDown2

$\Delta StateCounter$
$\Xi DoorOpeningEV$; $\Xi EVst$; $\Xi DoorClosingEV$; $\Xi Time$; $\Xi Handle$

$st = u1$
$hPos = down$
$st' = d1$

As can be seen *UpDown*2 has two preconditions:

- The handle is *down*; and
- *st* is in *u*1

In this way, every time *Up*1 "finishes", *Up*2*Ok* and *UpDown*2 can be enabled. However, in a given execution there will be just one of them enabled depending on whether the pilot has moved the handle or not in the meanwhile. In effect, if the pilot does not move the handle to the *down* position right after *Up*1, *Up*2*Ok* will be enabled, but if he or she moves the handle, then *UpDown*2 will be enabled. Note that when the handle is moved none of the *Down* schemas (see Sect. 4.6.3) become automatically enabled because they have as a precondition $st = d_i$ with $d_i \in \{d0, \dots, d7\}$, and at this moment *st* is equal to *u*1 since the retracting sequence is being executed.

In summary, the following sequences of schema activations can take place when the handle is moved (this is just an informal presentation):

- $Up1Ok \rightarrow UpOk2$
- $Up1Ok \rightarrow ChangeHandleUpDown \rightarrow UpDown2 \rightarrow Down2$

The following schemas complete the specification of this step of the retraction sequence. Note that *Up*2*E*, unlike *Up*1*E*, does not include the negation of $hPos = up$ because this is considered in *UpDown*2.

Up2E

$\Xi DoorOpeningEV$; $\Xi EVst$; $\Xi StateCounter$; $\Xi DoorClosingEV$; $\Xi Time$; $\Xi Handle$

$\neg (st = u1 \land 200 \leq now - stEV \land (100 \leq now - stDCEV \lor stDCEV = 0))$

$Up2 == Up2Ok \lor UpDown2 \lor Up2E$

The remaining schemas ($Up3, \dots, Up8$ as well as those of the outgoing sequence) are rather similar to $Up2$. Hence, only minimal explanations are given.

$Up31$ deals with the part of step 3 when the shock absorbers are in flight (relaxed). In this case the gear retraction electro-valve is stimulated. This schema adds two preconditions:

- All the doors are already opened ($\operatorname{ran} dOp = \{y\}$); and
- All the shock absorbers are in flight ($\operatorname{ran} sa = \{n\}$).

$$
\begin{array}{l}
\textbf{Up31} \\
\hline
\Delta GearsRetractingEV;\ \Delta EVst;\ \Delta StateCounter \\
\Xi GearsExtendingEV;\ \Xi DoorsOpening;\ \Xi ShockAbsorbers;\ \Xi Time;\ \Xi Handle \\
\hline
st = u2 \\
hPos = up \\
\operatorname{ran} dOp = \{y\} \\
\operatorname{ran} sa = \{n\} \\
200 \leq now - stEV \\
100 \leq now - stGEEV \lor stGEEV = 0 \\
grEV' = pressing \\
stGREV' = now \\
stEV' = now \\
st' = u3 \\
\hline
\end{array}
$$

$U32$ deals with the opposite case: one or more shock absorbers are not seen as in flight ($\operatorname{ran} sa \neq \{n\}$). In this case nothing is done except advancing the internal state to $u4$.

$$
\begin{array}{l}
\textbf{Up32} \\
\hline
\Delta StateCounter \\
\Xi DoorsOpening;\ \Xi ShockAbsorbers;\ \Xi Handle \\
\hline
st = u2 \\
hPos = up \\
\operatorname{ran} dOp = \{y\} \\
\operatorname{ran} sa \neq \{n\} \\
st' = u4 \\
\hline
\end{array}
$$

The schema for reverting the order at this point is:

$$
\begin{array}{l}
\textbf{UpDown3} \\
\hline
\Delta StateCounter \\
\Xi DoorsOpening;\ \Xi ShockAbsorbers;\ \Xi Handle \\
\hline
st = u2 \\
hPos = down \\
st' = d2 \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textbf{Up3E} \\
\hline
\Xi GearsRetractingEV;\ \Xi EVst;\ \Xi StateCounter;\ \Xi GearsExtendingEV \\
\Xi DoorsOpening;\ \Xi ShockAbsorbers;\ \Xi Time;\ \Xi Handle \\
\hline
\neg\,(st = u2 \\
\quad \land \operatorname{ran} dOp = \{y\} \land 200 \leq now - stEV \land (100 \leq now - stGEEV \lor stGEEV = 0)) \\
\hline
\end{array}
$$

$Up3 == Up31 \lor Up32 \lor UpDown3 \lor Up3E$

Since stopping the stimulation of electro-valves is subjected to timing restrictions (1 s between any two of them), the following schema takes the time marks concerning stopping the stimulation. The fourth step of the retracting sequences requires that all gears are locked up: $\operatorname{ran} gRec = \{y\}$.

$$
\begin{array}{l}
\textbf{Up4Ok} \\
\hline
\Delta GearsRetractingEV;\ \Delta EVsp;\ \Delta StateCounter \\
\Xi GearsRetracting;\ \Xi Time;\ \Xi Handle \\
\hline
st = u3 \\
hPos = up \\
\operatorname{ran} gRec = \{y\} \\
1000 \leq now - spEV \lor spEV = 0 \\
grEV' = idle \\
spEV' = now \\
st' = u4 \\
stGREV' = stGREV \\
\hline
\end{array}
$$

The schema for reverting the order at this point is:

$$
\begin{array}{l}
\textbf{UpDown4} \\
\hline
\Delta GearsRetractingEV;\ \Delta EVsp;\ \Delta StateCounter \\
\Xi GearsRetracting;\ \Xi Time;\ \Xi Handle \\
\hline
st = u3 \\
hPos = down \\
grEV' = idle \\
spEV' = now \\
st' = d2 \\
stGREV' = stGREV \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textbf{Up4E} \\
\hline
\Xi GearsRetractingEV;\ \Xi EVsp;\ \Xi StateCounter;\ \Xi GearsRetracting;\ \Xi Time;\ \Xi Handle \\
\hline
\neg\,(st = u3 \land \operatorname{ran} gRec = \{y\} \land (1000 \leq now - spEV \lor spEV = 0)) \\
\hline
\end{array}
$$

$Up4 == Up4Ok \lor UpDown4 \lor Up4E$

In $Up5$ is:

$$1000 \leq now - spEV$$

and not:

$$1000 \leq now - spEV \lor spEV = 0$$

because to get to this schema the system has passed through schema $Up4$ where the gear retraction electro-valves were stopped, so $spEV$ cannot be zero.

Up5Ok

$\Delta DoorOpeningEV$; $\Delta EVsp$; $\Delta StateCounter$
$\Xi Time$; $\Xi Handle$

$st = u4$
$hPos = up$
$1000 \leq now - spEV$
$doEV' = idle$
$spEV' = now$
$st' = u5$
$stDOEV' = stDOEV$

The schema for reverting the order at this point is:

UpDown5

$\Delta StateCounter$
$\Xi DoorOpeningEV$; $\Xi EVsp$; $\Xi Time$; $\Xi Handle$

$st = u4$
$hPos = down$
$st' = d2$

Up5E

$\Xi DoorOpeningEV$; $\Xi EVsp$; $\Xi StateCounter$; $\Xi Time$; $\Xi Handle$

$\neg (st = u4 \land 1000 \leq now - spEV)$

$$Up5 == Up5Ok \lor UpDown5 \lor Up5E$$

In $Up6$ is:

$$100 \leq now - stDOEV$$

and not:

$$100 \leq now - stDOEV \lor stDOEV = 0$$

because to get to this schema the system has passed through schema $Up2$ where the doors were opened, so $stDOEV$ cannot be zero.

$Up6Ok$

$\Delta DoorClosingEV$; $\Delta EVst$; $\Delta StateCounter$
$\Xi DoorOpeningEV$; $\Xi Time$; $\Xi Handle$

$st = u5$
$hPos = up$
$200 \leq now - stEV$
$100 \leq now - stDOEV$
$dcEV' = pressing$
$stDCEV' = now$
$stEV' = now$
$st' = u6$

The schema for reverting the order is:

$UpDown6$

$\Delta StateCounter$
$\Xi DoorClosingEV$; $\Xi EVst$; $\Xi DoorOpeningEV$; $\Xi Time$; $\Xi Handle$

$st = u5$
$hPos = down$
$st' = d1$

$Up6E$

$\Xi DoorClosingEV$; $\Xi EVst$; $\Xi StateCounter$; $\Xi DoorOpeningEV$; $\Xi Time$; $\Xi Handle$

$\neg (st = u5 \land 200 \leq now - stEV \land 100 \leq now - stDOEV)$

$Up6 == Up6Ok \lor UpDown6 \lor Up6E$

$Up7Ok$

$\Delta DoorClosingEV$; $\Delta EVsp$; $\Delta StateCounter$
$\Xi DoorsClosing$; $\Xi Time$; $\Xi Handle$

$st = u6$
$hPos = up$
$\operatorname{ran} dCl = \{y\}$
$1000 \leq now - spEV$
$dcEV' = idle$
$spEV' = now$
$st' = u7$
$stDCEV' = stDCEV$

The schema for reverting the order at this point is:

$$
\begin{array}{l}
\textit{UpDown7} \\
\hline
\Delta StateCounter;\ \Delta EVsp;\ \Delta DoorClosingEV \\
\Xi DoorsClosing;\ \Xi Time;\ \Xi Handle \\
\hline
st = u6 \\
hPos = down \\
dcEV' = idle \\
spEV' = now \\
st' = d1 \\
stDCEV' = stDCEV \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textit{Up7E} \\
\hline
\Xi DoorClosingEV;\ \Xi EVsp;\ \Xi StateCounter;\ \Xi DoorsClosing;\ \Xi Time;\ \Xi Handle \\
\hline
\neg\,(st = u6 \land \operatorname{ran} dCl = \{y\} \land 1000 \leq now - spEV) \\
\hline
\end{array}
$$

$$Up7 == Up7Ok \lor UpDown7 \lor Up7E$$

The last step in the sequence restores *sp* to *init*.

$$
\begin{array}{l}
\textit{Up8Ok} \\
\hline
\Delta GeneralEV;\ \Delta EVsp;\ \Delta StateCounter \\
\Xi Time;\ \Xi Handle \\
\hline
st = u7 \\
hPos = up \\
1000 \leq now - spEV \\
gEV' = idle \\
spGEV' = now \\
spEV' = now \\
st' = init \\
stGEV' = stGEV \\
\hline
\end{array}
$$

The schema for reverting the order at this point is:

$$
\begin{array}{l}
\textit{UpDown8} \\
\hline
\Delta StateCounter \\
\Xi GeneralEV;\ \Xi EVsp;\ \Xi Time;\ \Xi Handle \\
\hline
st = u7 \\
hPos = down \\
st' = d1 \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textit{Up8E} \\
\hline
\Xi \textit{GeneralEV};\ \Xi \textit{EVsp};\ \Xi \textit{StateCounter};\ \Xi \textit{Time};\ \Xi \textit{Handle} \\
\hline
\neg\ (st = u7 \land 1000 \leq now - spEV) \\
\hline
\end{array}
$$

$Up8 == Up8Ok \lor UpDown8 \lor Up8E$

4.6.3 Outgoing sequence

The outgoing sequence has a similar structure and similar predicates with respect to the retractions sequences, so no explanations are given.

$$
\begin{array}{l}
\textit{Down1Ok} \\
\hline
\Delta \textit{GeneralEV};\ \Delta \textit{EVst};\ \Delta \textit{StateCounter} \\
\Xi \textit{Time};\ \Xi \textit{Handle} \\
\hline
st = d0 \\
hPos = down \\
200 \leq now - stEV \lor stEV = 0 \\
gEV' = pressing \\
stGEV' = now \\
stEV' = now \\
st' = d1 \\
spGEV' = spGEV \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textit{Down1E} \\
\hline
\Xi \textit{GeneralEV};\ \Xi \textit{EVst};\ \Xi \textit{StateCounter};\ \Xi \textit{Time};\ \Xi \textit{Handle} \\
\hline
\neg\ (st = d0 \land hPos = down \land (200 \leq now - stEV \lor stEV = 0)) \\
\hline
\end{array}
$$

$Down1 == Down1Ok \lor Down1E$

$$
\begin{array}{l}
\textit{Down2Ok} \\
\hline
\Delta \textit{DoorOpeningEV};\ \Delta \textit{EVst};\ \Delta \textit{StateCounter} \\
\Xi \textit{DoorClosingEV};\ \Xi \textit{Time};\ \Xi \textit{Handle} \\
\hline
st = d1 \\
hPos = down \\
200 \leq now - stEV \\
100 \leq now - stDCEV \lor stDCEV = 0 \\
doEV' = pressing \\
stDOEV' = now \\
stEV' = now \\
st' = d2 \\
\hline
\end{array}
$$

$$\begin{array}{l}
\textbf{DownUp2} \\
\Delta StateCounter \\
\Xi DoorOpeningEV;\ \Delta EVst;\ \Xi DoorClosingEV;\ \Xi Time;\ \Xi Handle \\
\hline
hPos = up \\
st = d1 \\
st' = u1
\end{array}$$

$$\begin{array}{l}
\textbf{Down2E} \\
\Xi DoorOpeningEV;\ \Xi EVst;\ \Xi StateCounter;\ \Xi DoorClosingEV;\ \Xi Time;\ \Xi Handle \\
\hline
\neg\,(st = d1 \land 200 \leq now - stEV \land (100 \leq now - stDCEV \lor stDCEV = 0))
\end{array}$$

$$Down2 == Down2Ok \lor DownUp2 \lor Down2E$$

$$\begin{array}{l}
\textbf{Down3Ok} \\
\Delta GearsExtendingEV;\ \Delta EVst;\ \Delta StateCounter \\
\Xi GearsRetractingEV;\ \Xi DoorsOpening;\ \Xi Time;\ \Xi Handle \\
\hline
st = d2 \\
hPos = down \\
\operatorname{ran} dOp = \{y\} \\
200 \leq now - stEV \\
100 \leq now - stGREV \lor stGREV = 0 \\
geEV' = pressing \\
stGEEV' = now \\
stEV' = now \\
st' = d3
\end{array}$$

$$\begin{array}{l}
\textbf{DownUp3} \\
\Delta StateCounter \\
\Xi GearsExtendingEV;\ \Xi EVst;\ \Xi GearsRetractingEV;\ \Xi DoorsOpening;\ \Xi Time;\ \Xi Handle \\
\hline
hPos = up \\
st = d2 \\
st' = u2
\end{array}$$

$$\begin{array}{l}
\textbf{Down3E} \\
\Xi GearsExtendingEV;\ \Xi EVst;\ \Xi StateCounter \\
\Xi GearsRetractingEV;\ \Xi DoorsOpening;\ \Xi Time;\ \Xi Handle \\
\hline
\neg\,(st = d2 \\
\qquad \land \operatorname{ran} dOp = \{y\} \land 200 \leq now - stEV \land (100 \leq now - stGREV \lor stGREV = 0))
\end{array}$$

$$Down3 == Down3Ok \lor DownUp3 \lor Down3E$$

Down4Ok

$$\Delta GearsExtendingEV;\ \Delta EVsp;\ \Delta StateCounter$$
$$\Xi GearsExtending;\ \Xi Time;\ \Xi Handle$$

$$st = d3$$
$$hPos = down$$
$$\operatorname{ran} gExt = \{y\}$$
$$1000 \leq now - spEV \lor spEV = 0$$
$$geEV' = idle$$
$$spEV' = now$$
$$st' = d4$$
$$stGEEV' = stGEEV$$

DownUp4

$$\Delta StateCounter;\ \Delta EVsp;\ \Delta GearsExtendingEV$$
$$\Xi GearsExtending;\ \Xi Time;\ \Xi Handle$$

$$hPos = up$$
$$st = d3$$
$$geEV' = idle$$
$$spEV' = now$$
$$st' = u2$$
$$stGEEV' = stGEEV$$

Down4E

$$\Xi GearsExtendingEV;\ \Xi EVsp;\ \Xi StateCounter;\ \Xi GearsExtending;\ \Xi Time;\ \Xi Handle$$

$$\neg\, (st = d3 \land \operatorname{ran} gExt = \{y\} \land (1000 \leq now - spEV \lor spEV = 0))$$

$$Down4 == Down4Ok \lor DownUp4 \lor Down4E$$

Down5Ok

$$\Delta DoorOpeningEV;\ \Delta EVsp;\ \Delta StateCounter$$
$$\Xi Time;\ \Xi Handle$$

$$st = d4$$
$$hPos = down$$
$$1000 \leq now - spEV$$
$$doEV' = idle$$
$$spEV' = now$$
$$st' = d5$$
$$stDOEV' = stDOEV$$

$$\begin{array}{|l}
\multicolumn{1}{l}{__ DownUp5 ____} \\
\Delta StateCounter \\
\Xi DoorOpeningEV; \Xi EVsp; \Xi Time; \Xi Handle \\
\hline
hPos = up \\
st = d4 \\
st' = u2 \\
\hline
\end{array}$$

$$\begin{array}{|l}
\multicolumn{1}{l}{__ Down5E ____} \\
\Xi DoorOpeningEV; \Xi EVsp; \Xi StateCounter; \Xi Time; \Xi Handle \\
\hline
\neg (st = d4 \land 1000 \leq now - spEV) \\
\hline
\end{array}$$

$Down5 == Down5Ok \lor DownUp5 \lor Down5E$

$$\begin{array}{|l}
\multicolumn{1}{l}{__ Down6Ok ____} \\
\Delta DoorClosingEV; \Delta EVst; \Delta StateCounter \\
\Xi DoorOpeningEV; \Xi Time; \Xi Handle \\
\hline
st = d5 \\
hPos = down \\
200 \leq now - stEV \\
100 \leq now - stDOEV \\
dcEV' = pressing \\
stDCEV' = now \\
stEV' = now \\
st' = d6 \\
\hline
\end{array}$$

$$\begin{array}{|l}
\multicolumn{1}{l}{__ DownUp6 ____} \\
\Delta StateCounter \\
\Xi DoorClosingEV; \Xi EVst; \Xi DoorOpeningEV; \Xi Time; \Xi Handle \\
\hline
hPos = up \\
st = d5 \\
st' = u1 \\
\hline
\end{array}$$

$$\begin{array}{|l}
\multicolumn{1}{l}{__ Down6E ____} \\
\Xi DoorClosingEV; \Xi EVst; \Xi StateCounter; \Xi DoorOpeningEV; \Xi Time; \Xi Handle \\
\hline
\neg (st = d5 \land 200 \leq now - stEV \land 100 \leq now - stDOEV) \\
\hline
\end{array}$$

$Down6 == Down6Ok \lor DownUp6 \lor Down6E$

Down7Ok

$\Delta DoorClosingEV; \Delta EVsp; \Delta StateCounter$
$\Xi DoorsClosing; \Xi Time; \Xi Handle$

$st = d6$
$hPos = down$
$\operatorname{ran} dCl = \{y\}$
$1000 \leq now - spEV$
$dcEV' = idle$
$spEV' = now$
$st' = d7$
$stDCEV' = stDCEV$

DownUp7

$\Delta StateCounter; \Delta EVsp; \Delta DoorClosingEV$
$\Xi DoorsClosing; \Xi Time; \Xi Handle$

$hPos = up$
$st = d6$
$dcEV' = idle$
$spEV' = now$
$st' = u1$
$stDCEV' = stDCEV$

Down7E

$\Xi DoorClosingEV; \Xi EVsp; \Xi StateCounter; \Xi DoorsClosing; \Xi Time; \Xi Handle$

$\neg (st = d6 \wedge \operatorname{ran} dCl = \{y\} \wedge 1000 \leq now - spEV)$

$Down7 == Down7Ok \vee DownUp7 \vee Down7E$

Down8Ok

$\Delta GeneralEV; \Delta EVsp; \Delta StateCounter$
$\Xi Time; \Xi Handle$

$st = d7$
$hPos = down$
$1000 \leq now - spEV$
$gEV' = idle$
$spGEV' = now$
$spEV' = now$
$st' = init$
$stGEV' = stGEV$

$$\begin{array}{|l}
\hline
\textit{DownUp8} \\
\Delta \textit{StateCounter} \\
\Xi \textit{GeneralEV};\ \Xi \textit{EVsp};\ \Xi \textit{Time};\ \Xi \textit{Handle} \\
\hline
\textit{hPos} = \textit{up} \\
\textit{st} = \textit{d7} \\
\textit{st}' = \textit{u1} \\
\hline
\end{array}$$

$$\begin{array}{|l}
\hline
\textit{Down8E} \\
\Xi \textit{GeneralEV};\ \Xi \textit{EVsp};\ \Xi \textit{StateCounter};\ \Xi \textit{Time};\ \Xi \textit{Handle} \\
\hline
\neg\ (\textit{st} = \textit{d7} \land 1000 \leq \textit{now} - \textit{spEV}) \\
\hline
\end{array}$$

$$\textit{Down8} == \textit{Down8Ok} \lor \textit{DownUp8} \lor \textit{Down8E}$$

4.7 Time Advance

Although Z is not very well equipped to deal with time constraints and with real-time specifications in general [16], the following schema specifies that time always advances at a rate of one time unit (in this case one millisecond). Since *Tick* is always enabled it can be "executed" whenever there are no other operation being "executed".

$$\begin{array}{|l}
\hline
\textit{Tick} \\
\Delta \textit{Time} \\
\hline
\textit{now}' = \textit{now} + 1 \\
\hline
\end{array}$$

4.8 Operations Concerning Health Monitoring

Health monitoring concerns with detecting situations that are deemed as anomalies. The anomalies concerning sensor validity are formalized in Sect. 4.5. Then, this section contains the rest of the situations that can cause an anomaly. When an anomaly is detected (i.e. when the conditions for an anomaly become true) the software executes an action whose specification is the following schema:

$$\textit{Anomaly} == [\Delta \textit{CockpitA} \mid \textit{lgsfl}' = \textit{on}]$$

Recall that is has been assumed that the software stops when this action is executed and that *lgsfl* represents both the actual light in the cockpit and an internal state variable whose value can be observed and modified by the software.

4.8.1 Anomalies Related to the Analogical Switch

The first anomaly related to the analogical switch is produced when it is seen open 1 second after the handle position has changed. This situation is formalized in the following schema:

$$
\begin{array}{|l}
\hline
\textit{AnalogicalSwitchM1} \\
\Xi \textit{AnalogicalSwitch}; \Xi \textit{Handle}; \Xi \textit{Time} \\
\hline
as = y \\
1000 \leq now - lHPCh \\
lHPCh \neq 0 \\
\hline
\end{array}
$$

The analogical switch is open when $as = y$ and $lHPCh$ records the time of the last change of the handle if it is different than zero (because if it is equal to zero it means the handle was never changed).

If the conditions given in $AnalogicalSwitchM1$ are not true, then the system has nothing to do. Therefore, a schema totalizing the operation is given:

$$
\begin{array}{|l}
\hline
\textit{AnalogicalSwitchM1E} \\
\Xi \textit{AnalogicalSwitch}; \Xi \textit{Handle}; \Xi \textit{Time} \\
\hline
\neg (as = y \land 1000 \leq now - lHPCh \land lHPCh \neq 0) \\
\hline
\end{array}
$$

The second situation related to the analogical switch that causes an anomaly is produced when it is seen closed 1.5 second after a time interval of 20 seconds during which the handle position has not changed. This is formalized by the following schema:

$$
\begin{array}{|l}
\hline
\textit{AnalogicalSwitchM2} \\
\Xi \textit{AnalogicalSwitch}; \Xi \textit{Handle}; \Xi \textit{Time} \\
\hline
as = n \\
1500 \leq now - l20 \\
l20 \neq 0 \\
\hline
\end{array}
$$

The analogical switch is closed when $as = n$ and $l20$ records the last time the handle position has not changed for 20 seconds. This schema is complemented as the previous one:

$$
\begin{array}{|l}
\hline
\textit{AnalogicalSwitchM2E} \\
\Xi \textit{AnalogicalSwitch}; \Xi \textit{Handle}; \Xi \textit{Time} \\
\hline
\neg (as = n \land 1500 \leq now - l20 \land l20 \neq 0) \\
\hline
\end{array}
$$

The full operation is specified as follows:

$$
\begin{array}{l}
\textit{AnalogicalSwitchM} == \\
\quad ((\textit{AnalogicalSwitchM1} \lor \textit{AnalogicalSwitchM2}) \land \textit{Anomaly}) \\
\quad \lor \textit{AnalogicalSwitchM1E} \lor \textit{AnalogicalSwitchM2E}
\end{array}
$$

$AnalogicalSwitchM2$ shows that it is necessary to specify an operation that monitors when 20 seconds have elapsed since the last change of the handle position. The next schema updates the variable $l20$ when that condition holds:

$$\begin{array}{|l}\hline \textit{HandleNotChangedOk} \\ \Delta Handle;\ \Xi Time \\ \hline now - lHPCh = 20 \\ lHPCh \neq 0 \\ l20' = now \\ hPos' = hPos \\ lHPCh' = lHPCh \\ \hline \end{array}$$

$$\begin{array}{|l}\hline \textit{HandleNotChangedE} \\ \Xi Handle;\ \Xi Time \\ \hline \neg\ (now - lHPCh = 20 \land lHPCh \neq 0) \\ \hline \end{array}$$

$$HandleNotChanged == HandleNotChangedOk \lor HandleNotChangedE$$

Perhaps, as Lamport suggests [25, Chapter 9], the following precondition for *HandleNotChanged* would be more realizable:

$$|now - lHPCh| \leq 20 + \varepsilon$$

for some $\varepsilon > 0$. Or alternatively this one would be as realizable and closer to the real requirement:

$$now - lHPCh \geq 20 + \varepsilon$$

4.8.2 Anomalies Related to the Hydraulic Circuit

The situations where an anomaly related to the hydraulic circuit is produced are the following:

- If the hydraulic circuit is still unpressurized 2 seconds after the general electro-valve has been stimulated, then an anomaly is detected. The following schema formalizes this situation:

$$\begin{array}{|l}\hline \textit{HydraulicCircuitM1} \\ \Xi HydraulicCircuit;\ \Xi GeneralEV;\ \Xi Time \\ \hline hc = n \\ 2000 \leq now - stGEV \\ stGEV \neq 0 \\ \hline \end{array}$$

$$\begin{array}{|l}\hline \textit{HydraulicCircuitM1E} \\ \Xi HydraulicCircuit;\ \Xi GeneralEV;\ \Xi Time \\ \hline \neg\ (hc = n \land 2000 \leq now - stGEV \land stGEV \neq 0) \\ \hline \end{array}$$

- If the hydraulic circuit is still pressurized 10 seconds after the general electro- valve has been stopped, then an anomaly is detected. The following schema formalizes this situation:

$$
\begin{array}{l}
\textit{HydraulicCircuitM2} \\
\Xi \textit{HydraulicCircuit}; \Xi \textit{GeneralEV}; \Xi \textit{Time} \\
\hline
hc = y \\
10000 \leq now - spGEV \\
spGEV \neq 0
\end{array}
$$

$$
\begin{array}{l}
\textit{HydraulicCircuitM2E} \\
\Xi \textit{HydraulicCircuit}; \Xi \textit{GeneralEV}; \Xi \textit{Time} \\
\hline
\neg (hc = y \land 10000 \leq now - spGEV \land spGEV \neq 0)
\end{array}
$$

Therefore, the full operation is as follows:

$$
\begin{array}{l}
\textit{HydraulicCircuitM} == \\
\quad (\textit{HydraulicCircuitM1} \lor \textit{HydraulicCircuitM2}) \land \textit{Anomaly} \\
\quad \lor \textit{HydraulicCircuitM1E} \lor \textit{HydraulicCircuitM2E}
\end{array}
$$

4.8.3 Anomalies Related to Doors Motion

There are four situations related to door motion that lead to an anomaly. Since these are quite similar in spirit to the previous ones there will be no comments.

$$
\begin{array}{l}
\textit{DoorsMotionM1} \\
\Xi \textit{Doors}; \Xi \textit{DoorOpeningEV}; \Xi \textit{Time} \\
\hline
\operatorname{ran} dCl \neq \{n\} \\
7000 \leq now - stDOEV
\end{array}
$$

$$
\begin{array}{l}
\textit{DoorsMotionM1E} \\
\Xi \textit{Doors}; \Xi \textit{DoorOpeningEV}; \Xi \textit{Time} \\
\hline
\neg (\operatorname{ran} dCl \neq \{n\} \land 7000 \leq now - stDOEV)
\end{array}
$$

$$
\begin{array}{l}
\textit{DoorsMotionM2} \\
\Xi \textit{DoorsOpening}; \Xi \textit{DoorOpeningEV}; \Xi \textit{Time} \\
\hline
\operatorname{ran} dOp \neq \{y\} \\
7000 \leq now - stDOEV
\end{array}
$$

$$\begin{array}{l}\textbf{DoorsMotionM2E}\\ \hline \Xi DoorsOpening;\ \Xi DoorOpeningEV;\ \Xi Time\\ \hline \lnot\,(\operatorname{ran} dOp \neq \{y\} \land 7000 \leq now - stDOEV)\end{array}$$

$$\begin{array}{l}\textbf{DoorsMotionM3}\\ \hline \Xi DoorsOpening;\ \Xi DoorClosingEV;\ \Xi Time\\ \hline \operatorname{ran} dOp \neq \{n\}\\ 7000 \leq now - stDCEV\end{array}$$

$$\begin{array}{l}\textbf{DoorsMotionM3E}\\ \hline \Xi DoorsOpening;\ \Xi DoorClosingEV;\ \Xi Time\\ \hline \lnot\,(\operatorname{ran} dOp \neq \{n\} \land 7000 \leq now - stDCEV)\end{array}$$

$$\begin{array}{l}\textbf{DoorsMotionM4}\\ \hline \Xi Doors;\ \Xi DoorClosingEV;\ \Xi Time\\ \hline \operatorname{ran} dCl \neq \{y\}\\ 7000 \leq now - stDCEV\end{array}$$

$$\begin{array}{l}\textbf{DoorsMotionM4E}\\ \hline \Xi Doors;\ \Xi DoorClosingEV;\ \Xi Time\\ \hline \lnot\,(\operatorname{ran} dCl \neq \{y\} \land 7000 \leq now - stDCEV)\end{array}$$

$$\begin{array}{l}DoorsMotionM ==\\ \quad (DoorsMotionM1 \lor DoorsMotionM2 \lor DoorsMotionM3 \lor DoorsMotionM4) \land Anomaly\\ \quad \lor DoorsMotionM1E \lor DoorsMotionM2E \lor DoorsMotionM3E \lor DoorsMotionM4E\end{array}$$

4.8.4 Anomalies Related to Gears Motion

There are four situations related to gear motion that lead to an anomaly. Since these are quite similar in spirit to the previous ones there will be no comments.

$$\begin{array}{l}\textbf{GearsMotionM1}\\ \hline \Xi GearsRetracting;\ \Xi GearsRetractingEV;\ \Xi Time\\ \hline \operatorname{ran} gRec \neq \{n\}\\ 7000 \leq now - stGREV\end{array}$$

GearsMotionM1E

$\Xi GearsRetracting;\ \Xi GearsRetractingEV;\ \Xi Time$

$\neg\,(\operatorname{ran} gRec \neq \{n\} \wedge 7000 \leq now - stGREV)$

GearsMotionM2

$\Xi GearsRetracting;\ \Xi GearsRetractingEV;\ \Xi Time$

$\operatorname{ran} gRec \neq \{y\}$
$10000 \leq now - stGREV$

GearsMotionM2E

$\Xi GearsRetracting;\ \Xi GearsRetractingEV;\ \Xi Time$

$\neg\,(\operatorname{ran} gRec \neq \{y\} \wedge 10000 \leq now - stGREV)$

GearsMotionM3

$\Xi GearsExtending;\ \Xi GearsExtendingEV;\ \Xi Time$

$\operatorname{ran} gExt \neq \{n\}$
$7000 \leq now - stGEEV$

GearsMotionM3E

$\Xi GearsExtending;\ \Xi GearsExtendingEV;\ \Xi Time$

$\neg\,(\operatorname{ran} gExt \neq \{n\} \wedge 7000 \leq now - stGEEV)$

GearsMotionM4

$\Xi GearsExtending;\ \Xi GearsExtendingEV;\ \Xi Time$

$\operatorname{ran} gExt \neq \{y\}$
$10000 \leq now - stGEEV$

GearsMotionM4E

$\Xi GearsExtending;\ \Xi GearsExtendingEV;\ \Xi Time$

$\neg\,(\operatorname{ran} gExt \neq \{y\} \wedge 10000 \leq now - stGEEV)$

$$GearsMotionM == (GearsMotionM1 \vee GearsMotionM2 \vee GearsMotionM3 \vee GearsMotionM4) \wedge Anomaly \vee GearsMotionM1E \vee GearsMotionM2E \vee GearsMotionM3E \vee GearsMotionM4E$$

```
loadspec answer-ttf-ft.tex
replaceaxdef
selop ReadGearsExtending
selop ReadGearsRetracting
selop ReadShockAbsorbers
selop ReadDoorsOpening
selop ReadDoorsClosing
selop ReadHydraulicCircuit
selop ReadAnalogicalSwitch
selop ChangeHandle
selop GearsLockedDown
selop GearsManeuvering
selop Up1
selop Up2
selop Up3
selop Up4
selop Up5
selop Up6
selop Up7
selop Up8
selop Down1
selop Down2
selop Down3
selop Down4
selop Down5
selop Down6
selop Down7
selop Down8
selop AnalogicalSwitchM
selop HandleNotChanged
selop HydraulicCircuitM
selop DoorsMotionM
selop GearsMotionM
selop Valid
genalltt
addtactic ReadGearsExtending FT g?
addtactic ReadGearsRetracting FT g?
addtactic ReadShockAbsorbers FT g?
addtactic ReadDoorsOpening FT g?
addtactic ReadDoorsClosing FT g?
addtactic Valid SP \rres i? \rres \{y\}
addtactic Valid SP \rres i? \rres \{n\}
```

Figure 5: Fastest script (part 1)

```
addtactic Up1_DNF_1 SP - now - stEV
addtactic Up1_DNF_1 SP \leq 200 \leq now - stEV
addtactic Up2_DNF_1 SP - now - stEV
addtactic Up2_DNF_1 SP \leq 200 \leq now - stEV
addtactic Up2_DNF_1 SP - now - stDCEV
addtactic Up2_DNF_1 SP \leq 100 \leq now - stDCEV
addtactic Up3_DNF_1 SP - now - stGEEV
addtactic Up3_DNF_1 SP \leq 100 \leq now - stGEEV
addtactic Up3_DNF_1 SP - now - stEV
addtactic Up3_DNF_1 SP \leq 200 \leq now - stEV
addtactic Up4_DNF_1 SP - now - spEV
addtactic Up4_DNF_1 SP \leq 1000 \leq now - spEV
addtactic Up5_DNF_1 SP - now - spEV
addtactic Up5_DNF_1 SP \leq 1000 \leq now - spEV
addtactic Up6_DNF_1 SP - now - stEV
addtactic Up6_DNF_1 SP \leq 200 \leq now - stEV
addtactic Up6_DNF_1 SP - now - stDOEV
addtactic Up6_DNF_1 SP \leq 100 \leq now - stDOEV
addtactic Up7_DNF_1 SP - now - spEV
addtactic Up7_DNF_1 SP \leq 1000 \leq now - spEV
addtactic Up8_DNF_1 SP - now - spEV
addtactic Up8_DNF_1 SP \leq 1000 \leq now - spEV
addtactic Down1_DNF_1 SP - now - stEV
addtactic Down1_DNF_1 SP \leq 200 \leq now - stEV
addtactic Down2_DNF_1 SP - now - stEV
addtactic Down2_DNF_1 SP \leq 200 \leq now - stEV
addtactic Down2_DNF_1 SP - now - stDCEV
addtactic Down2_DNF_1 SP \leq 100 \leq now - stDCEV
addtactic Down3_DNF_1 SP - now - stEV
addtactic Down3_DNF_1 SP \leq 200 \leq now - stEV
addtactic Down3_DNF_1 SP - now - stGREV
addtactic Down3_DNF_1 SP \leq 100 \leq now - stGREV
addtactic Down4_DNF_1 SP - now - spEV
addtactic Down4_DNF_1 SP \leq 1000 \leq now - spEV
addtactic Down5_DNF_1 SP - now - spEV
addtactic Down5_DNF_1 SP \leq 1000 \leq now - spEV
addtactic Down6_DNF_1 SP - now - stEV
addtactic Down6_DNF_1 SP \leq 200 \leq now - stEV
addtactic Down6_DNF_1 SP - now - stDOEV
addtactic Down6_DNF_1 SP \leq 100 \leq now - stDOEV
addtactic Down7_DNF_1 SP - now - spEV
addtactic Down7_DNF_1 SP \leq 1000 \leq now - spEV
addtactic Down8_DNF_1 SP - now - spEV
addtactic Down8_DNF_1 SP \leq 1000 \leq now - spEV
```

Figure 6: Fastest script (part 2)

```
addtactic AnalogicalSwitchM_DNF_1 SP - now - lHPCh
addtactic AnalogicalSwitchM_DNF_1 SP \leq 1000 \leq now - lHPCh
addtactic AnalogicalSwitchM_DNF_2 SP - now - l20
addtactic AnalogicalSwitchM_DNF_2 SP \leq 1500 \leq now - l20
addtactic HandleNotChanged_DNF_1 SP - now - lHPCh
addtactic HydraulicCircuitM_DNF_1 SP - now - stGEV
addtactic HydraulicCircuitM_DNF_1 SP \leq 2000 \leq now - stGEV
addtactic HydraulicCircuitM_DNF_2 SP - now - spGEV
addtactic HydraulicCircuitM_DNF_2 SP \leq 10000 \leq now - spGEV
addtactic DoorsMotionM_DNF_1 SP - now - stDOEV
addtactic DoorsMotionM_DNF_1 SP \leq 7000 \leq now - stDOEV
addtactic DoorsMotionM_DNF_2 SP - now - stDOEV
addtactic DoorsMotionM_DNF_2 SP \leq 7000 \leq now - stDOEV
addtactic DoorsMotionM_DNF_3 SP - now - stDCEV
addtactic DoorsMotionM_DNF_3 SP \leq 7000 \leq now - stDCEV
addtactic DoorsMotionM_DNF_4 SP - now - stDCEV
addtactic DoorsMotionM_DNF_4 SP \leq 7000 \leq now - stDCEV
addtactic GearsMotionM_DNF_1 SP - now - stGREV
addtactic GearsMotionM_DNF_1 SP \leq 7000 \leq now - stGREV
addtactic GearsMotionM_DNF_2 SP - now - stGREV
addtactic GearsMotionM_DNF_2 SP \leq 10000 \leq now - stGREV
addtactic GearsMotionM_DNF_3 SP - now - stGEEV
addtactic GearsMotionM_DNF_3 SP \leq 7000 \leq now - stGEEV
addtactic GearsMotionM_DNF_4 SP - now - stGEEV
addtactic GearsMotionM_DNF_4 SP \leq 10000 \leq now - stGEEV
genalltt
prunett
genalltca
```

Figure 7: Fastest script (part 3)

5 Test Case Generation from the Z Specification

Figures 5-7 list the Fastest script used to generate test case from the LGS specification. The resulting test conditions and abstract test cases automatically generated are listed in appendices A and B, respectively.

6 Conclusions

After writing a Z specification of the LGS the Fastest tool was used to automatically generate almost 400 functional test cases. This shows that a formal specification helps not only to write the implementation from a solid document but that it also helps in verifying the former.

References

[1] J.-R. Abrial (1996): *The B-book: Assigning Programs to Meanings*. Cambridge University Press, New York, NY, USA.

[2] J. Barnes, R. Chapman, R. Johnson, J. Widmaier, D. Cooper & B. Everett (2006): *Engineering the Tokeneer enclave protection software*. In: *Proceedings of the IEEE International Symposium on Secure Software Engineering*, IEEE.

[3] Len Bass, Paul Clements & Rick Kazman (2003): *Software Architecture in Practice*, 2 edition. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[4] E. Bernard, B. Legeard, X. Luck & F. Peureux (2004): *Generation of Test Sequences from Formal Specifications: GSM 11-11 Standard Case Study*. *International Journal of Software Practice and Experience* 34(10), pp. 915–948.

[5] Gilles Bernot, Marie Claude Gaudel & Bruno Marre (1991): *Software testing based on formal specifications: a theory and a tool*. *Softw. Eng. J.* 6(6), pp. 387–405.

[6] Frédéric Boniol & Virginie Wiels (2014): *Landing gear system*. Technical Report, ONERA. Available at `http://www.irit.fr/ABZ2014/landing_system.pdf`.

[7] Jonathan Bowen: *Formal Methods*. http://vl.fmnet.info/.

[8] Frederick P. Brooks, Jr. (1995): *The mythical man-month (anniversary ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[9] Coq Development Team (2008): *The Coq Proof Assistant Reference Manual, Version 8.2*. LogiCal Project, Palaiseau, France.

[10] Maximiliano Cristiá (2014): *Test Case Generation from a Z Specification of the Landing Gear System*. Technical Report, CIFASIS and UNR. Available at `https://www.dropbox.com/s/8dlyu2mctmzw57m/answer-ttf.pdf`.

[11] Maximiliano Cristiá, Pablo Albertengo, Claudia Frydman, Brian Plüss & Pablo Rodríguez Monetti (2014): *Tool support for the Test Template Framework*. *Software Testing, Verification and Reliability* 24(1), pp. 3–37, doi:10.1002/stvr.1477. Available at `http://dx.doi.org/10.1002/stvr.1477`.

[12] Maximiliano Cristiá, Diego Hollmann, Pablo Albertengo, Claudia S. Frydman & Pablo Rodríguez Monetti (2011): *A Language for Test Case Refinement in the Test Template Framework*. In Shengchao Qin & Zongyan Qiu, editors: *ICFEM*, *Lecture Notes in Computer Science* 6991, Springer, pp. 601–616. Available at `http://dx.doi.org/10.1007/978-3-642-24559-6_40`.

[13] Maximiliano Cristiá, Gianfranco Rossi & Claudia S. Frydman (2013): *{log} as a Test Case Generator for the Test Template Framework*. In Robert M. Hierons, Mercedes G. Merayo & Mario Bravetti, editors: *SEFM*, *Lecture Notes in Computer Science* 8137, Springer, pp. 229–243. Available at `http://dx.doi.org/10.1007/978-3-642-40561-7_16`.

[14] Jeremy Dick & Alain Faivre (1993): *Automating the Generation and Sequencing of Test Cases from Model-Based Specifications*. In: *FME '93: Proceedings of the First International Symposium of Formal Methods Europe on Industrial-Strength Formal Methods*, Springer-Verlag, London, UK, pp. 268–284.

[15] R. Dupuis, P. Bourque, A. Abran, J. W. Moore & L. L. Tripp (2001): *The SWEBOK Project: Guide to the Software Engineering Body of Knowledge*. Stone Man Trial Version 1.00, http://www.swebok.org/ [01/12/2003].

[16] Andy S. Evans (1994): *Specifying and verifying concurrent systems using Z*. In Maurice Naftalin, Tim Denvir & Miquel Bertran, editors: *FME '94: Industrial Benefit of Formal Methods*, pp. 366–380.

[17] Leo Freitas, Mark Utting, Petra Malik & Tim Miller: *Community Z Tools (CZT) Project*. Available at `http://czt.sourceforge.net`. Last access: November 2011.

[18] Marie-Claude Gaudel (1995): *Testing Can Be Formal, Too*. In Peter D. Mosses, Mogens Nielsen & Michael I. Schwartzbach, editors: *TAPSOFT*, *Lecture Notes in Computer Science* 915, Springer, pp. 82–96. Available at `http://dx.doi.org/10.1007/3-540-59293-8_188`.

[19] Carlo Ghezzi, Mehdi Jazayeri & Dino Mandrioli (2003): *Fundamentals of software engineering (2nd ed.)*. Prentice Hall.

[20] Wolfgang Grieskamp, Yuri Gurevich, Wolfram Schulte & Margus Veanes (2002): *Generating finite state machines from abstract state machines*. In: *ISSTA '02: Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis*, ACM, New York, NY, USA, pp. 112–122.

[21] Robert M. Hierons, Kirill Bogdanov, Jonathan P. Bowen, Rance Cleaveland, John Derrick, Jeremy Dick, Marian Gheorghe, Mark Harman, Kalpesh Kapoor, Paul Krause, Gerald Lüttgen, Anthony J. H. Simons, Sergiy Vilkomir, Martin R. Woodward & Hussein Zedan (2009): *Using formal specifications to support testing*. *ACM Comput. Surv.* 41(2), pp. 1–76.

[22] M.G. Hinchey & J.P. Bowen (1999): *Industrial-strength formal methods in practice*. Formal approaches to computing and information technology, Springer. Available at `http://books.google.com/books?id=CWTu_Xs5sRcC`.

[23] ISO (2002): *Information Technology – Z Formal Specification Notation – Syntax, Type System and Semantics*. Technical Report ISO/IEC 13568, International Organization for Standardization.

[24] Jonathan Jacky (1996): *The way of Z: practical programming with formal methods*. Cambridge University Press, New York, NY, USA.

[25] Leslie Lamport (2002): *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[26] Bruno Legeard, Fabien Peureux & Mark Utting (2002): *A Comparison of the BTT and TTF Test-Generation Methods*. In: *ZB '02: Proceedings of the 2nd International Conference of B and Z Users on Formal Specification and Development in Z and B*, Springer-Verlag, London, UK, pp. 309–329.

[27] Steve McConnell (2004): *Code Complete, Second Edition*. Microsoft Press, Redmond, WA, USA.

[28] Bertrand Meyer, Arno Fiva, Ilinca Ciupa, Andreas Leitner, Yi Wei & Emmanuel Stapf (2009): *Programs That Test Themselves*. *Computer* 42, pp. 46–55, doi:10.1109/MC.2009.296. Available at `http://portal.acm.org/citation.cfm?id=1638584.1638626`.

[29] Arilo Dias Neto, Rajesh Subramanyan, Marlon Vieira, Guilherme Horta Travassos & Forrest Shull (2008): *Improving Evidence about Software Technologies: A Look at Model-Based Testing*. *IEEE Softw.* 25(3), pp. 10–13, doi:http://dx.doi.org/10.1109/MS.2008.64.

[30] Shari Lawrence Pfleeger (2001): *Software Engineering: Theory and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

[31] Gianfranco Rossi: $\{log\}$. Available at `http://www.math.unipr.it/~gianfr/setlog.Home.html`. Last access: July 2012.

[32] RTI (2002): *The Economic Impacts of Inadequate Infrastructure for Software Testing*. Planning Report 02-3, National Institute of Standards and Technology, Gaithersburg, MD. Available at `http://www.nist.gov/director/prog-ofc/report02-3.pdf`.

[33] Mark Saaltink (1997): *The Z/EVES System*. In Jonathan P. Bowen, Michael G. Hinchey & David Till, editors: *ZUM*, *Lecture Notes in Computer Science* 1212, Springer, pp. 72–85. Available at `http://dx.doi.org/10.1007/BFb0027284`.

[34] S. Souza, J. Maldonado, S. Fabbri & P. Masiero (2000): *Statecharts Specifications: A Family of Coverage Testing Criteria*. In: *CLEI'2000 - XXVI Latin-American Conference of Informatics*, CLEI, México DF, México.

[35] J. M. Spivey (1992): *The Z notation: a reference manual*. Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK.

[36] P. Stocks (1993): *Applying Formal Methods to Software Testing*. Ph.D. thesis, Department of Computer Science, University of Queensland.

[37] P. Stocks & D. Carrington (1996): *A Framework for Specification-Based Testing*. *IEEE Transactions on Software Engineering* 22(11), pp. 777–793.

[38] Mark Utting & Bruno Legeard (2006): *Practical Model-Based Testing: A Tools Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

A Test Conditions

ReadGearsExtending_DNF_2

ReadGearsExtending_VIS

$\#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\})$

$\mathrm{dom}(v? \rhd \{n\}) = SENS$

ReadGearsExtending_FT_4

ReadGearsExtending_DNF_2

$g? = forward$

ReadGearsExtending_FT_5

ReadGearsExtending_DNF_2

$g? = left$

ReadGearsExtending_FT_6

ReadGearsExtending_DNF_2

$g? = right$

ReadGearsExtending_DNF_3

ReadGearsExtending_VIS

$\mathrm{dom}(v? \rhd \{y\}) = SENS$

$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

ReadGearsExtending_FT_7

ReadGearsExtending_DNF_3

$g? = forward$

ReadGearsExtending_FT_8

ReadGearsExtending_DNF_3

$g? = left$

ReadGearsExtending_FT_9

ReadGearsExtending_DNF_3

$g? = right$

ReadGearsExtending_DNF_6

ReadGearsExtending_VIS

$sGExt\, g? = SENS$

$\mathrm{dom}(v? \rhd \{y\}) \subset SENS$

$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

ReadGearsExtending_FT_13

ReadGearsExtending_DNF_6

$g? = forward$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_14} \\ \textit{ReadGearsExtending\_DNF\_6} \\ \hline g? = \textit{left} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_15} \\ \textit{ReadGearsExtending\_DNF\_6} \\ \hline g? = \textit{right} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_DNF\_7} \\ \textit{ReadGearsExtending\_VIS} \\ \hline \textit{sGExt}\, g? = \textit{SENS} \\ \#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\}) \\ \mathrm{dom}(v? \rhd \{n\}) \subset \textit{SENS} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_16} \\ \textit{ReadGearsExtending\_DNF\_7} \\ \hline g? = \textit{forward} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_17} \\ \textit{ReadGearsExtending\_DNF\_7} \\ \hline g? = \textit{left} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_18} \\ \textit{ReadGearsExtending\_DNF\_7} \\ \hline g? = \textit{right} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_DNF\_8} \\ \textit{ReadGearsExtending\_VIS} \\ \hline \textit{sGExt}\, g? = \textit{SENS} \\ \mathrm{dom}(v? \rhd \{y\}) \subset \textit{SENS} \\ \mathrm{dom}(v? \rhd \{n\}) \subset \textit{SENS} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_19} \\ \textit{ReadGearsExtending\_DNF\_8} \\ \hline g? = \textit{forward} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_20} \\ \textit{ReadGearsExtending\_DNF\_8} \\ \hline g? = \textit{left} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_21} \\ \textit{ReadGearsExtending\_DNF\_8} \\ \hline g? = \textit{right} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_DNF\_10} \\ \textit{ReadGearsExtending\_VIS} \\ \hline \textit{sGExt}\, g? \subset \textit{SENS} \\ \mathrm{dom}(\textit{sGExt}\, g? \lhd v? \rhd \{y\}) = \textit{sGExt}\, g? \\ \#(\textit{sGExt}\, g? \lhd v? \rhd \{y\}) > \#(\textit{sGExt}\, g? \lhd v? \rhd \{n\}) \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_25} \\ \textit{ReadGearsExtending\_DNF\_10} \\ \hline g? = \textit{forward} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_26} \\ \textit{ReadGearsExtending\_DNF\_10} \\ \hline g? = \textit{left} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_FT\_27} \\ \textit{ReadGearsExtending\_DNF\_10} \\ \hline g? = \textit{right} \\ \hline\end{array}$$

$$\begin{array}{l}\hline \textit{ReadGearsExtending\_DNF\_11} \\ \textit{ReadGearsExtending\_VIS} \\ \hline \textit{sGExt}\, g? \subset \textit{SENS} \\ \#(\textit{sGExt}\, g? \lhd v? \rhd \{y\}) \leq \#(\textit{sGExt}\, g? \lhd v? \rhd \{n\}) \\ \mathrm{dom}(\textit{sGExt}\, g? \lhd v? \rhd \{n\}) = \textit{sGExt}\, g? \\ \hline\end{array}$$

$ReadGearsExtending\_FT\_28$
$ReadGearsExtending\_DNF\_11$
$g? = forward$

$ReadGearsExtending\_FT\_29$
$ReadGearsExtending\_DNF\_11$
$g? = left$

$ReadGearsExtending\_FT\_30$
$ReadGearsExtending\_DNF\_11$
$g? = right$

$ReadGearsExtending\_DNF\_12$
$ReadGearsExtending\_VIS$
$sGExt\, g? \subset SENS$
$\mathrm{dom}(sGExt\, g? \lhd v? \rhd \{y\}) = sGExt\, g?$
$\mathrm{dom}(sGExt\, g? \lhd v? \rhd \{n\}) = sGExt\, g?$

$ReadGearsExtending\_FT\_31$
$ReadGearsExtending\_DNF\_12$
$g? = forward$

$ReadGearsExtending\_FT\_32$
$ReadGearsExtending\_DNF\_12$
$g? = left$

$ReadGearsExtending\_FT\_33$
$ReadGearsExtending\_DNF\_12$
$g? = right$

$ReadGearsRetracting\_DNF\_2$
$ReadGearsRetracting\_VIS$
$\#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\})$
$\mathrm{dom}(v? \rhd \{n\}) = SENS$

$ReadGearsRetracting\_FT\_4$
$ReadGearsRetracting\_DNF\_2$
$g? = forward$

$ReadGearsRetracting\_FT\_5$
$ReadGearsRetracting\_DNF\_2$
$g? = left$

$ReadGearsRetracting\_FT\_6$
$ReadGearsRetracting\_DNF\_2$
$g? = right$

$ReadGearsRetracting\_DNF\_3$
$ReadGearsRetracting\_VIS$
$\mathrm{dom}(v? \rhd \{y\}) = SENS$
$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

$ReadGearsRetracting\_FT\_7$
$ReadGearsRetracting\_DNF\_3$
$g? = forward$

$ReadGearsRetracting\_FT\_8$
$ReadGearsRetracting\_DNF\_3$
$g? = left$

$ReadGearsRetracting\_FT\_9$
$ReadGearsRetracting\_DNF\_3$
$g? = right$

$ReadGearsRetracting\_DNF\_6$
$ReadGearsRetracting\_VIS$
$sGRec\, g? = SENS$
$\mathrm{dom}(v? \rhd \{y\}) \subset SENS$
$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_13 \\ ReadGearsRetracting\_DNF\_6 \\ \hline g? = forward \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_14 \\ ReadGearsRetracting\_DNF\_6 \\ \hline g? = left \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_15 \\ ReadGearsRetracting\_DNF\_6 \\ \hline g? = right \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_DNF\_7 \\ ReadGearsRetracting\_VIS \\ \hline sGRec\, g? = SENS \\ \#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\}) \\ \mathrm{dom}(v? \rhd \{n\}) \subset SENS \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_16 \\ ReadGearsRetracting\_DNF\_7 \\ \hline g? = forward \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_17 \\ ReadGearsRetracting\_DNF\_7 \\ \hline g? = left \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_18 \\ ReadGearsRetracting\_DNF\_7 \\ \hline g? = right \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_DNF\_8 \\ ReadGearsRetracting\_VIS \\ \hline sGRec\, g? = SENS \\ \mathrm{dom}(v? \rhd \{y\}) \subset SENS \\ \mathrm{dom}(v? \rhd \{n\}) \subset SENS \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_19 \\ ReadGearsRetracting\_DNF\_8 \\ \hline g? = forward \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_20 \\ ReadGearsRetracting\_DNF\_8 \\ \hline g? = left \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_21 \\ ReadGearsRetracting\_DNF\_8 \\ \hline g? = right \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_DNF\_10 \\ ReadGearsRetracting\_VIS \\ \hline sGRec\, g? \subset SENS \\ \mathrm{dom}(sGRec\, g? \lhd v? \rhd \{y\}) = sGRec\, g? \\ \#(sGRec\, g? \lhd v? \rhd \{y\}) > \#(sGRec\, g? \lhd v? \rhd \{n\}) \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_25 \\ ReadGearsRetracting\_DNF\_10 \\ \hline g? = forward \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_26 \\ ReadGearsRetracting\_DNF\_10 \\ \hline g? = left \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_FT\_27 \\ ReadGearsRetracting\_DNF\_10 \\ \hline g? = right \\ \hline\end{array}$$

$$\begin{array}{|l}\hline ReadGearsRetracting\_DNF\_11 \\ ReadGearsRetracting\_VIS \\ \hline sGRec\, g? \subset SENS \\ \#(sGRec\, g? \lhd v? \rhd \{y\}) \leq \#(sGRec\, g? \lhd v? \rhd \{n\}) \\ \mathrm{dom}(sGRec\, g? \lhd v? \rhd \{n\}) = sGRec\, g? \\ \hline\end{array}$$

$ReadGearsRetracting\_FT\_28$
$ReadGearsRetracting\_DNF\_11$
$g? = forward$

$ReadGearsRetracting\_FT\_29$
$ReadGearsRetracting\_DNF\_11$
$g? = left$

$ReadGearsRetracting\_FT\_30$
$ReadGearsRetracting\_DNF\_11$
$g? = right$

$ReadGearsRetracting\_DNF\_12$
$ReadGearsRetracting\_VIS$
$sGRec\, g? \subset SENS$
$\mathrm{dom}(sGRec\, g? \lhd v? \rhd \{y\}) = sGRec\, g?$
$\mathrm{dom}(sGRec\, g? \lhd v? \rhd \{n\}) = sGRec\, g?$

$ReadGearsRetracting\_FT\_31$
$ReadGearsRetracting\_DNF\_12$
$g? = forward$

$ReadGearsRetracting\_FT\_32$
$ReadGearsRetracting\_DNF\_12$
$g? = left$

$ReadGearsRetracting\_FT\_33$
$ReadGearsRetracting\_DNF\_12$
$g? = right$

$HandleNotChanged\_DNF\_1$
$HandleNotChanged\_VIS$
$now - lHPCh = 20$
$lHPCh \neq 0$

$HandleNotChanged\_SP\_4$
$HandleNotChanged\_DNF\_1$
$now > 0$
$lHPCh > 0$
$now < lHPCh$

$HandleNotChanged\_SP\_6$
$HandleNotChanged\_DNF\_1$
$now > 0$
$lHPCh > 0$
$now > lHPCh$

$HandleNotChanged\_DNF\_2$
$HandleNotChanged\_VIS$
$now - lHPCh \neq 20$

$HandleNotChanged\_DNF\_3$
$HandleNotChanged\_VIS$
$lHPCh = 0$

$Up2\_DNF\_1$
$Up2\_VIS$
$st = u1$
$hPos = up$
$200 \leq now - stEV$
$100 \leq now - stDCEV$

$Up2\_SP\_3$
$Up2\_DNF\_1$
$now > 0$
$stEV = 0$

$Up2\_SP\_257$
$Up2\_SP\_3$
$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up2\_SP\_321$

$Up2\_SP\_257$

$now > 0$
$stDCEV = 0$

$Up2\_SP\_351$

$Up2\_SP\_258$

$now > 0$
$stDCEV = 0$

$Up2\_SP\_335$

$Up2\_SP\_321$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Up2\_SP\_365$

$Up2\_SP\_351$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Up2\_SP\_324$

$Up2\_SP\_257$

$now > 0$
$stDCEV > 0$
$now > stDCEV$

$Up2\_SP\_354$

$Up2\_SP\_258$

$now > 0$
$stDCEV > 0$
$now > stDCEV$

$Up2\_SP\_347$

$Up2\_SP\_324$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Up2\_SP\_377$

$Up2\_SP\_354$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Up2\_SP\_348$

$Up2\_SP\_324$

$100 > 0$
$now - stDCEV > 0$
$100 = now - stDCEV$

$Up2\_SP\_378$

$Up2\_SP\_354$

$100 > 0$
$now - stDCEV > 0$
$100 = now - stDCEV$

$Up2\_SP\_258$

$Up2\_SP\_3$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up2\_SP\_6$

$Up2\_DNF\_1$

$now > 0$
$stEV > 0$
$now > stEV$

$Up2\_SP\_629$
$Up2\_SP\_6$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up2\_SP\_693$
$Up2\_SP\_629$

$now > 0$
$stDCEV = 0$

$Up2\_SP\_707$
$Up2\_SP\_693$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Up2\_SP\_696$
$Up2\_SP\_629$

$now > 0$
$stDCEV > 0$
$now > stDCEV$

$Up2\_SP\_719$
$Up2\_SP\_696$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Up2\_SP\_720$
$Up2\_SP\_696$

$100 > 0$
$now - stDCEV > 0$
$100 = now - stDCEV$

$Up2\_SP\_630$
$Up2\_SP\_6$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up2\_SP\_723$
$Up2\_SP\_630$

$now > 0$
$stDCEV = 0$

$Up2\_SP\_737$
$Up2\_SP\_723$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Up2\_SP\_726$
$Up2\_SP\_630$

$now > 0$
$stDCEV > 0$
$now > stDCEV$

$Up2\_SP\_749$
$Up2\_SP\_726$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Up2\_SP\_750$
$Up2\_SP\_726$

$100 > 0$
$now - stDCEV > 0$
$100 = now - stDCEV$

Up2_DNF_2

$Up2\_VIS$

$st = u1$
$hPos = up$
$200 \leq now - stEV$
$stDCEV = 0$

Up2_DNF_3

$Up2\_VIS$

$st = u1$
$hPos = down$

Up2_DNF_4

$Up2\_VIS$

$st \neq u1$

Up2_DNF_5

$Up2\_VIS$

$200 > now - stEV$

Up2_DNF_6

$Up2\_VIS$

$100 > now - stDCEV$
$stDCEV \neq 0$

ReadAnalogicalSwitch_DNF_2

$ReadAnalogicalSwitch\_VIS$

$\#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\})$
$\mathrm{dom}(v? \rhd \{n\}) = SENS$

ReadAnalogicalSwitch_DNF_3

$ReadAnalogicalSwitch\_VIS$

$\mathrm{dom}(v? \rhd \{y\}) = SENS$
$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

ReadAnalogicalSwitch_DNF_6

$ReadAnalogicalSwitch\_VIS$

$sAS = SENS$
$\mathrm{dom}(v? \rhd \{y\}) \subset SENS$
$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

ReadAnalogicalSwitch_DNF_7

$ReadAnalogicalSwitch\_VIS$

$sAS = SENS$
$\#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\})$
$\mathrm{dom}(v? \rhd \{n\}) \subset SENS$

ReadAnalogicalSwitch_DNF_8

$ReadAnalogicalSwitch\_VIS$

$sAS = SENS$
$\mathrm{dom}(v? \rhd \{y\}) \subset SENS$
$\mathrm{dom}(v? \rhd \{n\}) \subset SENS$

ReadAnalogicalSwitch_DNF_10

$ReadAnalogicalSwitch\_VIS$

$sAS \subset SENS$
$\mathrm{dom}(sAS \lhd v? \rhd \{y\}) = sAS$
$\#(sAS \lhd v? \rhd \{y\}) > \#(sAS \lhd v? \rhd \{n\})$

ReadAnalogicalSwitch_DNF_11

$ReadAnalogicalSwitch\_VIS$

$sAS \subset SENS$
$\#(sAS \lhd v? \rhd \{y\}) \leq \#(sAS \lhd v? \rhd \{n\})$
$\mathrm{dom}(sAS \lhd v? \rhd \{n\}) = sAS$

ReadAnalogicalSwitch_DNF_12

$ReadAnalogicalSwitch\_VIS$

$sAS \subset SENS$
$\mathrm{dom}(sAS \lhd v? \rhd \{y\}) = sAS$
$\mathrm{dom}(sAS \lhd v? \rhd \{n\}) = sAS$

$Up1\_DNF\_1$
$Up1\_VIS$
$st = u0$
$hPos = up$
$200 \leq now - stEV$

$Up1\_SP\_3$
$Up1\_DNF\_1$
$now > 0$
$stEV = 0$

$Up1\_SP\_17$
$Up1\_SP\_3$
$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up1\_SP\_18$
$Up1\_SP\_3$
$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up1\_SP\_6$
$Up1\_DNF\_1$
$now > 0$
$stEV > 0$
$now > stEV$

$Up1\_SP\_29$
$Up1\_SP\_6$
$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up1\_SP\_30$
$Up1\_SP\_6$
$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up1\_DNF\_2$
$Up1\_VIS$
$st = u0$
$hPos = up$
$stEV = 0$

$Up1\_DNF\_3$
$Up1\_VIS$
$st \neq u0$

$Up1\_DNF\_4$
$Up1\_VIS$
$hPos \neq up$

$Up1\_DNF\_5$
$Up1\_VIS$
$200 > now - stEV$
$stEV \neq 0$

$Up4\_DNF\_1$
$Up4\_VIS$
$st = u3$
$hPos = up$
$\operatorname{ran} gRec = \{y\}$
$1000 \leq now - spEV$

$Up4\_SP\_3$
$Up4\_DNF\_1$
$now > 0$
$spEV = 0$

$Up4\_SP\_17$
$$\begin{array}{l} Up4\_SP\_3 \\ \hline 1000 > 0 \\ now - spEV > 0 \\ 1000 < now - spEV \end{array}$$

$Up4\_SP\_18$
$$\begin{array}{l} Up4\_SP\_3 \\ \hline 1000 > 0 \\ now - spEV > 0 \\ 1000 = now - spEV \end{array}$$

$Up4\_SP\_6$
$$\begin{array}{l} Up4\_DNF\_1 \\ \hline now > 0 \\ spEV > 0 \\ now > spEV \end{array}$$

$Up4\_SP\_29$
$$\begin{array}{l} Up4\_SP\_6 \\ \hline 1000 > 0 \\ now - spEV > 0 \\ 1000 < now - spEV \end{array}$$

$Up4\_SP\_30$
$$\begin{array}{l} Up4\_SP\_6 \\ \hline 1000 > 0 \\ now - spEV > 0 \\ 1000 = now - spEV \end{array}$$

$Up4\_DNF\_2$
$$\begin{array}{l} Up4\_VIS \\ \hline st = u3 \\ hPos = up \\ \operatorname{ran} gRec = \{y\} \\ spEV = 0 \end{array}$$

$Up4\_DNF\_3$
$$\begin{array}{l} Up4\_VIS \\ \hline st = u3 \\ hPos = down \end{array}$$

$Up4\_DNF\_4$
$$\begin{array}{l} Up4\_VIS \\ \hline st \neq u3 \end{array}$$

$Up4\_DNF\_5$
$$\begin{array}{l} Up4\_VIS \\ \hline \operatorname{ran} gRec \neq \{y\} \end{array}$$

$Up4\_DNF\_6$
$$\begin{array}{l} Up4\_VIS \\ \hline 1000 > now - spEV \\ spEV \neq 0 \end{array}$$

$ReadDoorsOpening\_DNF\_2$
$$\begin{array}{l} ReadDoorsOpening\_VIS \\ \hline \#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\}) \\ \operatorname{dom}(v? \rhd \{n\}) = SENS \end{array}$$

$ReadDoorsOpening\_FT\_4$
$$\begin{array}{l} ReadDoorsOpening\_DNF\_2 \\ \hline g? = forward \end{array}$$

$ReadDoorsOpening\_FT\_5$
$$\begin{array}{l} ReadDoorsOpening\_DNF\_2 \\ \hline g? = left \end{array}$$

$ReadDoorsOpening\_FT\_6$
$$\begin{array}{l} ReadDoorsOpening\_DNF\_2 \\ \hline g? = right \end{array}$$

ReadDoorsOpening_DNF_3

ReadDoorsOpening_VIS

$\mathrm{dom}(v? \rhd \{y\}) = SENS$
$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

ReadDoorsOpening_FT_7

ReadDoorsOpening_DNF_3

$g? = forward$

ReadDoorsOpening_FT_8

ReadDoorsOpening_DNF_3

$g? = left$

ReadDoorsOpening_FT_9

ReadDoorsOpening_DNF_3

$g? = right$

ReadDoorsOpening_DNF_6

ReadDoorsOpening_VIS

$sDOp\ g? = SENS$
$\mathrm{dom}(v? \rhd \{y\}) \subset SENS$
$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

ReadDoorsOpening_FT_13

ReadDoorsOpening_DNF_6

$g? = forward$

ReadDoorsOpening_FT_14

ReadDoorsOpening_DNF_6

$g? = left$

ReadDoorsOpening_FT_15

ReadDoorsOpening_DNF_6

$g? = right$

ReadDoorsOpening_DNF_7

ReadDoorsOpening_VIS

$sDOp\ g? = SENS$
$\#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\})$
$\mathrm{dom}(v? \rhd \{n\}) \subset SENS$

ReadDoorsOpening_FT_16

ReadDoorsOpening_DNF_7

$g? = forward$

ReadDoorsOpening_FT_17

ReadDoorsOpening_DNF_7

$g? = left$

ReadDoorsOpening_FT_18

ReadDoorsOpening_DNF_7

$g? = right$

ReadDoorsOpening_DNF_8

ReadDoorsOpening_VIS

$sDOp\ g? = SENS$
$\mathrm{dom}(v? \rhd \{y\}) \subset SENS$
$\mathrm{dom}(v? \rhd \{n\}) \subset SENS$

ReadDoorsOpening_FT_19

ReadDoorsOpening_DNF_8

$g? = forward$

ReadDoorsOpening_FT_20

ReadDoorsOpening_DNF_8

$g? = left$

ReadDoorsOpening_FT_21

ReadDoorsOpening_DNF_8

$g? = right$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_DNF\_10} \\ \textit{ReadDoorsOpening\_VIS} \\ \hline \textit{sDOp}\ g? \subset \textit{SENS} \\ \mathrm{dom}(\textit{sDOp}\ g? \lhd v? \rhd \{y\}) = \textit{sDOp}\ g? \\ \#(\textit{sDOp}\ g? \lhd v? \rhd \{y\}) > \#(\textit{sDOp}\ g? \lhd v? \rhd \{n\}) \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_25} \\ \textit{ReadDoorsOpening\_DNF\_10} \\ \hline g? = \textit{forward} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_26} \\ \textit{ReadDoorsOpening\_DNF\_10} \\ \hline g? = \textit{left} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_27} \\ \textit{ReadDoorsOpening\_DNF\_10} \\ \hline g? = \textit{right} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_DNF\_11} \\ \textit{ReadDoorsOpening\_VIS} \\ \hline \textit{sDOp}\ g? \subset \textit{SENS} \\ \#(\textit{sDOp}\ g? \lhd v? \rhd \{y\}) \leq \#(\textit{sDOp}\ g? \lhd v? \rhd \{n\}) \\ \mathrm{dom}(\textit{sDOp}\ g? \lhd v? \rhd \{n\}) = \textit{sDOp}\ g? \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_28} \\ \textit{ReadDoorsOpening\_DNF\_11} \\ \hline g? = \textit{forward} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_29} \\ \textit{ReadDoorsOpening\_DNF\_11} \\ \hline g? = \textit{left} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_30} \\ \textit{ReadDoorsOpening\_DNF\_11} \\ \hline g? = \textit{right} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_DNF\_12} \\ \textit{ReadDoorsOpening\_VIS} \\ \hline \textit{sDOp}\ g? \subset \textit{SENS} \\ \mathrm{dom}(\textit{sDOp}\ g? \lhd v? \rhd \{y\}) = \textit{sDOp}\ g? \\ \mathrm{dom}(\textit{sDOp}\ g? \lhd v? \rhd \{n\}) = \textit{sDOp}\ g? \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_31} \\ \textit{ReadDoorsOpening\_DNF\_12} \\ \hline g? = \textit{forward} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_32} \\ \textit{ReadDoorsOpening\_DNF\_12} \\ \hline g? = \textit{left} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ReadDoorsOpening\_FT\_33} \\ \textit{ReadDoorsOpening\_DNF\_12} \\ \hline g? = \textit{right} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ChangeHandle\_DNF\_1} \\ \textit{ChangeHandle\_VIS} \\ \hline \textit{hPos} = \textit{down} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{ChangeHandle\_DNF\_2} \\ \textit{ChangeHandle\_VIS} \\ \hline \textit{hPos} = \textit{up} \\ \hline \end{array}$$

$$\begin{array}{l}\hline \textit{Up3\_DNF\_1} \\ \textit{Up3\_VIS} \\ \hline \textit{st} = \textit{u2} \\ \textit{hPos} = \textit{up} \\ \mathrm{ran}\, \textit{dOp} = \{y\} \\ \mathrm{ran}\, \textit{sa} = \{n\} \\ 200 \leq \textit{now} - \textit{stEV} \\ 100 \leq \textit{now} - \textit{stGEEV} \\ \hline \end{array}$$

$Up3\_SP\_3$
$Up3\_DNF\_1$

$now > 0$
$stGEEV = 0$

$Up3\_SP\_257$
$Up3\_SP\_3$

$100 > 0$
$now - stGEEV > 0$
$100 < now - stGEEV$

$Up3\_SP\_321$
$Up3\_SP\_257$

$now > 0$
$stEV = 0$

$Up3\_SP\_335$
$Up3\_SP\_321$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up3\_SP\_336$
$Up3\_SP\_321$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up3\_SP\_324$
$Up3\_SP\_257$

$now > 0$
$stEV > 0$
$now > stEV$

$Up3\_SP\_347$
$Up3\_SP\_324$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up3\_SP\_348$
$Up3\_SP\_324$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up3\_SP\_6$
$Up3\_DNF\_1$

$now > 0$
$stGEEV > 0$
$now > stGEEV$

$Up3\_SP\_629$
$Up3\_SP\_6$

$100 > 0$
$now - stGEEV > 0$
$100 < now - stGEEV$

$Up3\_SP\_693$
$Up3\_SP\_629$

$now > 0$
$stEV = 0$

$Up3\_SP\_707$
$Up3\_SP\_693$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up3\_SP\_708$
$Up3\_SP\_693$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up3\_SP\_696$

$Up3\_SP\_629$

$now > 0$
$stEV > 0$
$now > stEV$

$Up3\_SP\_719$

$Up3\_SP\_696$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up3\_SP\_720$

$Up3\_SP\_696$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up3\_SP\_630$

$Up3\_SP\_6$

$100 > 0$
$now - stGEEV > 0$
$100 = now - stGEEV$

$Up3\_SP\_723$

$Up3\_SP\_630$

$now > 0$
$stEV = 0$

$Up3\_SP\_737$

$Up3\_SP\_723$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up3\_SP\_738$

$Up3\_SP\_723$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up3\_SP\_726$

$Up3\_SP\_630$

$now > 0$
$stEV > 0$
$now > stEV$

$Up3\_SP\_749$

$Up3\_SP\_726$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up3\_SP\_750$

$Up3\_SP\_726$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up3\_DNF\_2$

$Up3\_VIS$

$st = u2$
$hPos = up$
$\operatorname{ran} dOp = \{y\}$
$\operatorname{ran} sa = \{n\}$
$200 \leq now - stEV$
$stGEEV = 0$

$Up3\_DNF\_3$

$Up3\_VIS$

$st = u2$
$hPos = up$
$\operatorname{ran} dOp = \{y\}$
$\operatorname{ran} sa \neq \{n\}$

$Up3\_DNF\_4$

$Up3\_VIS$

$st = u2$
$hPos = down$

$Up3\_DNF\_5$

$Up3\_VIS$

$st \neq u2$

$Up3\_DNF\_6$

$Up3\_VIS$

$\operatorname{ran} dOp \neq \{y\}$

$Up3\_DNF\_7$

$Up3\_VIS$

$200 > now - stEV$

$Up3\_DNF\_8$

$Up3\_VIS$

$100 > now - stGEEV$
$stGEEV \neq 0$

$Up6\_DNF\_1$

$Up6\_VIS$

$st = u5$
$hPos = up$
$200 \leq now - stEV$
$100 \leq now - stDOEV$

$Up6\_SP\_3$

$Up6\_DNF\_1$

$now > 0$
$stEV = 0$

$Up6\_SP\_257$

$Up6\_SP\_3$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up6\_SP\_321$

$Up6\_SP\_257$

$now > 0$
$stDOEV = 0$

$Up6\_SP\_335$

$Up6\_SP\_321$

$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Up6\_SP\_324$

$Up6\_SP\_257$

$now > 0$
$stDOEV > 0$
$now > stDOEV$

$Up6\_SP\_347$

$Up6\_SP\_324$

$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Up6\_SP\_348$

$Up6\_SP\_324$

$100 > 0$
$now - stDOEV > 0$
$100 = now - stDOEV$

$Up6\_SP\_258$
$Up6\_SP\_3$
$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up6\_SP\_351$
$Up6\_SP\_258$
$now > 0$
$stDOEV = 0$

$Up6\_SP\_365$
$Up6\_SP\_351$
$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Up6\_SP\_354$
$Up6\_SP\_258$
$now > 0$
$stDOEV > 0$
$now > stDOEV$

$Up6\_SP\_377$
$Up6\_SP\_354$
$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Up6\_SP\_378$
$Up6\_SP\_354$
$100 > 0$
$now - stDOEV > 0$
$100 = now - stDOEV$

$Up6\_SP\_6$
$Up6\_DNF\_1$
$now > 0$
$stEV > 0$
$now > stEV$

$Up6\_SP\_629$
$Up6\_SP\_6$
$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Up6\_SP\_693$
$Up6\_SP\_629$
$now > 0$
$stDOEV = 0$

$Up6\_SP\_707$
$Up6\_SP\_693$
$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Up6\_SP\_696$
$Up6\_SP\_629$
$now > 0$
$stDOEV > 0$
$now > stDOEV$

$Up6\_SP\_719$
$Up6\_SP\_696$
$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Up6\_SP\_720$
$Up6\_SP\_696$
$100 > 0$
$now - stDOEV > 0$
$100 = now - stDOEV$

$Up6\_SP\_630$
$Up6\_SP\_6$
$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Up6\_SP\_723$
$Up6\_SP\_630$
$now > 0$
$stDOEV = 0$

$Up6\_SP\_737$
$Up6\_SP\_723$
$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Up6\_SP\_726$
$Up6\_SP\_630$
$now > 0$
$stDOEV > 0$
$now > stDOEV$

$Up6\_SP\_749$
$Up6\_SP\_726$
$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Up6\_SP\_750$
$Up6\_SP\_726$
$100 > 0$
$now - stDOEV > 0$
$100 = now - stDOEV$

$Up6\_DNF\_2$
$Up6\_VIS$
$st = u5$
$hPos = down$

$Up6\_DNF\_3$
$Up6\_VIS$
$st \neq u5$

$Up6\_DNF\_4$
$Up6\_VIS$
$200 > now - stEV$

$Up6\_DNF\_5$
$Up6\_VIS$
$100 > now - stDOEV$

$Up5\_DNF\_1$
$Up5\_VIS$
$st = u4$
$hPos = up$
$1000 \leq now - spEV$

$Up5\_SP\_3$
$Up5\_DNF\_1$
$now > 0$
$spEV = 0$

$Up5\_SP\_17$

$Up5\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Up5\_SP\_18$

$Up5\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Up5\_SP\_6$

$Up5\_DNF\_1$

$now > 0$
$spEV > 0$
$now > spEV$

$Up5\_SP\_29$

$Up5\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Up5\_SP\_30$

$Up5\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Up5\_DNF\_2$

$Up5\_VIS$

$st = u4$
$hPos = down$

$Up5\_DNF\_3$

$Up5\_VIS$

$st \neq u4$

$Up5\_DNF\_4$

$Up5\_VIS$

$1000 > now - spEV$

$Up8\_DNF\_1$

$Up8\_VIS$

$st = u7$
$hPos = up$
$1000 \leq now - spEV$

$Up8\_SP\_3$

$Up8\_DNF\_1$

$now > 0$
$spEV = 0$

$Up8\_SP\_17$

$Up8\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Up8\_SP\_18$

$Up8\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Up8\_SP\_6$

$Up8\_DNF\_1$

$now > 0$
$spEV > 0$
$now > spEV$

Up8_SP_29

$Up8\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

Up8_SP_30

$Up8\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

Up8_DNF_2

$Up8\_VIS$

$st = u7$
$hPos = down$

Up8_DNF_3

$Up8\_VIS$

$st \neq u7$

Up8_DNF_4

$Up8\_VIS$

$1000 > now - spEV$

Up7_DNF_1

$Up7\_VIS$

$st = u6$
$hPos = up$
$\mathrm{ran}\, dCl = \{y\}$
$1000 \leq now - spEV$

Up7_SP_3

$Up7\_DNF\_1$

$now > 0$
$spEV = 0$

Up7_SP_17

$Up7\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

Up7_SP_18

$Up7\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

Up7_SP_6

$Up7\_DNF\_1$

$now > 0$
$spEV > 0$
$now > spEV$

Up7_SP_29

$Up7\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

Up7_SP_30

$Up7\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

Up7_DNF_2

$Up7\_VIS$

$st = u6$
$hPos = down$

$Up7\_DNF\_3$
$Up7\_VIS$
$st \neq u6$

$Up7\_DNF\_4$
$Up7\_VIS$
$\operatorname{ran} dCl \neq \{y\}$

$Up7\_DNF\_5$
$Up7\_VIS$
$1000 > now - spEV$

$GearsManeuvering\_DNF\_1$
$GearsManeuvering\_VIS$
$\operatorname{ran} gExt \neq \{y\}$
$\operatorname{ran} gRec \neq \{y\}$

$GearsManeuvering\_DNF\_2$
$GearsManeuvering\_VIS$
$\operatorname{ran} dCl \neq \{y\}$

$GearsManeuvering\_DNF\_3$
$GearsManeuvering\_VIS$
$\operatorname{ran} gExt = \{y\}$
$\operatorname{ran} dCl = \{y\}$

$GearsManeuvering\_DNF\_4$
$GearsManeuvering\_VIS$
$\operatorname{ran} gRec = \{y\}$
$\operatorname{ran} dCl = \{y\}$

$Down5\_DNF\_1$
$Down5\_VIS$
$st = d4$
$hPos = down$
$1000 \leq now - spEV$

$Down5\_SP\_3$
$Down5\_DNF\_1$
$now > 0$
$spEV = 0$

$Down5\_SP\_17$
$Down5\_SP\_3$
$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Down5\_SP\_18$
$Down5\_SP\_3$
$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Down5\_SP\_6$
$Down5\_DNF\_1$
$now > 0$
$spEV > 0$
$now > spEV$

$Down5\_SP\_29$
$Down5\_SP\_6$
$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Down5\_SP\_30$
$Down5\_SP\_6$
$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Down5\_DNF\_2$
$Down5\_VIS$

$hPos = up$
$st = d4$

$Down5\_DNF\_3$
$Down5\_VIS$

$st \neq d4$

$Down5\_DNF\_4$
$Down5\_VIS$

$1000 > now - spEV$

$Down4\_DNF\_1$
$Down4\_VIS$

$st = d3$
$hPos = down$
$\mathrm{ran}\, gExt = \{y\}$
$1000 \leq now - spEV$

$Down4\_SP\_3$
$Down4\_DNF\_1$

$now > 0$
$spEV = 0$

$Down4\_SP\_17$
$Down4\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Down4\_SP\_18$
$Down4\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Down4\_SP\_6$
$Down4\_DNF\_1$

$now > 0$
$spEV > 0$
$now > spEV$

$Down4\_SP\_29$
$Down4\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Down4\_SP\_30$
$Down4\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Down4\_DNF\_2$
$Down4\_VIS$

$st = d3$
$hPos = down$
$\mathrm{ran}\, gExt = \{y\}$
$spEV = 0$

$Down4\_DNF\_3$
$Down4\_VIS$

$hPos = up$
$st = d3$

$Down4\_DNF\_4$
$Down4\_VIS$

$st \neq d3$

$Down4\_DNF\_5$

$Down4\_VIS$

$\operatorname{ran} gExt \neq \{y\}$

$Down4\_DNF\_6$

$Down4\_VIS$

$1000 > now - spEV$
$spEV \neq 0$

$HydraulicCircuitM\_DNF\_1$

$HydraulicCircuitM\_VIS$

$hc = n$
$2000 \leq now - stGEV$
$stGEV \neq 0$

$HydraulicCircuitM\_SP\_36$

$HydraulicCircuitM\_DNF\_1$

$now > 0$
$stGEV > 0$
$now > stGEV$

$HydraulicCircuitM\_SP\_59$

$HydraulicCircuitM\_SP\_36$

$2000 > 0$
$now - stGEV > 0$
$2000 < now - stGEV$

$HydraulicCircuitM\_SP\_60$

$HydraulicCircuitM\_SP\_36$

$2000 > 0$
$now - stGEV > 0$
$2000 = now - stGEV$

$HydraulicCircuitM\_DNF\_2$

$HydraulicCircuitM\_VIS$

$hc = y$
$10000 \leq now - spGEV$
$spGEV \neq 0$

$HydraulicCircuitM\_SP\_6$

$HydraulicCircuitM\_DNF\_2$

$now > 0$
$spGEV > 0$
$now > spGEV$

$HydraulicCircuitM\_SP\_29$

$HydraulicCircuitM\_SP\_6$

$10000 > 0$
$now - spGEV > 0$
$10000 < now - spGEV$

$HydraulicCircuitM\_SP\_30$

$HydraulicCircuitM\_SP\_6$

$10000 > 0$
$now - spGEV > 0$
$10000 = now - spGEV$

$HydraulicCircuitM\_DNF\_3$

$HydraulicCircuitM\_VIS$

$hc \neq n$

$HydraulicCircuitM\_DNF\_4$

$HydraulicCircuitM\_VIS$

$2000 > now - stGEV$

$HydraulicCircuitM\_DNF\_5$

$HydraulicCircuitM\_VIS$

$stGEV = 0$

$HydraulicCircuitM\_DNF\_6$

$HydraulicCircuitM\_VIS$

$hc \neq y$

$HydraulicCircuitM\_DNF\_7$
$HydraulicCircuitM\_VIS$

$10000 > now - spGEV$

$HydraulicCircuitM\_DNF\_8$
$HydraulicCircuitM\_VIS$

$spGEV = 0$

$Down3\_DNF\_1$
$Down3\_VIS$

$st = d2$
$hPos = down$
$\mathrm{ran}\, dOp = \{y\}$
$200 \leq now - stEV$
$100 \leq now - stGREV$

$Down3\_SP\_3$
$Down3\_DNF\_1$

$now > 0$
$stEV = 0$

$Down3\_SP\_257$
$Down3\_SP\_3$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Down3\_SP\_321$
$Down3\_SP\_257$

$now > 0$
$stGREV = 0$

$Down3\_SP\_335$
$Down3\_SP\_321$

$100 > 0$
$now - stGREV > 0$
$100 < now - stGREV$

$Down3\_SP\_324$
$Down3\_SP\_257$

$now > 0$
$stGREV > 0$
$now > stGREV$

$Down3\_SP\_347$
$Down3\_SP\_324$

$100 > 0$
$now - stGREV > 0$
$100 < now - stGREV$

$Down3\_SP\_348$
$Down3\_SP\_324$

$100 > 0$
$now - stGREV > 0$
$100 = now - stGREV$

$Down3\_SP\_258$
$Down3\_SP\_3$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Down3\_SP\_351$
$Down3\_SP\_258$

$now > 0$
$stGREV = 0$

$Down3\_SP\_365$
$Down3\_SP\_351$

$100 > 0$
$now - stGREV > 0$
$100 < now - stGREV$

$Down3\_SP\_354$
$Down3\_SP\_258$
$now > 0$
$stGREV > 0$
$now > stGREV$

$Down3\_SP\_377$
$Down3\_SP\_354$
$100 > 0$
$now - stGREV > 0$
$100 < now - stGREV$

$Down3\_SP\_378$
$Down3\_SP\_354$
$100 > 0$
$now - stGREV > 0$
$100 = now - stGREV$

$Down3\_SP\_6$
$Down3\_DNF\_1$
$now > 0$
$stEV > 0$
$now > stEV$

$Down3\_SP\_629$
$Down3\_SP\_6$
$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Down3\_SP\_693$
$Down3\_SP\_629$
$now > 0$
$stGREV = 0$

$Down3\_SP\_707$
$Down3\_SP\_693$
$100 > 0$
$now - stGREV > 0$
$100 < now - stGREV$

$Down3\_SP\_696$
$Down3\_SP\_629$
$now > 0$
$stGREV > 0$
$now > stGREV$

$Down3\_SP\_719$
$Down3\_SP\_696$
$100 > 0$
$now - stGREV > 0$
$100 < now - stGREV$

$Down3\_SP\_720$
$Down3\_SP\_696$
$100 > 0$
$now - stGREV > 0$
$100 = now - stGREV$

$Down3\_SP\_630$
$Down3\_SP\_6$
$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Down3\_SP\_723$
$Down3\_SP\_630$
$now > 0$
$stGREV = 0$

$Down3\_SP\_737$

$Down3\_SP\_723$

$100 > 0$
$now - stGREV > 0$
$100 < now - stGREV$

$Down3\_SP\_726$

$Down3\_SP\_630$

$now > 0$
$stGREV > 0$
$now > stGREV$

$Down3\_SP\_749$

$Down3\_SP\_726$

$100 > 0$
$now - stGREV > 0$
$100 < now - stGREV$

$Down3\_SP\_750$

$Down3\_SP\_726$

$100 > 0$
$now - stGREV > 0$
$100 = now - stGREV$

$Down3\_DNF\_2$

$Down3\_VIS$

$st = d2$
$hPos = down$
$\operatorname{ran} dOp = \{y\}$
$200 \leq now - stEV$
$stGREV = 0$

$Down3\_DNF\_3$

$Down3\_VIS$

$hPos = up$
$st = d2$

$Down3\_DNF\_4$

$Down3\_VIS$

$st = d2$

$Down3\_DNF\_5$

$Down3\_VIS$

$\operatorname{ran} dOp \neq \{y\}$

$Down3\_DNF\_6$

$Down3\_VIS$

$200 > now - stEV$

$Down3\_DNF\_7$

$Down3\_VIS$

$100 > now - stGREV$
$stGREV \neq 0$

$Down2\_DNF\_1$

$Down2\_VIS$

$st = d1$
$hPos = down$
$200 \leq now - stEV$
$100 \leq now - stDCEV$

$Down2\_SP\_3$

$Down2\_DNF\_1$

$now > 0$
$stEV = 0$

$Down2\_SP\_257$

$Down2\_SP\_3$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Down2\_SP\_321$

$Down2\_SP\_257$

$now > 0$
$stDCEV = 0$

$Down2\_SP\_335$

$Down2\_SP\_321$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Down2\_SP\_324$

$Down2\_SP\_257$

$now > 0$
$stDCEV > 0$
$now > stDCEV$

$Down2\_SP\_347$

$Down2\_SP\_324$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Down2\_SP\_348$

$Down2\_SP\_324$

$100 > 0$
$now - stDCEV > 0$
$100 = now - stDCEV$

$Down2\_SP\_258$

$Down2\_SP\_3$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Down2\_SP\_351$

$Down2\_SP\_258$

$now > 0$
$stDCEV = 0$

$Down2\_SP\_365$

$Down2\_SP\_351$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Down2\_SP\_354$

$Down2\_SP\_258$

$now > 0$
$stDCEV > 0$
$now > stDCEV$

$Down2\_SP\_377$

$Down2\_SP\_354$

$100 > 0$
$now - stDCEV > 0$
$100 < now - stDCEV$

$Down2\_SP\_378$

$Down2\_SP\_354$

$100 > 0$
$now - stDCEV > 0$
$100 = now - stDCEV$

$Down2\_SP\_6$

$Down2\_DNF\_1$

$now > 0$
$stEV > 0$
$now > stEV$

Down2_SP_629

$Down2\_SP\_6$

$$200 > 0$$
$$now - stEV > 0$$
$$200 < now - stEV$$

Down2_SP_693

$Down2\_SP\_629$

$$now > 0$$
$$stDCEV = 0$$

Down2_SP_707

$Down2\_SP\_693$

$$100 > 0$$
$$now - stDCEV > 0$$
$$100 < now - stDCEV$$

Down2_SP_696

$Down2\_SP\_629$

$$now > 0$$
$$stDCEV > 0$$
$$now > stDCEV$$

Down2_SP_719

$Down2\_SP\_696$

$$100 > 0$$
$$now - stDCEV > 0$$
$$100 < now - stDCEV$$

Down2_SP_720

$Down2\_SP\_696$

$$100 > 0$$
$$now - stDCEV > 0$$
$$100 = now - stDCEV$$

Down2_SP_630

$Down2\_SP\_6$

$$200 > 0$$
$$now - stEV > 0$$
$$200 = now - stEV$$

Down2_SP_723

$Down2\_SP\_630$

$$now > 0$$
$$stDCEV = 0$$

Down2_SP_737

$Down2\_SP\_723$

$$100 > 0$$
$$now - stDCEV > 0$$
$$100 < now - stDCEV$$

Down2_SP_726

$Down2\_SP\_630$

$$now > 0$$
$$stDCEV > 0$$
$$now > stDCEV$$

Down2_SP_749

$Down2\_SP\_726$

$$100 > 0$$
$$now - stDCEV > 0$$
$$100 < now - stDCEV$$

Down2_SP_750

$Down2\_SP\_726$

$$100 > 0$$
$$now - stDCEV > 0$$
$$100 = now - stDCEV$$

$$
\begin{array}{l}
\textit{Down2\_DNF\_2} \\
\hline
\textit{Down2\_VIS} \\
\hline
st = d1 \\
hPos = down \\
200 \leq now - stEV \\
stDCEV = 0
\end{array}
$$

$$
\begin{array}{l}
\textit{Down2\_DNF\_3} \\
\hline
\textit{Down2\_VIS} \\
\hline
hPos = up \\
st = d1
\end{array}
$$

$$
\begin{array}{l}
\textit{Down2\_DNF\_4} \\
\hline
\textit{Down2\_VIS} \\
\hline
st \neq d1
\end{array}
$$

$$
\begin{array}{l}
\textit{Down2\_DNF\_5} \\
\hline
\textit{Down2\_VIS} \\
\hline
200 > now - stEV
\end{array}
$$

$$
\begin{array}{l}
\textit{Down2\_DNF\_6} \\
\hline
\textit{Down2\_VIS} \\
\hline
100 > now - stDCEV \\
stDCEV \neq 0
\end{array}
$$

$$
\begin{array}{l}
\textit{GearsLockedDown\_DNF\_1} \\
\hline
\textit{GearsLockedDown\_VIS} \\
\hline
\operatorname{ran} gExt = \{y\}
\end{array}
$$

$$
\begin{array}{l}
\textit{GearsLockedDown\_DNF\_2} \\
\hline
\textit{GearsLockedDown\_VIS} \\
\hline
\operatorname{ran} gExt \neq \{y\}
\end{array}
$$

$$
\begin{array}{l}
\textit{Down8\_DNF\_1} \\
\hline
\textit{Down8\_VIS} \\
\hline
st = d7 \\
hPos = down \\
1000 \leq now - spEV
\end{array}
$$

$$
\begin{array}{l}
\textit{Down8\_SP\_3} \\
\hline
\textit{Down8\_DNF\_1} \\
\hline
now > 0 \\
spEV = 0
\end{array}
$$

$$
\begin{array}{l}
\textit{Down8\_SP\_17} \\
\hline
\textit{Down8\_SP\_3} \\
\hline
1000 > 0 \\
now - spEV > 0 \\
1000 < now - spEV
\end{array}
$$

$$
\begin{array}{l}
\textit{Down8\_SP\_18} \\
\hline
\textit{Down8\_SP\_3} \\
\hline
1000 > 0 \\
now - spEV > 0 \\
1000 = now - spEV
\end{array}
$$

$$
\begin{array}{l}
\textit{Down8\_SP\_6} \\
\hline
\textit{Down8\_DNF\_1} \\
\hline
now > 0 \\
spEV > 0 \\
now > spEV
\end{array}
$$

$$
\begin{array}{l}
\textit{Down8\_SP\_29} \\
\hline
\textit{Down8\_SP\_6} \\
\hline
1000 > 0 \\
now - spEV > 0 \\
1000 < now - spEV
\end{array}
$$

$$\begin{array}{l} \textbf{Down8\_SP\_30} \\ \hline Down8\_SP\_6 \\ \hline 1000 > 0 \\ now - spEV > 0 \\ 1000 = now - spEV \end{array}$$

$$\begin{array}{l} \textbf{Down8\_DNF\_2} \\ \hline Down8\_VIS \\ \hline hPos = up \\ st = d7 \end{array}$$

$$\begin{array}{l} \textbf{Down8\_DNF\_3} \\ \hline Down8\_VIS \\ \hline st \neq d7 \end{array}$$

$$\begin{array}{l} \textbf{Down8\_DNF\_4} \\ \hline Down8\_VIS \\ \hline 1000 > now - spEV \end{array}$$

$$\begin{array}{l} \textbf{ReadHydraulicCircuit\_DNF\_2} \\ \hline ReadHydraulicCircuit\_VIS \\ \hline \#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\}) \\ \mathrm{dom}(v? \rhd \{n\}) = SENS \end{array}$$

$$\begin{array}{l} \textbf{ReadHydraulicCircuit\_DNF\_3} \\ \hline ReadHydraulicCircuit\_VIS \\ \hline \mathrm{dom}(v? \rhd \{y\}) = SENS \\ \#(v? \rhd \{y\}) > \#(v? \rhd \{n\}) \end{array}$$

$$\begin{array}{l} \textbf{ReadHydraulicCircuit\_DNF\_6} \\ \hline ReadHydraulicCircuit\_VIS \\ \hline sHC = SENS \\ \mathrm{dom}(v? \rhd \{y\}) \subset SENS \\ \#(v? \rhd \{y\}) > \#(v? \rhd \{n\}) \end{array}$$

$$\begin{array}{l} \textbf{ReadHydraulicCircuit\_DNF\_7} \\ \hline ReadHydraulicCircuit\_VIS \\ \hline sHC = SENS \\ \#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\}) \\ \mathrm{dom}(v? \rhd \{n\}) \subset SENS \end{array}$$

$$\begin{array}{l} \textbf{ReadHydraulicCircuit\_DNF\_8} \\ \hline ReadHydraulicCircuit\_VIS \\ \hline sHC = SENS \\ \mathrm{dom}(v? \rhd \{y\}) \subset SENS \\ \mathrm{dom}(v? \rhd \{n\}) \subset SENS \end{array}$$

$$\begin{array}{l} \textbf{ReadHydraulicCircuit\_DNF\_10} \\ \hline ReadHydraulicCircuit\_VIS \\ \hline sHC \subset SENS \\ \mathrm{dom}(sHC \lhd v? \rhd \{y\}) = sHC \\ \#(sHC \lhd v? \rhd \{y\}) > \#(sHC \lhd v? \rhd \{n\}) \end{array}$$

$$\begin{array}{l} \textbf{ReadHydraulicCircuit\_DNF\_11} \\ \hline ReadHydraulicCircuit\_VIS \\ \hline sHC \subset SENS \\ \#(sHC \lhd v? \rhd \{y\}) \leq \#(sHC \lhd v? \rhd \{n\}) \\ \mathrm{dom}(sHC \lhd v? \rhd \{n\}) = sHC \end{array}$$

$$\begin{array}{l} \textbf{ReadHydraulicCircuit\_DNF\_12} \\ \hline ReadHydraulicCircuit\_VIS \\ \hline sHC \subset SENS \\ \mathrm{dom}(sHC \lhd v? \rhd \{y\}) = sHC \\ \mathrm{dom}(sHC \lhd v? \rhd \{n\}) = sHC \end{array}$$

$$\begin{array}{l} \textbf{ReadDoorsClosing\_DNF\_2} \\ \hline ReadDoorsClosing\_VIS \\ \hline \#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\}) \\ \mathrm{dom}(v? \rhd \{n\}) = SENS \end{array}$$

$$\begin{array}{l} \textbf{ReadDoorsClosing\_FT\_4} \\ \hline ReadDoorsClosing\_DNF\_2 \\ \hline g? = forward \end{array}$$

$ReadDoorsClosing\_FT\_5$
$$ReadDoorsClosing\_DNF\_2$$
$$g? = left$$

$ReadDoorsClosing\_FT\_6$
$$ReadDoorsClosing\_DNF\_2$$
$$g? = right$$

$ReadDoorsClosing\_DNF\_3$
$$ReadDoorsClosing\_VIS$$
$$\mathrm{dom}(v? \rhd \{y\}) = SENS$$
$$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$$

$ReadDoorsClosing\_FT\_7$
$$ReadDoorsClosing\_DNF\_3$$
$$g? = forward$$

$ReadDoorsClosing\_FT\_8$
$$ReadDoorsClosing\_DNF\_3$$
$$g? = left$$

$ReadDoorsClosing\_FT\_9$
$$ReadDoorsClosing\_DNF\_3$$
$$g? = right$$

$ReadDoorsClosing\_DNF\_6$
$$ReadDoorsClosing\_VIS$$
$$sDCl\, g? = SENS$$
$$\mathrm{dom}(v? \rhd \{y\}) \subset SENS$$
$$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$$

$ReadDoorsClosing\_FT\_13$
$$ReadDoorsClosing\_DNF\_6$$
$$g? = forward$$

$ReadDoorsClosing\_FT\_14$
$$ReadDoorsClosing\_DNF\_6$$
$$g? = left$$

$ReadDoorsClosing\_FT\_15$
$$ReadDoorsClosing\_DNF\_6$$
$$g? = right$$

$ReadDoorsClosing\_DNF\_7$
$$ReadDoorsClosing\_VIS$$
$$sDCl\, g? = SENS$$
$$\#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\})$$
$$\mathrm{dom}(v? \rhd \{n\}) \subset SENS$$

$ReadDoorsClosing\_FT\_16$
$$ReadDoorsClosing\_DNF\_7$$
$$g? = forward$$

$ReadDoorsClosing\_FT\_17$
$$ReadDoorsClosing\_DNF\_7$$
$$g? = left$$

$ReadDoorsClosing\_FT\_18$
$$ReadDoorsClosing\_DNF\_7$$
$$g? = right$$

$ReadDoorsClosing\_DNF\_8$
$$ReadDoorsClosing\_VIS$$
$$sDCl\, g? = SENS$$
$$\mathrm{dom}(v? \rhd \{y\}) \subset SENS$$
$$\mathrm{dom}(v? \rhd \{n\}) \subset SENS$$

$ReadDoorsClosing\_FT\_19$
$$ReadDoorsClosing\_DNF\_8$$
$$g? = forward$$

$ReadDoorsClosing\_FT\_20$
$ReadDoorsClosing\_DNF\_8$
$g? = left$

$ReadDoorsClosing\_FT\_21$
$ReadDoorsClosing\_DNF\_8$
$g? = right$

$ReadDoorsClosing\_DNF\_10$
$ReadDoorsClosing\_VIS$
$sDCl\, g? \subset SENS$
$\mathrm{dom}(sDCl\, g? \lhd v? \rhd \{y\}) = sDCl\, g?$
$\#(sDCl\, g? \lhd v? \rhd \{y\}) > \#(sDCl\, g? \lhd v? \rhd \{n\})$

$ReadDoorsClosing\_FT\_25$
$ReadDoorsClosing\_DNF\_10$
$g? = forward$

$ReadDoorsClosing\_FT\_26$
$ReadDoorsClosing\_DNF\_10$
$g? = left$

$ReadDoorsClosing\_FT\_27$
$ReadDoorsClosing\_DNF\_10$
$g? = right$

$ReadDoorsClosing\_DNF\_11$
$ReadDoorsClosing\_VIS$
$sDCl\, g? \subset SENS$
$\#(sDCl\, g? \lhd v? \rhd \{y\}) \leq \#(sDCl\, g? \lhd v? \rhd \{n\})$
$\mathrm{dom}(sDCl\, g? \lhd v? \rhd \{n\}) = sDCl\, g?$

$ReadDoorsClosing\_FT\_28$
$ReadDoorsClosing\_DNF\_11$
$g? = forward$

$ReadDoorsClosing\_FT\_29$
$ReadDoorsClosing\_DNF\_11$
$g? = left$

$ReadDoorsClosing\_FT\_30$
$ReadDoorsClosing\_DNF\_11$
$g? = right$

$ReadDoorsClosing\_DNF\_12$
$ReadDoorsClosing\_VIS$
$sDCl\, g? \subset SENS$
$\mathrm{dom}(sDCl\, g? \lhd v? \rhd \{y\}) = sDCl\, g?$
$\mathrm{dom}(sDCl\, g? \lhd v? \rhd \{n\}) = sDCl\, g?$

$ReadDoorsClosing\_FT\_31$
$ReadDoorsClosing\_DNF\_12$
$g? = forward$

$ReadDoorsClosing\_FT\_32$
$ReadDoorsClosing\_DNF\_12$
$g? = left$

$ReadDoorsClosing\_FT\_33$
$ReadDoorsClosing\_DNF\_12$
$g? = right$

$Down7\_DNF\_1$
$Down7\_VIS$
$st = d6$
$hPos = down$
$\mathrm{ran}\, dCl = \{y\}$
$1000 \leq now - spEV$

$Down7\_SP\_3$

$Down7\_DNF\_1$

$now > 0$
$spEV = 0$

$Down7\_SP\_17$

$Down7\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Down7\_SP\_18$

$Down7\_SP\_3$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Down7\_SP\_6$

$Down7\_DNF\_1$

$now > 0$
$spEV > 0$
$now > spEV$

$Down7\_SP\_29$

$Down7\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 < now - spEV$

$Down7\_SP\_30$

$Down7\_SP\_6$

$1000 > 0$
$now - spEV > 0$
$1000 = now - spEV$

$Down7\_DNF\_2$

$Down7\_VIS$

$hPos = up$
$st = d6$

$Down7\_DNF\_3$

$Down7\_VIS$

$st \neq d6$

$Down7\_DNF\_4$

$Down7\_VIS$

$\operatorname{ran} dCl \neq \{y\}$

$Down7\_DNF\_5$

$Down7\_VIS$

$1000 > now - spEV$

$Down6\_DNF\_1$

$Down6\_VIS$

$st = d5$
$hPos = down$
$200 \leq now - stEV$
$100 \leq now - stDOEV$

$Down6\_SP\_3$

$Down6\_DNF\_1$

$now > 0$
$stEV = 0$

$Down6\_SP\_257$

$Down6\_SP\_3$

$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Down6\_SP\_321$
$Down6\_SP\_257$

$now > 0$
$stDOEV = 0$

$Down6\_SP\_335$
$Down6\_SP\_321$

$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Down6\_SP\_324$
$Down6\_SP\_257$

$now > 0$
$stDOEV > 0$
$now > stDOEV$

$Down6\_SP\_347$
$Down6\_SP\_324$

$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Down6\_SP\_348$
$Down6\_SP\_324$

$100 > 0$
$now - stDOEV > 0$
$100 = now - stDOEV$

$Down6\_SP\_258$
$Down6\_SP\_3$

$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Down6\_SP\_351$
$Down6\_SP\_258$

$now > 0$
$stDOEV = 0$

$Down6\_SP\_365$
$Down6\_SP\_351$

$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Down6\_SP\_354$
$Down6\_SP\_258$

$now > 0$
$stDOEV > 0$
$now > stDOEV$

$Down6\_SP\_377$
$Down6\_SP\_354$

$100 > 0$
$now - stDOEV > 0$
$100 < now - stDOEV$

$Down6\_SP\_378$
$Down6\_SP\_354$

$100 > 0$
$now - stDOEV > 0$
$100 = now - stDOEV$

$Down6\_SP\_6$
$Down6\_DNF\_1$

$now > 0$
$stEV > 0$
$now > stEV$

$$\begin{array}{l} \hline Down6\_SP\_629 \\ \hline Down6\_SP\_6 \\ \hline 200 > 0 \\ now - stEV > 0 \\ 200 < now - stEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_693 \\ \hline Down6\_SP\_629 \\ \hline now > 0 \\ stDOEV = 0 \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_707 \\ \hline Down6\_SP\_693 \\ \hline 100 > 0 \\ now - stDOEV > 0 \\ 100 < now - stDOEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_696 \\ \hline Down6\_SP\_629 \\ \hline now > 0 \\ stDOEV > 0 \\ now > stDOEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_719 \\ \hline Down6\_SP\_696 \\ \hline 100 > 0 \\ now - stDOEV > 0 \\ 100 < now - stDOEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_720 \\ \hline Down6\_SP\_696 \\ \hline 100 > 0 \\ now - stDOEV > 0 \\ 100 = now - stDOEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_630 \\ \hline Down6\_SP\_6 \\ \hline 200 > 0 \\ now - stEV > 0 \\ 200 = now - stEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_723 \\ \hline Down6\_SP\_630 \\ \hline now > 0 \\ stDOEV = 0 \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_737 \\ \hline Down6\_SP\_723 \\ \hline 100 > 0 \\ now - stDOEV > 0 \\ 100 < now - stDOEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_726 \\ \hline Down6\_SP\_630 \\ \hline now > 0 \\ stDOEV > 0 \\ now > stDOEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_749 \\ \hline Down6\_SP\_726 \\ \hline 100 > 0 \\ now - stDOEV > 0 \\ 100 < now - stDOEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_SP\_750 \\ \hline Down6\_SP\_726 \\ \hline 100 > 0 \\ now - stDOEV > 0 \\ 100 = now - stDOEV \\ \hline \end{array}$$

$$\begin{array}{l} \hline Down6\_DNF\_2 \\ \hline Down6\_VIS \\ \hline hPos = up \\ st = d5 \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{Down6\_DNF\_3} \\ \hline Down6\_VIS \\ \hline st \neq d5 \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{Down6\_DNF\_4} \\ \hline Down6\_VIS \\ \hline 200 > now - stEV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{Down6\_DNF\_5} \\ \hline Down6\_VIS \\ \hline 100 > now - stDOEV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_DNF\_1} \\ \hline GearsMotionM\_VIS \\ \hline \operatorname{ran} gRec \neq \{n\} \\ 7000 \leq now - stGREV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_33} \\ \hline GearsMotionM\_DNF\_1 \\ \hline now > 0 \\ stGREV = 0 \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_47} \\ \hline GearsMotionM\_SP\_33 \\ \hline 7000 > 0 \\ now - stGREV > 0 \\ 7000 < now - stGREV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_48} \\ \hline GearsMotionM\_SP\_33 \\ \hline 7000 > 0 \\ now - stGREV > 0 \\ 7000 = now - stGREV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_36} \\ \hline GearsMotionM\_DNF\_1 \\ \hline now > 0 \\ stGREV > 0 \\ now > stGREV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_59} \\ \hline GearsMotionM\_SP\_36 \\ \hline 7000 > 0 \\ now - stGREV > 0 \\ 7000 < now - stGREV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_60} \\ \hline GearsMotionM\_SP\_36 \\ \hline 7000 > 0 \\ now - stGREV > 0 \\ 7000 = now - stGREV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_DNF\_2} \\ \hline GearsMotionM\_VIS \\ \hline \operatorname{ran} gRec \neq \{y\} \\ 10000 \leq now - stGREV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_3} \\ \hline GearsMotionM\_DNF\_2 \\ \hline now > 0 \\ stGREV = 0 \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_17} \\ \hline GearsMotionM\_SP\_3 \\ \hline 10000 > 0 \\ now - stGREV > 0 \\ 10000 < now - stGREV \\ \hline \end{array}$$

$$\begin{array}{|l} \multicolumn{1}{l}{GearsMotionM\_SP\_18} \\ \hline GearsMotionM\_SP\_3 \\ \hline 10000 > 0 \\ now - stGREV > 0 \\ 10000 = now - stGREV \\ \hline \end{array}$$

$GearsMotionM\_SP\_6$
$GearsMotionM\_DNF\_2$
$now > 0$
$stGREV > 0$
$now > stGREV$

$GearsMotionM\_SP\_29$
$GearsMotionM\_SP\_6$
$10000 > 0$
$now - stGREV > 0$
$10000 < now - stGREV$

$GearsMotionM\_SP\_30$
$GearsMotionM\_SP\_6$
$10000 > 0$
$now - stGREV > 0$
$10000 = now - stGREV$

$GearsMotionM\_DNF\_3$
$GearsMotionM\_VIS$
$\operatorname{ran} gExt \neq \{n\}$
$7000 \leq now - stGEEV$

$GearsMotionM\_SP\_93$
$GearsMotionM\_DNF\_3$
$now > 0$
$stGEEV = 0$

$GearsMotionM\_SP\_107$
$GearsMotionM\_SP\_93$
$7000 > 0$
$now - stGEEV > 0$
$7000 < now - stGEEV$

$GearsMotionM\_SP\_108$
$GearsMotionM\_SP\_93$
$7000 > 0$
$now - stGEEV > 0$
$7000 = now - stGEEV$

$GearsMotionM\_SP\_96$
$GearsMotionM\_DNF\_3$
$now > 0$
$stGEEV > 0$
$now > stGEEV$

$GearsMotionM\_SP\_119$
$GearsMotionM\_SP\_96$
$7000 > 0$
$now - stGEEV > 0$
$7000 < now - stGEEV$

$GearsMotionM\_SP\_120$
$GearsMotionM\_SP\_96$
$7000 > 0$
$now - stGEEV > 0$
$7000 = now - stGEEV$

$GearsMotionM\_DNF\_4$
$GearsMotionM\_VIS$
$\operatorname{ran} gExt \neq \{y\}$
$10000 \leq now - stGEEV$

$GearsMotionM\_SP\_63$
$GearsMotionM\_DNF\_4$
$now > 0$
$stGEEV = 0$

$GearsMotionM\_SP\_77$
$GearsMotionM\_SP\_63$
$10000 > 0$
$now - stGEEV > 0$
$10000 < now - stGEEV$

$GearsMotionM\_SP\_78$

$GearsMotionM\_SP\_63$

$10000 > 0$
$now - stGEEV > 0$
$10000 = now - stGEEV$

$GearsMotionM\_SP\_66$

$GearsMotionM\_DNF\_4$

$now > 0$
$stGEEV > 0$
$now > stGEEV$

$GearsMotionM\_SP\_89$

$GearsMotionM\_SP\_66$

$10000 > 0$
$now - stGEEV > 0$
$10000 < now - stGEEV$

$GearsMotionM\_SP\_90$

$GearsMotionM\_SP\_66$

$10000 > 0$
$now - stGEEV > 0$
$10000 = now - stGEEV$

$GearsMotionM\_DNF\_5$

$GearsMotionM\_VIS$

$\mathrm{ran}\, gRec = \{n\}$

$GearsMotionM\_DNF\_6$

$GearsMotionM\_VIS$

$7000 > now - stGREV$

$GearsMotionM\_DNF\_7$

$GearsMotionM\_VIS$

$\mathrm{ran}\, gRec = \{y\}$

$GearsMotionM\_DNF\_8$

$GearsMotionM\_VIS$

$10000 > now - stGREV$

$GearsMotionM\_DNF\_9$

$GearsMotionM\_VIS$

$\mathrm{ran}\, gExt = \{n\}$

$GearsMotionM\_DNF\_10$

$GearsMotionM\_VIS$

$7000 > now - stGEEV$

$GearsMotionM\_DNF\_11$

$GearsMotionM\_VIS$

$\mathrm{ran}\, gExt = \{y\}$

$GearsMotionM\_DNF\_12$

$GearsMotionM\_VIS$

$10000 > now - stGEEV$

$DoorsMotionM\_DNF\_1$

$DoorsMotionM\_VIS$

$\mathrm{ran}\, dCl \neq \{n\}$
$7000 \leq now - stDOEV$

$DoorsMotionM\_SP\_63$

$DoorsMotionM\_DNF\_1$

$now > 0$
$stDOEV = 0$

$DoorsMotionM\_SP\_77$

$DoorsMotionM\_SP\_63$

$7000 > 0$
$now - stDOEV > 0$
$7000 < now - stDOEV$

DoorsMotionM_SP_78

DoorsMotionM_SP_63

$7000 > 0$
$now - stDOEV > 0$
$7000 = now - stDOEV$

DoorsMotionM_SP_66

DoorsMotionM_DNF_1

$now > 0$
$stDOEV > 0$
$now > stDOEV$

DoorsMotionM_SP_89

DoorsMotionM_SP_66

$7000 > 0$
$now - stDOEV > 0$
$7000 < now - stDOEV$

DoorsMotionM_SP_90

DoorsMotionM_SP_66

$7000 > 0$
$now - stDOEV > 0$
$7000 = now - stDOEV$

DoorsMotionM_DNF_2

DoorsMotionM_VIS

$\operatorname{ran} dOp \neq \{y\}$
$7000 \leq now - stDOEV$

DoorsMotionM_SP_33

DoorsMotionM_DNF_2

$now > 0$
$stDOEV = 0$

DoorsMotionM_SP_47

DoorsMotionM_SP_33

$7000 > 0$
$now - stDOEV > 0$
$7000 < now - stDOEV$

DoorsMotionM_SP_48

DoorsMotionM_SP_33

$7000 > 0$
$now - stDOEV > 0$
$7000 = now - stDOEV$

DoorsMotionM_SP_36

DoorsMotionM_DNF_2

$now > 0$
$stDOEV > 0$
$now > stDOEV$

DoorsMotionM_SP_59

DoorsMotionM_SP_36

$7000 > 0$
$now - stDOEV > 0$
$7000 < now - stDOEV$

DoorsMotionM_SP_60

DoorsMotionM_SP_36

$7000 > 0$
$now - stDOEV > 0$
$7000 = now - stDOEV$

DoorsMotionM_DNF_3

DoorsMotionM_VIS

$\operatorname{ran} dOp \neq \{n\}$
$7000 \leq now - stDCEV$

DoorsMotionM_SP_3

DoorsMotionM_DNF_3

$now > 0$
$stDCEV = 0$

DoorsMotionM_SP_17

DoorsMotionM_SP_3

$7000 > 0$
$now - stDCEV > 0$
$7000 < now - stDCEV$

DoorsMotionM_SP_18

DoorsMotionM_SP_3

$7000 > 0$
$now - stDCEV > 0$
$7000 = now - stDCEV$

DoorsMotionM_SP_6

DoorsMotionM_DNF_3

$now > 0$
$stDCEV > 0$
$now > stDCEV$

DoorsMotionM_SP_29

DoorsMotionM_SP_6

$7000 > 0$
$now - stDCEV > 0$
$7000 < now - stDCEV$

DoorsMotionM_SP_30

DoorsMotionM_SP_6

$7000 > 0$
$now - stDCEV > 0$
$7000 = now - stDCEV$

DoorsMotionM_DNF_4

DoorsMotionM_VIS

$\mathrm{ran}\, dCl \neq \{y\}$
$7000 \leq now - stDCEV$

DoorsMotionM_SP_93

DoorsMotionM_DNF_4

$now > 0$
$stDCEV = 0$

DoorsMotionM_SP_107

DoorsMotionM_SP_93

$7000 > 0$
$now - stDCEV > 0$
$7000 < now - stDCEV$

DoorsMotionM_SP_108

DoorsMotionM_SP_93

$7000 > 0$
$now - stDCEV > 0$
$7000 = now - stDCEV$

DoorsMotionM_SP_96

DoorsMotionM_DNF_4

$now > 0$
$stDCEV > 0$
$now > stDCEV$

DoorsMotionM_SP_119

DoorsMotionM_SP_96

$7000 > 0$
$now - stDCEV > 0$
$7000 < now - stDCEV$

DoorsMotionM_SP_120

DoorsMotionM_SP_96

$7000 > 0$
$now - stDCEV > 0$
$7000 = now - stDCEV$

DoorsMotionM_DNF_5

DoorsMotionM_VIS

$\mathrm{ran}\, dCl = \{n\}$

$DoorsMotionM\_DNF\_6$
$DoorsMotionM\_VIS$
$7000 > now - stDOEV$

$DoorsMotionM\_DNF\_7$
$DoorsMotionM\_VIS$
$\operatorname{ran} dOp = \{y\}$

$DoorsMotionM\_DNF\_8$
$DoorsMotionM\_VIS$
$\operatorname{ran} dOp = \{n\}$

$DoorsMotionM\_DNF\_9$
$DoorsMotionM\_VIS$
$7000 > now - stDCEV$

$DoorsMotionM\_DNF\_10$
$DoorsMotionM\_VIS$
$\operatorname{ran} dCl = \{y\}$

$ReadShockAbsorbers\_DNF\_2$
$ReadShockAbsorbers\_VIS$
$\#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\})$
$\operatorname{dom}(v? \rhd \{n\}) = SENS$

$ReadShockAbsorbers\_FT\_4$
$ReadShockAbsorbers\_DNF\_2$
$g? = forward$

$ReadShockAbsorbers\_FT\_5$
$ReadShockAbsorbers\_DNF\_2$
$g? = left$

$ReadShockAbsorbers\_FT\_6$
$ReadShockAbsorbers\_DNF\_2$
$g? = right$

$ReadShockAbsorbers\_DNF\_3$
$ReadShockAbsorbers\_VIS$
$\operatorname{dom}(v? \rhd \{y\}) = SENS$
$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

$ReadShockAbsorbers\_FT\_7$
$ReadShockAbsorbers\_DNF\_3$
$g? = forward$

$ReadShockAbsorbers\_FT\_8$
$ReadShockAbsorbers\_DNF\_3$
$g? = left$

$ReadShockAbsorbers\_FT\_9$
$ReadShockAbsorbers\_DNF\_3$
$g? = right$

$ReadShockAbsorbers\_DNF\_6$
$ReadShockAbsorbers\_VIS$
$sSA\, g? = SENS$
$\operatorname{dom}(v? \rhd \{y\}) \subset SENS$
$\#(v? \rhd \{y\}) > \#(v? \rhd \{n\})$

$ReadShockAbsorbers\_FT\_13$
$ReadShockAbsorbers\_DNF\_6$
$g? = forward$

$ReadShockAbsorbers\_FT\_14$
$ReadShockAbsorbers\_DNF\_6$
$g? = left$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_15} \\ \hline \textit{ReadShockAbsorbers\_DNF\_6} \\ \hline g? = \textit{right} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_DNF\_7} \\ \hline \textit{ReadShockAbsorbers\_VIS} \\ \hline \textit{sSA}\ g? = \textit{SENS} \\ \#(v? \rhd \{y\}) \leq \#(v? \rhd \{n\}) \\ \mathrm{dom}(v? \rhd \{n\}) \subset \textit{SENS} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_16} \\ \hline \textit{ReadShockAbsorbers\_DNF\_7} \\ \hline g? = \textit{forward} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_17} \\ \hline \textit{ReadShockAbsorbers\_DNF\_7} \\ \hline g? = \textit{left} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_18} \\ \hline \textit{ReadShockAbsorbers\_DNF\_7} \\ \hline g? = \textit{right} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_DNF\_8} \\ \hline \textit{ReadShockAbsorbers\_VIS} \\ \hline \textit{sSA}\ g? = \textit{SENS} \\ \mathrm{dom}(v? \rhd \{y\}) \subset \textit{SENS} \\ \mathrm{dom}(v? \rhd \{n\}) \subset \textit{SENS} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_19} \\ \hline \textit{ReadShockAbsorbers\_DNF\_8} \\ \hline g? = \textit{forward} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_20} \\ \hline \textit{ReadShockAbsorbers\_DNF\_8} \\ \hline g? = \textit{left} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_21} \\ \hline \textit{ReadShockAbsorbers\_DNF\_8} \\ \hline g? = \textit{right} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_DNF\_10} \\ \hline \textit{ReadShockAbsorbers\_VIS} \\ \hline \textit{sSA}\ g? \subset \textit{SENS} \\ \mathrm{dom}(\textit{sSA}\ g? \lhd v? \rhd \{y\}) = \textit{sSA}\ g? \\ \#(\textit{sSA}\ g? \lhd v? \rhd \{y\}) > \#(\textit{sSA}\ g? \lhd v? \rhd \{n\}) \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_25} \\ \hline \textit{ReadShockAbsorbers\_DNF\_10} \\ \hline g? = \textit{forward} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_26} \\ \hline \textit{ReadShockAbsorbers\_DNF\_10} \\ \hline g? = \textit{left} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_27} \\ \hline \textit{ReadShockAbsorbers\_DNF\_10} \\ \hline g? = \textit{right} \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_DNF\_11} \\ \hline \textit{ReadShockAbsorbers\_VIS} \\ \hline \textit{sSA}\ g? \subset \textit{SENS} \\ \#(\textit{sSA}\ g? \lhd v? \rhd \{y\}) \leq \#(\textit{sSA}\ g? \lhd v? \rhd \{n\}) \\ \mathrm{dom}(\textit{sSA}\ g? \lhd v? \rhd \{n\}) = \textit{sSA}\ g? \end{array}$$

$$\begin{array}{l} \textit{ReadShockAbsorbers\_FT\_28} \\ \hline \textit{ReadShockAbsorbers\_DNF\_11} \\ \hline g? = \textit{forward} \end{array}$$

ReadShockAbsorbers_FT_29

ReadShockAbsorbers_DNF_11

$g? = left$

ReadShockAbsorbers_FT_30

ReadShockAbsorbers_DNF_11

$g? = right$

ReadShockAbsorbers_DNF_12

ReadShockAbsorbers_VIS

$sSA\, g? \subset SENS$
$\mathrm{dom}(sSA\, g? \lhd v? \rhd \{y\}) = sSA\, g?$
$\mathrm{dom}(sSA\, g? \lhd v? \rhd \{n\}) = sSA\, g?$

ReadShockAbsorbers_FT_31

ReadShockAbsorbers_DNF_12

$g? = forward$

ReadShockAbsorbers_FT_32

ReadShockAbsorbers_DNF_12

$g? = left$

ReadShockAbsorbers_FT_33

ReadShockAbsorbers_DNF_12

$g? = right$

AnalogicalSwitchM_DNF_1

AnalogicalSwitchM_VIS

$as = y$
$1000 \leq now - lHPCh$
$lHPCh \neq 0$

AnalogicalSwitchM_SP_6

AnalogicalSwitchM_DNF_1

$now > 0$
$lHPCh > 0$
$now > lHPCh$

AnalogicalSwitchM_SP_29

AnalogicalSwitchM_SP_6

$1000 > 0$
$now - lHPCh > 0$
$1000 < now - lHPCh$

AnalogicalSwitchM_SP_30

AnalogicalSwitchM_SP_6

$1000 > 0$
$now - lHPCh > 0$
$1000 = now - lHPCh$

AnalogicalSwitchM_DNF_2

AnalogicalSwitchM_VIS

$as = n$
$1500 \leq now - l20$
$l20 \neq 0$

AnalogicalSwitchM_SP_36

AnalogicalSwitchM_DNF_2

$now > 0$
$l20 > 0$
$now > l20$

AnalogicalSwitchM_SP_59

AnalogicalSwitchM_SP_36

$1500 > 0$
$now - l20 > 0$
$1500 < now - l20$

AnalogicalSwitchM_SP_60

AnalogicalSwitchM_SP_36

$1500 > 0$
$now - l20 > 0$
$1500 = now - l20$

$AnalogicalSwitchM\_DNF\_3$
$AnalogicalSwitchM\_VIS$
$as \neq y$

$AnalogicalSwitchM\_DNF\_4$
$AnalogicalSwitchM\_VIS$
$1000 > now - lHPCh$

$AnalogicalSwitchM\_DNF\_5$
$AnalogicalSwitchM\_VIS$
$lHPCh = 0$

$AnalogicalSwitchM\_DNF\_6$
$AnalogicalSwitchM\_VIS$
$as \neq n$

$AnalogicalSwitchM\_DNF\_7$
$AnalogicalSwitchM\_VIS$
$1500 > now - l20$

$AnalogicalSwitchM\_DNF\_8$
$AnalogicalSwitchM\_VIS$
$l20 = 0$

$Valid\_DNF\_1$
$Valid\_VIS$
$\#(i? \rhd \{y\}) > \#(i? \rhd \{n\})$

$Valid\_SP\_3$
$Valid\_DNF\_1$
$i? \neq \{\}$
$\{y\} = \operatorname{ran} i?$

$Valid\_SP\_26$
$Valid\_SP\_3$
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \cap \operatorname{ran} i? = \{\}$

$Valid\_SP\_4$
$Valid\_DNF\_1$
$i? \neq \{\}$
$\{y\} \neq \{\}$
$\{y\} \subset \operatorname{ran} i?$

$Valid\_SP\_32$
$Valid\_SP\_4$
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \subset \operatorname{ran} i?$

$Valid\_SP\_33$
$Valid\_SP\_4$
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \cap \operatorname{ran} i? = \{\}$

$Valid\_SP\_5$
$Valid\_DNF\_1$
$i? \neq \{\}$
$\{y\} \neq \{\}$
$\{y\} \cap \operatorname{ran} i? = \{\}$

$Valid\_SP\_38$
$Valid\_SP\_5$
$i? \neq \{\}$
$\{n\} = \operatorname{ran} i?$

Valid_SP_39
Valid_SP_5
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \subset \operatorname{ran} i?$

Valid_SP_40
Valid_SP_5
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \cap \operatorname{ran} i? = \{\}$

Valid_DNF_2
Valid_VIS
$\#(i? \rhd \{y\}) \leq \#(i? \rhd \{n\})$

Valid_SP_57
Valid_DNF_2
$i? = \{\}$

Valid_SP_64
Valid_SP_57
$i? = \{\}$

Valid_SP_59
Valid_DNF_2
$i? \neq \{\}$
$\{y\} = \operatorname{ran} i?$

Valid_SP_82
Valid_SP_59
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \cap \operatorname{ran} i? = \{\}$

Valid_SP_60
Valid_DNF_2
$i? \neq \{\}$
$\{y\} \neq \{\}$
$\{y\} \subset \operatorname{ran} i?$

Valid_SP_88
Valid_SP_60
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \subset \operatorname{ran} i?$

Valid_SP_89
Valid_SP_60
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \cap \operatorname{ran} i? = \{\}$

Valid_SP_61
Valid_DNF_2
$i? \neq \{\}$
$\{y\} \neq \{\}$
$\{y\} \cap \operatorname{ran} i? = \{\}$

Valid_SP_94
Valid_SP_61
$i? \neq \{\}$
$\{n\} = \operatorname{ran} i?$

Valid_SP_95
Valid_SP_61
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \subset \operatorname{ran} i?$

$Valid\_SP\_96$
$Valid\_SP\_61$
$i? \neq \{\}$
$\{n\} \neq \{\}$
$\{n\} \cap \mathrm{ran}\, i? = \{\}$

$Down1\_DNF\_1$
$Down1\_VIS$
$st = d0$
$hPos = down$
$200 \leq now - stEV$

$Down1\_SP\_3$
$Down1\_DNF\_1$
$now > 0$
$stEV = 0$

$Down1\_SP\_17$
$Down1\_SP\_3$
$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Down1\_SP\_18$
$Down1\_SP\_3$
$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Down1\_SP\_6$
$Down1\_DNF\_1$
$now > 0$
$stEV > 0$
$now > stEV$

$Down1\_SP\_29$
$Down1\_SP\_6$
$200 > 0$
$now - stEV > 0$
$200 < now - stEV$

$Down1\_SP\_30$
$Down1\_SP\_6$
$200 > 0$
$now - stEV > 0$
$200 = now - stEV$

$Down1\_DNF\_2$
$Down1\_VIS$
$st = d0$
$hPos = down$
$stEV = 0$

$Down1\_DNF\_3$
$Down1\_VIS$
$st \neq d0$

$Down1\_DNF\_4$
$Down1\_VIS$
$hPos \neq down$

$Down1\_DNF\_5$
$Down1\_VIS$
$200 > now - stEV$
$stEV \neq 0$

B Abstract Test Cases

$$\begin{array}{l}
\textit{ReadGearsExtending_FT_7_TCASE} \\
\hline
\textit{ReadGearsExtending_FT_7} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = forward \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadGearsExtending_FT_8_TCASE} \\
\hline
\textit{ReadGearsExtending_FT_8} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = left \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadGearsExtending_FT_9_TCASE} \\
\hline
\textit{ReadGearsExtending_FT_9} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = right \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadGearsExtending_FT_13_TCASE} \\
\hline
\textit{ReadGearsExtending_FT_13} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = forward \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadGearsExtending_FT_14_TCASE} \\
\hline
\textit{ReadGearsExtending_FT_14} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = left \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}
\end{array}$$

$ReadGearsExtending\_FT\_15\_TCASE$
$ReadGearsExtending\_FT\_15$

$$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$$
$$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$$
$$lgsfl = on$$
$$g? = right$$
$$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$$

$ReadGearsExtending\_FT\_16\_TCASE$
$ReadGearsExtending\_FT\_16$

$$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$$
$$sGExt = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$$
$$lgsfl = on$$
$$g? = forward$$
$$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$$

$ReadGearsExtending\_FT\_17\_TCASE$
$ReadGearsExtending\_FT\_17$

$$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$$
$$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$$
$$lgsfl = on$$
$$g? = left$$
$$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$$

$ReadGearsExtending\_FT\_18\_TCASE$
$ReadGearsExtending\_FT\_18$

$$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$$
$$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$$
$$lgsfl = on$$
$$g? = right$$
$$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$$

$ReadGearsExtending\_FT\_19\_TCASE$
$ReadGearsExtending\_FT\_19$

$$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$$
$$sGExt = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$$
$$lgsfl = on$$
$$g? = forward$$
$$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$$

$$
\begin{array}{l}
\textit{ReadGearsExtending\_FT\_20\_TCASE} \\
\hline
\textit{ReadGearsExtending\_FT\_20} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = left \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}
\end{array}
$$

$$
\begin{array}{l}
\textit{ReadGearsExtending\_FT\_21\_TCASE} \\
\hline
\textit{ReadGearsExtending\_FT\_21} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\} \\
lgsfl = on \\
g? = right \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}
\end{array}
$$

$$
\begin{array}{l}
\textit{ReadGearsExtending\_FT\_28\_TCASE} \\
\hline
\textit{ReadGearsExtending\_FT\_28} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = forward \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}
$$

$$
\begin{array}{l}
\textit{ReadGearsExtending\_FT\_29\_TCASE} \\
\hline
\textit{ReadGearsExtending\_FT\_29} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = left \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}
$$

$$
\begin{array}{l}
\textit{ReadGearsExtending\_FT\_30\_TCASE} \\
\hline
\textit{ReadGearsExtending\_FT\_30} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\} \\
lgsfl = on \\
g? = right \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}
$$

ReadGearsExtending_FT_31_TCASE

$ReadGearsExtending\_FT\_31$

$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadGearsExtending_FT_32_TCASE

$ReadGearsExtending\_FT\_32$

$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadGearsExtending_FT_33_TCASE

$ReadGearsExtending\_FT\_33$

$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\}$
$lgsfl = on$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadGearsRetracting_FT_7_TCASE

$ReadGearsRetracting\_FT\_7$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$g? = forward$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadGearsRetracting_FT_8_TCASE

$ReadGearsRetracting\_FT\_8$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$g? = left$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadGearsRetracting_FT_9_TCASE

$ReadGearsRetracting\_FT\_9$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$g? = right$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadGearsRetracting_FT_13_TCASE

$ReadGearsRetracting\_FT\_13$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$g? = forward$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

ReadGearsRetracting_FT_14_TCASE

$ReadGearsRetracting\_FT\_14$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$g? = left$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

ReadGearsRetracting_FT_15_TCASE

$ReadGearsRetracting\_FT\_15$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$g? = right$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

ReadGearsRetracting_FT_16_TCASE

$ReadGearsRetracting\_FT\_16$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$g? = forward$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$

$ReadGearsRetracting\_FT\_17\_TCASE$

$ReadGearsRetracting\_FT\_17$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$g? = left$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$

$ReadGearsRetracting\_FT\_18\_TCASE$

$ReadGearsRetracting\_FT\_18$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$g? = right$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$

$ReadGearsRetracting\_FT\_19\_TCASE$

$ReadGearsRetracting\_FT\_19$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$g? = forward$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$ReadGearsRetracting\_FT\_20\_TCASE$

$ReadGearsRetracting\_FT\_20$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$g? = left$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$ReadGearsRetracting\_FT\_21\_TCASE$

$ReadGearsRetracting\_FT\_21$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$g? = right$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$$\begin{array}{l}
\textit{ReadGearsRetracting_FT_28_TCASE} \\
\hline
\textit{ReadGearsRetracting_FT_28} \\
\hline
lgsfl = on \\
sGRec = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
g? = forward \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadGearsRetracting_FT_29_TCASE} \\
\hline
\textit{ReadGearsRetracting_FT_29} \\
\hline
lgsfl = on \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\} \\
g? = left \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadGearsRetracting_FT_30_TCASE} \\
\hline
\textit{ReadGearsRetracting_FT_30} \\
\hline
lgsfl = on \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\} \\
g? = right \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadGearsRetracting_FT_31_TCASE} \\
\hline
\textit{ReadGearsRetracting_FT_31} \\
\hline
lgsfl = on \\
sGRec = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
g? = forward \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadGearsRetracting_FT_32_TCASE} \\
\hline
\textit{ReadGearsRetracting_FT_32} \\
\hline
lgsfl = on \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\} \\
g? = left \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$ReadGearsRetracting\_FT\_33\_TCASE$

$ReadGearsRetracting\_FT\_33$

$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\}$
$g? = right$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$HandleNotChanged\_SP\_6\_TCASE$

$HandleNotChanged\_SP\_6$

$l20 = 0$
$hPos = down$
$now = 21$
$lHPCh = 1$

$HandleNotChanged\_DNF\_2\_TCASE$

$HandleNotChanged\_DNF\_2$

$l20 = 0$
$hPos = down$
$now = 0$
$lHPCh = 0$

$HandleNotChanged\_DNF\_3\_TCASE$

$HandleNotChanged\_DNF\_3$

$l20 = 0$
$hPos = down$
$now = 0$
$lHPCh = 0$

$Up2\_SP\_335\_TCASE$

$Up2\_SP\_335$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u1$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Up2_SP_347_TCASE

Up2_SP_347

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u1$
$stDCEV = 1$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Up2_SP_348_TCASE

Up2_SP_348

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u1$
$stDCEV = 101$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Up2_SP_365_TCASE

Up2_SP_365

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u1$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up2\_SP\_377\_TCASE$

$Up2\_SP\_377$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u1$
$stDCEV = 1$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up2\_SP\_378\_TCASE$

$Up2\_SP\_378$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u1$
$stDCEV = 100$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up2\_SP\_707\_TCASE$

$Up2\_SP\_707$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u1$
$stDCEV = 0$
$now = 202$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up2\_SP\_719\_TCASE$

$Up2\_SP\_719$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u1$
$stDCEV = 1$
$now = 202$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up2\_SP\_720\_TCASE$

$Up2\_SP\_720$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u1$
$stDCEV = 102$
$now = 202$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up2\_SP\_737\_TCASE$

$Up2\_SP\_737$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u1$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Up2_SP_749_TCASE

Up2_SP_749

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u1$
$stDCEV = 1$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Up2_SP_750_TCASE

Up2_SP_750

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u1$
$stDCEV = 101$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Up2_DNF_2_TCASE

Up2_DNF_2

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u1$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$$
\begin{array}{l}
\textit{Up2\_DNF\_3\_TCASE} \\
\hline
\textit{Up2\_DNF\_3} \\
\hline
l20 = 0 \\
doEV = pressing \\
stEV = 0 \\
hPos = down \\
st = u1 \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
stDOEV = 0 \\
lHPCh = 0
\end{array}
$$

$$
\begin{array}{l}
\textit{Up2\_DNF\_4\_TCASE} \\
\hline
\textit{Up2\_DNF\_4} \\
\hline
l20 = 0 \\
doEV = pressing \\
stEV = 0 \\
hPos = down \\
st = init \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
stDOEV = 0 \\
lHPCh = 0
\end{array}
$$

$$
\begin{array}{l}
\textit{Up2\_DNF\_5\_TCASE} \\
\hline
\textit{Up2\_DNF\_5} \\
\hline
l20 = 0 \\
doEV = pressing \\
stEV = 0 \\
hPos = down \\
st = init \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
stDOEV = 0 \\
lHPCh = 0
\end{array}
$$

Up2_DNF_6_TCASE
Up2_DNF_6

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stDCEV = 1$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

ReadAnalogicalSwitch_DNF_3_TCASE
ReadAnalogicalSwitch_DNF_3

$lgsfl = on$
$sAS = \emptyset$
$as = y$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadAnalogicalSwitch_DNF_6_TCASE
ReadAnalogicalSwitch_DNF_6

$lgsfl = on$
$sAS = \{s1, s2, s3\}$
$as = y$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

ReadAnalogicalSwitch_DNF_7_TCASE
ReadAnalogicalSwitch_DNF_7

$lgsfl = on$
$sAS = \{s1, s2, s3\}$
$as = y$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$

ReadAnalogicalSwitch_DNF_8_TCASE
ReadAnalogicalSwitch_DNF_8

$lgsfl = on$
$sAS = \{s1, s2, s3\}$
$as = y$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$ReadAnalogicalSwitch\_DNF\_11\_TCASE$

$ReadAnalogicalSwitch\_DNF\_11$

$lgsfl = on$
$sAS = \emptyset$
$as = y$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$ReadAnalogicalSwitch\_DNF\_12\_TCASE$

$ReadAnalogicalSwitch\_DNF\_12$

$lgsfl = on$
$sAS = \emptyset$
$as = y$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$Up1\_SP\_17\_TCASE$

$Up1\_SP\_17$

$l20 = 0$
$stEV = 0$
$hPos = up$
$st = u0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 201$
$lHPCh = 0$

$Up1\_SP\_18\_TCASE$

$Up1\_SP\_18$

$l20 = 0$
$stEV = 0$
$hPos = up$
$st = u0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 200$
$lHPCh = 0$

$Up1\_SP\_29\_TCASE$

$Up1\_SP\_29$

$l20 = 0$
$stEV = 1$
$hPos = up$
$st = u0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 202$
$lHPCh = 0$

$Up1\_SP\_30\_TCASE$

$Up1\_SP\_30$

$l20 = 0$
$stEV = 1$
$hPos = up$
$st = u0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 201$
$lHPCh = 0$

$Up1\_DNF\_2\_TCASE$

$Up1\_DNF\_2$

$l20 = 0$
$stEV = 0$
$hPos = up$
$st = u0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Up1\_DNF\_3\_TCASE$
$Up1\_DNF\_3$

$l20 = 0$
$stEV = 0$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Up1\_DNF\_4\_TCASE$
$Up1\_DNF\_4$

$l20 = 0$
$stEV = 0$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Up1\_DNF\_5\_TCASE$
$Up1\_DNF\_5$

$l20 = 0$
$stEV = 1$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Up4\_SP\_17\_TCASE$

$Up4\_SP\_17$

$l20 = 0$
$grEV = pressing$
$hPos = up$
$st = u3$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$spEV = 0$
$now = 1001$
$lHPCh = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Up4\_SP\_18\_TCASE$

$Up4\_SP\_18$

$l20 = 0$
$grEV = pressing$
$hPos = up$
$st = u3$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$spEV = 0$
$now = 1000$
$lHPCh = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Up4\_SP\_29\_TCASE$

$Up4\_SP\_29$

$l20 = 0$
$grEV = pressing$
$hPos = up$
$st = u3$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$spEV = 1$
$now = 1002$
$lHPCh = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Up4\_SP\_30\_TCASE$

$Up4\_SP\_30$

$$
\begin{array}{l}
l20 = 0 \\
grEV = pressing \\
hPos = up \\
st = u3 \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
stGREV = 0 \\
spEV = 1 \\
now = 1001 \\
lHPCh = 0 \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$Up4\_DNF\_2\_TCASE$

$Up4\_DNF\_2$

$$
\begin{array}{l}
l20 = 0 \\
grEV = pressing \\
hPos = up \\
st = u3 \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
stGREV = 0 \\
spEV = 0 \\
now = 0 \\
lHPCh = 0 \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$Up4\_DNF\_3\_TCASE$

$Up4\_DNF\_3$

$$
\begin{array}{l}
l20 = 0 \\
grEV = pressing \\
hPos = down \\
st = u3 \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
stGREV = 0 \\
spEV = 0 \\
now = 0 \\
lHPCh = 0 \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$Up4\_DNF\_4\_TCASE$

$Up4\_DNF\_4$

$l20 = 0$
$grEV = pressing$
$hPos = down$
$st = init$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$spEV = 0$
$now = 0$
$lHPCh = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Up4\_DNF\_5\_TCASE$

$Up4\_DNF\_5$

$l20 = 0$
$grEV = pressing$
$hPos = down$
$st = init$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$spEV = 0$
$now = 0$
$lHPCh = 0$
$gRec = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$

$Up4\_DNF\_6\_TCASE$

$Up4\_DNF\_6$

$l20 = 0$
$grEV = pressing$
$hPos = down$
$st = init$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$spEV = 1$
$now = 0$
$lHPCh = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$$\begin{array}{l}
\textit{ReadDoorsOpening_FT_7_TCASE} \\
\hline
\textit{ReadDoorsOpening_FT_7} \\
\hline
lgsfl = on \\
g? = forward \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadDoorsOpening_FT_8_TCASE} \\
\hline
\textit{ReadDoorsOpening_FT_8} \\
\hline
lgsfl = on \\
g? = left \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadDoorsOpening_FT_9_TCASE} \\
\hline
\textit{ReadDoorsOpening_FT_9} \\
\hline
lgsfl = on \\
g? = right \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadDoorsOpening_FT_13_TCASE} \\
\hline
\textit{ReadDoorsOpening_FT_13} \\
\hline
lgsfl = on \\
g? = forward \\
sDOp = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadDoorsOpening_FT_14_TCASE} \\
\hline
\textit{ReadDoorsOpening_FT_14} \\
\hline
lgsfl = on \\
g? = left \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$ReadDoorsOpening\_FT\_15\_TCASE$

$ReadDoorsOpening\_FT\_15$

$lgsfl = on$
$g? = right$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadDoorsOpening\_FT\_16\_TCASE$

$ReadDoorsOpening\_FT\_16$

$lgsfl = on$
$g? = forward$
$sDOp = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadDoorsOpening\_FT\_17\_TCASE$

$ReadDoorsOpening\_FT\_17$

$lgsfl = on$
$g? = left$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadDoorsOpening\_FT\_18\_TCASE$

$ReadDoorsOpening\_FT\_18$

$lgsfl = on$
$g? = right$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadDoorsOpening\_FT\_19\_TCASE$

$ReadDoorsOpening\_FT\_19$

$lgsfl = on$
$g? = forward$
$sDOp = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$$
\begin{array}{l}
\textbf{ReadDoorsOpening\_FT\_20\_TCASE} \\
\hline
ReadDoorsOpening\_FT\_20 \\
\hline
lgsfl = on \\
g? = left \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$$
\begin{array}{l}
\textbf{ReadDoorsOpening\_FT\_21\_TCASE} \\
\hline
ReadDoorsOpening\_FT\_21 \\
\hline
lgsfl = on \\
g? = right \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$$
\begin{array}{l}
\textbf{ReadDoorsOpening\_FT\_28\_TCASE} \\
\hline
ReadDoorsOpening\_FT\_28 \\
\hline
lgsfl = on \\
g? = forward \\
sDOp = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$$
\begin{array}{l}
\textbf{ReadDoorsOpening\_FT\_29\_TCASE} \\
\hline
ReadDoorsOpening\_FT\_29 \\
\hline
lgsfl = on \\
g? = left \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$$
\begin{array}{l}
\textbf{ReadDoorsOpening\_FT\_30\_TCASE} \\
\hline
ReadDoorsOpening\_FT\_30 \\
\hline
lgsfl = on \\
g? = right \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

ReadDoorsOpening_FT_31_TCASE

$ReadDoorsOpening\_FT\_31$

$lgsfl = on$
$g? = forward$
$sDOp = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

ReadDoorsOpening_FT_32_TCASE

$ReadDoorsOpening\_FT\_32$

$lgsfl = on$
$g? = left$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

ReadDoorsOpening_FT_33_TCASE

$ReadDoorsOpening\_FT\_33$

$lgsfl = on$
$g? = right$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\}$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

ChangeHandle_DNF_1_TCASE

$ChangeHandle\_DNF\_1$

$l20 = 0$
$hPos = down$
$st = init$
$now = 0$
$lHPCh = 0$

ChangeHandle_DNF_2_TCASE

$ChangeHandle\_DNF\_2$

$l20 = 0$
$hPos = up$
$st = init$
$now = 0$
$lHPCh = 0$

$$
\begin{array}{l}
\textit{Up3\_SP\_335\_TCASE} \\
\hline
Up3\_SP\_335 \\
\hline
l20 = 0 \\
stEV = 0 \\
now = 201 \\
lHPCh = 0 \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
grEV = pressing \\
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
hPos = up \\
st = u2 \\
geEV = pressing \\
stGREV = 0 \\
stGEEV = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\} \\
\hline
\end{array}
$$

$$
\begin{array}{l}
\textit{Up3\_SP\_336\_TCASE} \\
\hline
Up3\_SP\_336 \\
\hline
l20 = 0 \\
stEV = 0 \\
now = 200 \\
lHPCh = 0 \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
grEV = pressing \\
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
hPos = up \\
st = u2 \\
geEV = pressing \\
stGREV = 0 \\
stGEEV = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\} \\
\hline
\end{array}
$$

$Up3\_SP\_347\_TCASE$

$Up3\_SP\_347$

$l20 = 0$
$stEV = 1$
$now = 202$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$

$Up3\_SP\_348\_TCASE$

$Up3\_SP\_348$

$l20 = 0$
$stEV = 1$
$now = 201$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$

$$
\begin{array}{l}
\textbf{Up3\_SP\_707\_TCASE} \\
Up3\_SP\_707 \\
\hline
l20 = 0 \\
stEV = 0 \\
now = 201 \\
lHPCh = 0 \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
grEV = pressing \\
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
hPos = up \\
st = u2 \\
geEV = pressing \\
stGREV = 0 \\
stGEEV = 1 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}
\end{array}
$$

$$
\begin{array}{l}
\textbf{Up3\_SP\_708\_TCASE} \\
Up3\_SP\_708 \\
\hline
l20 = 0 \\
stEV = 0 \\
now = 200 \\
lHPCh = 0 \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
grEV = pressing \\
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
hPos = up \\
st = u2 \\
geEV = pressing \\
stGREV = 0 \\
stGEEV = 1 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}
\end{array}
$$

$$
\begin{array}{|l}
\hline
\textit{Up3\_SP\_719\_TCASE} \\
\textit{Up3\_SP\_719} \\
\hline
l20 = 0 \\
stEV = 1 \\
now = 202 \\
lHPCh = 0 \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
grEV = pressing \\
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
hPos = up \\
st = u2 \\
geEV = pressing \\
stGREV = 0 \\
stGEEV = 1 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\} \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\textit{Up3\_SP\_720\_TCASE} \\
\textit{Up3\_SP\_720} \\
\hline
l20 = 0 \\
stEV = 1 \\
now = 201 \\
lHPCh = 0 \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
grEV = pressing \\
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
hPos = up \\
st = u2 \\
geEV = pressing \\
stGREV = 0 \\
stGEEV = 1 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\} \\
\hline
\end{array}
$$

$Up3\_SP\_737\_TCASE$

$Up3\_SP\_737$

$l20 = 0$
$stEV = 0$
$now = 201$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 101$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$

$Up3\_SP\_738\_TCASE$

$Up3\_SP\_738$

$l20 = 0$
$stEV = 0$
$now = 200$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 100$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$

$Up3\_SP\_749\_TCASE$

$Up3\_SP\_749$

$l20 = 0$
$stEV = 1$
$now = 202$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 102$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$

$Up3\_SP\_750\_TCASE$

$Up3\_SP\_750$

$l20 = 0$
$stEV = 1$
$now = 201$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 101$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$

$Up3\_DNF\_2\_TCASE$

$Up3\_DNF\_2$

$l20 = 0$
$stEV = 0$
$now = 200$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$

$Up3\_DNF\_3\_TCASE$

$Up3\_DNF\_3$

$l20 = 0$
$stEV = 0$
$now = 0$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Up3\_DNF\_4\_TCASE$
$Up3\_DNF\_4$

$l20 = 0$
$stEV = 0$
$now = 0$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = u2$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Up3\_DNF\_5\_TCASE$
$Up3\_DNF\_5$

$l20 = 0$
$stEV = 0$
$now = 0$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = init$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$$
\begin{array}{|l}
\hline
\textit{Up3\_DNF\_6\_TCASE} \\
\hline
\textit{Up3\_DNF\_6} \\
\hline
l20 = 0 \\
stEV = 0 \\
now = 0 \\
lHPCh = 0 \\
dOp = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\} \\
grEV = pressing \\
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
hPos = down \\
st = init \\
geEV = pressing \\
stGREV = 0 \\
stGEEV = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
\hline
\textit{Up3\_DNF\_7\_TCASE} \\
\hline
\textit{Up3\_DNF\_7} \\
\hline
l20 = 0 \\
stEV = 0 \\
now = 0 \\
lHPCh = 0 \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
grEV = pressing \\
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
hPos = down \\
st = init \\
geEV = pressing \\
stGREV = 0 \\
stGEEV = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
\hline
\end{array}
$$

Up3_DNF_8_TCASE

$Up3\_DNF\_8$

$l20 = 0$
$stEV = 0$
$now = 0$
$lHPCh = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$grEV = pressing$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = init$
$geEV = pressing$
$stGREV = 0$
$stGEEV = 1$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

Up6_SP_335_TCASE

$Up6\_SP\_335$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Up6_SP_347_TCASE

$Up6\_SP\_347$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 1$
$lHPCh = 0$

$Up6\_SP\_348\_TCASE$

$Up6\_SP\_348$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 101$
$lHPCh = 0$

$Up6\_SP\_365\_TCASE$

$Up6\_SP\_365$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up6\_SP\_377\_TCASE$

$Up6\_SP\_377$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 1$
$lHPCh = 0$

$Up6\_SP\_378\_TCASE$

$Up6\_SP\_378$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 100$
$lHPCh = 0$

$Up6\_SP\_707\_TCASE$

$Up6\_SP\_707$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 202$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up6\_SP\_719\_TCASE$

$Up6\_SP\_719$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 202$
$dcEV = pressing$
$stDOEV = 1$
$lHPCh = 0$

$Up6\_SP\_720\_TCASE$

$Up6\_SP\_720$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 202$
$dcEV = pressing$
$stDOEV = 102$
$lHPCh = 0$

$Up6\_SP\_737\_TCASE$

$Up6\_SP\_737$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up6\_SP\_749\_TCASE$

$Up6\_SP\_749$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 1$
$lHPCh = 0$

$Up6\_SP\_750\_TCASE$

$Up6\_SP\_750$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = up$
$st = u5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 101$
$lHPCh = 0$

$Up6\_DNF\_2\_TCASE$

$Up6\_DNF\_2$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = u5$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Up6\_DNF\_3\_TCASE$

$Up6\_DNF\_3$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$$
\begin{array}{l}
\textbf{Up6\_DNF\_4\_TCASE} \\
\hline
Up6\_DNF\_4 \\
\hline
l20 = 0 \\
doEV = pressing \\
stEV = 0 \\
hPos = down \\
st = init \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
stDOEV = 0 \\
lHPCh = 0
\end{array}
$$

$$
\begin{array}{l}
\textbf{Up6\_DNF\_5\_TCASE} \\
\hline
Up6\_DNF\_5 \\
\hline
l20 = 0 \\
doEV = pressing \\
stEV = 0 \\
hPos = down \\
st = init \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
stDOEV = 0 \\
lHPCh = 0
\end{array}
$$

$$
\begin{array}{l}
\textbf{Up5\_SP\_17\_TCASE} \\
\hline
Up5\_SP\_17 \\
\hline
l20 = 0 \\
doEV = pressing \\
hPos = up \\
st = u4 \\
spEV = 0 \\
now = 1001 \\
stDOEV = 0 \\
lHPCh = 0
\end{array}
$$

Up5_SP_18_TCASE

$Up5\_SP\_18$

$$
\begin{aligned}
& l20 = 0 \\
& doEV = pressing \\
& hPos = up \\
& st = u4 \\
& spEV = 0 \\
& now = 1000 \\
& stDOEV = 0 \\
& lHPCh = 0
\end{aligned}
$$

Up5_SP_29_TCASE

$Up5\_SP\_29$

$$
\begin{aligned}
& l20 = 0 \\
& doEV = pressing \\
& hPos = up \\
& st = u4 \\
& spEV = 1 \\
& now = 1002 \\
& stDOEV = 0 \\
& lHPCh = 0
\end{aligned}
$$

Up5_SP_30_TCASE

$Up5\_SP\_30$

$$
\begin{aligned}
& l20 = 0 \\
& doEV = pressing \\
& hPos = up \\
& st = u4 \\
& spEV = 1 \\
& now = 1001 \\
& stDOEV = 0 \\
& lHPCh = 0
\end{aligned}
$$

$Up5\_DNF\_2\_TCASE$

$Up5\_DNF\_2$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = u4$
$spEV = 0$
$now = 0$
$stDOEV = 0$
$lHPCh = 0$

$Up5\_DNF\_3\_TCASE$

$Up5\_DNF\_3$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = init$
$spEV = 0$
$now = 0$
$stDOEV = 0$
$lHPCh = 0$

$Up5\_DNF\_4\_TCASE$

$Up5\_DNF\_4$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = init$
$spEV = 0$
$now = 0$
$stDOEV = 0$
$lHPCh = 0$

$Up8\_SP\_17\_TCASE$

$Up8\_SP\_17$

$l20 = 0$
$hPos = up$
$st = u7$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 1001$
$lHPCh = 0$

$Up8\_SP\_18\_TCASE$

$Up8\_SP\_18$

$l20 = 0$
$hPos = up$
$st = u7$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 1000$
$lHPCh = 0$

$Up8\_SP\_29\_TCASE$

$Up8\_SP\_29$

$l20 = 0$
$hPos = up$
$st = u7$
$gEV = pressing$
$stGEV = 0$
$spEV = 1$
$spGEV = 0$
$now = 1002$
$lHPCh = 0$

$Up8\_SP\_30\_TCASE$

$Up8\_SP\_30$

$l20 = 0$
$hPos = up$
$st = u7$
$gEV = pressing$
$stGEV = 0$
$spEV = 1$
$spGEV = 0$
$now = 1001$
$lHPCh = 0$

$Up8\_DNF\_2\_TCASE$

$Up8\_DNF\_2$

$l20 = 0$
$hPos = down$
$st = u7$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Up8\_DNF\_3\_TCASE$

$Up8\_DNF\_3$

$l20 = 0$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Up8\_DNF\_4\_TCASE$

$Up8\_DNF\_4$

$l20 = 0$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Up7\_SP\_17\_TCASE$

$Up7\_SP\_17$

$l20 = 0$
$hPos = up$
$st = u6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 1001$
$dcEV = pressing$
$lHPCh = 0$

$Up7\_SP\_18\_TCASE$

$Up7\_SP\_18$

$l20 = 0$
$hPos = up$
$st = u6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 1000$
$dcEV = pressing$
$lHPCh = 0$

$Up7\_SP\_29\_TCASE$

$Up7\_SP\_29$

$l20 = 0$
$hPos = up$
$st = u6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 1$
$stDCEV = 0$
$now = 1002$
$dcEV = pressing$
$lHPCh = 0$

$Up7\_SP\_30\_TCASE$

$Up7\_SP\_30$

$l20 = 0$
$hPos = up$
$st = u6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 1$
$stDCEV = 0$
$now = 1001$
$dcEV = pressing$
$lHPCh = 0$

$Up7\_DNF\_2\_TCASE$

$Up7\_DNF\_2$

$l20 = 0$
$hPos = down$
$st = u6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$lHPCh = 0$

$Up7\_DNF\_3\_TCASE$

$Up7\_DNF\_3$

$l20 = 0$
$hPos = down$
$st = init$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$lHPCh = 0$

$Up7\_DNF\_4\_TCASE$

$Up7\_DNF\_4$

$l20 = 0$
$hPos = down$
$st = init$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$lHPCh = 0$

$Up7\_DNF\_5\_TCASE$

$Up7\_DNF\_5$

$l20 = 0$
$hPos = down$
$st = init$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$lHPCh = 0$

$$\begin{array}{l}
\textit{GearsManeuvering_DNF_1_TCASE} \\
\hline
\textit{GearsManeuvering_DNF_1} \\
\hline
gExt = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
gml = on \\
sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
gldl = on \\
gRec = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{GearsManeuvering_DNF_2_TCASE} \\
\hline
\textit{GearsManeuvering_DNF_2} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
gml = on \\
sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dCl = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\} \\
gldl = on \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{GearsManeuvering_DNF_3_TCASE} \\
\hline
\textit{GearsManeuvering_DNF_3} \\
\hline
gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
gml = on \\
sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
gldl = on \\
gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$GearsManeuvering\_DNF\_4\_TCASE$

$GearsManeuvering\_DNF\_4$

$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$gml = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$gldl = on$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down5\_SP\_17\_TCASE$

$Down5\_SP\_17$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = d4$
$spEV = 0$
$now = 1001$
$stDOEV = 0$
$lHPCh = 0$

$Down5\_SP\_18\_TCASE$

$Down5\_SP\_18$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = d4$
$spEV = 0$
$now = 1000$
$stDOEV = 0$
$lHPCh = 0$

$Down5\_SP\_29\_TCASE$

$Down5\_SP\_29$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = d4$
$spEV = 1$
$now = 1002$
$stDOEV = 0$
$lHPCh = 0$

$Down5\_SP\_30\_TCASE$

$Down5\_SP\_30$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = d4$
$spEV = 1$
$now = 1001$
$stDOEV = 0$
$lHPCh = 0$

$Down5\_DNF\_2\_TCASE$

$Down5\_DNF\_2$

$l20 = 0$
$doEV = pressing$
$hPos = up$
$st = d4$
$spEV = 0$
$now = 0$
$stDOEV = 0$
$lHPCh = 0$

$Down5\_DNF\_3\_TCASE$

$Down5\_DNF\_3$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = init$
$spEV = 0$
$now = 0$
$stDOEV = 0$
$lHPCh = 0$

$Down5\_DNF\_4\_TCASE$

$Down5\_DNF\_4$

$l20 = 0$
$doEV = pressing$
$hPos = down$
$st = init$
$spEV = 0$
$now = 0$
$stDOEV = 0$
$lHPCh = 0$

$Down4\_SP\_17\_TCASE$

$Down4\_SP\_17$

$l20 = 0$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = d3$
$geEV = pressing$
$spEV = 0$
$now = 1001$
$lHPCh = 0$
$stGEEV = 0$

$Down4\_SP\_18\_TCASE$

$Down4\_SP\_18$

$l20 = 0$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = d3$
$geEV = pressing$
$spEV = 0$
$now = 1000$
$lHPCh = 0$
$stGEEV = 0$

$Down4\_SP\_29\_TCASE$

$Down4\_SP\_29$

$l20 = 0$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = d3$
$geEV = pressing$
$spEV = 1$
$now = 1002$
$lHPCh = 0$
$stGEEV = 0$

$Down4\_SP\_30\_TCASE$

$Down4\_SP\_30$

$l20 = 0$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = d3$
$geEV = pressing$
$spEV = 1$
$now = 1001$
$lHPCh = 0$
$stGEEV = 0$

$Down4\_DNF\_2\_TCASE$

$Down4\_DNF\_2$

$l20 = 0$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = d3$
$geEV = pressing$
$spEV = 0$
$now = 0$
$lHPCh = 0$
$stGEEV = 0$

$Down4\_DNF\_3\_TCASE$

$Down4\_DNF\_3$

$l20 = 0$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = up$
$st = d3$
$geEV = pressing$
$spEV = 0$
$now = 0$
$lHPCh = 0$
$stGEEV = 0$

$Down4\_DNF\_4\_TCASE$

$Down4\_DNF\_4$

$l20 = 0$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = init$
$geEV = pressing$
$spEV = 0$
$now = 0$
$lHPCh = 0$
$stGEEV = 0$

$Down4\_DNF\_5\_TCASE$

$Down4\_DNF\_5$

$l20 = 0$
$gExt = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = init$
$geEV = pressing$
$spEV = 0$
$now = 0$
$lHPCh = 0$
$stGEEV = 0$

$Down4\_DNF\_6\_TCASE$

$Down4\_DNF\_6$

$l20 = 0$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$hPos = down$
$st = init$
$geEV = pressing$
$spEV = 1$
$now = 0$
$lHPCh = 0$
$stGEEV = 0$

$HydraulicCircuitM\_SP\_59\_TCASE$

$HydraulicCircuitM\_SP\_59$

$hc = n$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 1$
$spGEV = 0$
$now = 2002$

HydraulicCircuitM_SP_60_TCASE

HydraulicCircuitM_SP_60

$hc = n$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 1$
$spGEV = 0$
$now = 2001$

HydraulicCircuitM_SP_29_TCASE

HydraulicCircuitM_SP_29

$hc = y$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 0$
$spGEV = 1$
$now = 10002$

HydraulicCircuitM_SP_30_TCASE

HydraulicCircuitM_SP_30

$hc = y$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 0$
$spGEV = 1$
$now = 10001$

HydraulicCircuitM_DNF_3_TCASE

HydraulicCircuitM_DNF_3

$hc = y$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 0$
$spGEV = 0$
$now = 0$

HydraulicCircuitM_DNF_4_TCASE

$HydraulicCircuitM\_DNF\_4$

$hc = y$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 0$
$spGEV = 0$
$now = 0$

HydraulicCircuitM_DNF_5_TCASE

$HydraulicCircuitM\_DNF\_5$

$hc = y$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 0$
$spGEV = 0$
$now = 0$

HydraulicCircuitM_DNF_6_TCASE

$HydraulicCircuitM\_DNF\_6$

$hc = n$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 0$
$spGEV = 0$
$now = 0$

HydraulicCircuitM_DNF_7_TCASE

$HydraulicCircuitM\_DNF\_7$

$hc = y$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 0$
$spGEV = 0$
$now = 0$

$HydraulicCircuitM\_DNF\_8\_TCASE$

$HydraulicCircuitM\_DNF\_8$

$hc = y$
$lgsfl = on$
$gEV = pressing$
$sHC = \emptyset$
$stGEV = 0$
$spGEV = 0$
$now = 0$

$Down3\_SP\_335\_TCASE$

$Down3\_SP\_335$

$l20 = 0$
$grEV = pressing$
$stEV = 0$
$hPos = down$
$st = d2$
$stGREV = 0$
$geEV = pressing$
$now = 201$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_SP\_347\_TCASE$

$Down3\_SP\_347$

$l20 = 0$
$grEV = pressing$
$stEV = 0$
$hPos = down$
$st = d2$
$stGREV = 1$
$geEV = pressing$
$now = 201$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_SP\_348\_TCASE$

$Down3\_SP\_348$

$l20 = 0$
$grEV = pressing$
$stEV = 0$
$hPos = down$
$st = d2$
$stGREV = 101$
$geEV = pressing$
$now = 201$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_SP\_365\_TCASE$

$Down3\_SP\_365$

$l20 = 0$
$grEV = pressing$
$stEV = 0$
$hPos = down$
$st = d2$
$stGREV = 0$
$geEV = pressing$
$now = 200$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_SP\_377\_TCASE$

$$Down3\_SP\_377$$

$$
\begin{aligned}
& l20 = 0 \\
& grEV = pressing \\
& stEV = 0 \\
& hPos = down \\
& st = d2 \\
& stGREV = 1 \\
& geEV = pressing \\
& now = 200 \\
& lHPCh = 0 \\
& sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
& stGEEV = 0 \\
& dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{aligned}
$$

$Down3\_SP\_378\_TCASE$

$$Down3\_SP\_378$$

$$
\begin{aligned}
& l20 = 0 \\
& grEV = pressing \\
& stEV = 0 \\
& hPos = down \\
& st = d2 \\
& stGREV = 100 \\
& geEV = pressing \\
& now = 200 \\
& lHPCh = 0 \\
& sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
& stGEEV = 0 \\
& dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{aligned}
$$

$Down3\_SP\_707\_TCASE$

$Down3\_SP\_707$

$l20 = 0$
$grEV = pressing$
$stEV = 1$
$hPos = down$
$st = d2$
$stGREV = 0$
$geEV = pressing$
$now = 202$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_SP\_719\_TCASE$

$Down3\_SP\_719$

$l20 = 0$
$grEV = pressing$
$stEV = 1$
$hPos = down$
$st = d2$
$stGREV = 1$
$geEV = pressing$
$now = 202$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_SP\_720\_TCASE$

$Down3\_SP\_720$

$l20 = 0$
$grEV = pressing$
$stEV = 1$
$hPos = down$
$st = d2$
$stGREV = 102$
$geEV = pressing$
$now = 202$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_SP\_737\_TCASE$

$Down3\_SP\_737$

$l20 = 0$
$grEV = pressing$
$stEV = 1$
$hPos = down$
$st = d2$
$stGREV = 0$
$geEV = pressing$
$now = 201$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_SP\_749\_TCASE$

$Down3\_SP\_749$

$$
\begin{aligned}
& l20 = 0 \\
& grEV = pressing \\
& stEV = 1 \\
& hPos = down \\
& st = d2 \\
& stGREV = 1 \\
& geEV = pressing \\
& now = 201 \\
& lHPCh = 0 \\
& sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
& stGEEV = 0 \\
& dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{aligned}
$$

$Down3\_SP\_750\_TCASE$

$Down3\_SP\_750$

$$
\begin{aligned}
& l20 = 0 \\
& grEV = pressing \\
& stEV = 1 \\
& hPos = down \\
& st = d2 \\
& stGREV = 101 \\
& geEV = pressing \\
& now = 201 \\
& lHPCh = 0 \\
& sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
& stGEEV = 0 \\
& dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{aligned}
$$

Down3_DNF_2_TCASE

$Down3\_DNF\_2$

$l20 = 0$
$grEV = pressing$
$stEV = 0$
$hPos = down$
$st = d2$
$stGREV = 0$
$geEV = pressing$
$now = 200$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

Down3_DNF_3_TCASE

$Down3\_DNF\_3$

$l20 = 0$
$grEV = pressing$
$stEV = 0$
$hPos = up$
$st = d2$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$$
\begin{array}{l}
\textbf{Down3\_DNF\_4\_TCASE} \\
\hline
Down3\_DNF\_4 \\
\hline
l20 = 0 \\
grEV = pressing \\
stEV = 0 \\
hPos = down \\
st = d2 \\
stGREV = 0 \\
geEV = pressing \\
now = 0 \\
lHPCh = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
stGEEV = 0 \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$$
\begin{array}{l}
\textbf{Down3\_DNF\_5\_TCASE} \\
\hline
Down3\_DNF\_5 \\
\hline
l20 = 0 \\
grEV = pressing \\
stEV = 0 \\
hPos = down \\
st = init \\
stGREV = 0 \\
geEV = pressing \\
now = 0 \\
lHPCh = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
stGEEV = 0 \\
dOp = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}
\end{array}
$$

$Down3\_DNF\_6\_TCASE$

$Down3\_DNF\_6$

$l20 = 0$
$grEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down3\_DNF\_7\_TCASE$

$Down3\_DNF\_7$

$l20 = 0$
$grEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stGREV = 1$
$geEV = pressing$
$now = 0$
$lHPCh = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGEEV = 0$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$Down2\_SP\_335\_TCASE$

$Down2\_SP\_335$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d1$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_SP_347_TCASE

$Down2\_SP\_347$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d1$
$stDCEV = 1$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_SP_348_TCASE

$Down2\_SP\_348$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d1$
$stDCEV = 101$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_SP_365_TCASE

$Down2\_SP\_365$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d1$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_SP_377_TCASE

Down2_SP_377

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d1$
$stDCEV = 1$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_SP_378_TCASE

Down2_SP_378

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d1$
$stDCEV = 100$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_SP_707_TCASE

Down2_SP_707

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d1$
$stDCEV = 0$
$now = 202$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down2\_SP\_719\_TCASE$

$Down2\_SP\_719$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d1$
$stDCEV = 1$
$now = 202$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down2\_SP\_720\_TCASE$

$Down2\_SP\_720$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d1$
$stDCEV = 102$
$now = 202$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down2\_SP\_737\_TCASE$

$Down2\_SP\_737$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d1$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_SP_749_TCASE

Down2_SP_749

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d1$
$stDCEV = 1$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_SP_750_TCASE

Down2_SP_750

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d1$
$stDCEV = 101$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

Down2_DNF_2_TCASE

Down2_DNF_2

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d1$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down2\_DNF\_3\_TCASE$

$Down2\_DNF\_3$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = d1$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down2\_DNF\_4\_TCASE$

$Down2\_DNF\_4$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down2\_DNF\_5\_TCASE$

$Down2\_DNF\_5$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down2\_DNF\_6\_TCASE$

$Down2\_DNF\_6$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stDCEV = 1$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$GearsLockedDown\_DNF\_1\_TCASE$

$GearsLockedDown\_DNF\_1$

$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$gml = on$
$gldl = on$

$GearsLockedDown\_DNF\_2\_TCASE$

$GearsLockedDown\_DNF\_2$

$gExt = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$gml = on$
$gldl = on$

$Down8\_SP\_17\_TCASE$

$Down8\_SP\_17$

$l20 = 0$
$hPos = down$
$st = d7$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 1001$
$lHPCh = 0$

$Down8\_SP\_18\_TCASE$

$Down8\_SP\_18$

$l20 = 0$
$hPos = down$
$st = d7$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 1000$
$lHPCh = 0$

$Down8\_SP\_29\_TCASE$

$Down8\_SP\_29$

$l20 = 0$
$hPos = down$
$st = d7$
$gEV = pressing$
$stGEV = 0$
$spEV = 1$
$spGEV = 0$
$now = 1002$
$lHPCh = 0$

$Down8\_SP\_30\_TCASE$

$Down8\_SP\_30$

$l20 = 0$
$hPos = down$
$st = d7$
$gEV = pressing$
$stGEV = 0$
$spEV = 1$
$spGEV = 0$
$now = 1001$
$lHPCh = 0$

Down8_DNF_2_TCASE

$Down8\_DNF\_2$

$l20 = 0$
$hPos = up$
$st = d7$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

Down8_DNF_3_TCASE

$Down8\_DNF\_3$

$l20 = 0$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

Down8_DNF_4_TCASE

$Down8\_DNF\_4$

$l20 = 0$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

ReadHydraulicCircuit_DNF_3_TCASE

$ReadHydraulicCircuit\_DNF\_3$

$hc = y$
$lgsfl = on$
$sHC = \emptyset$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$$\begin{array}{l}
\textbf{ReadHydraulicCircuit_DNF_6_TCASE} \\
\hline
ReadHydraulicCircuit_DNF_6 \\
\hline
hc = y \\
lgsfl = on \\
sHC = \{s1, s2, s3\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}
\end{array}$$

$$\begin{array}{l}
\textbf{ReadHydraulicCircuit_DNF_7_TCASE} \\
\hline
ReadHydraulicCircuit_DNF_7 \\
\hline
hc = y \\
lgsfl = on \\
sHC = \{s1, s2, s3\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}
\end{array}$$

$$\begin{array}{l}
\textbf{ReadHydraulicCircuit_DNF_8_TCASE} \\
\hline
ReadHydraulicCircuit_DNF_8 \\
\hline
hc = y \\
lgsfl = on \\
sHC = \{s1, s2, s3\} \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}
\end{array}$$

$$\begin{array}{l}
\textbf{ReadHydraulicCircuit_DNF_11_TCASE} \\
\hline
ReadHydraulicCircuit_DNF_11 \\
\hline
hc = y \\
lgsfl = on \\
sHC = \emptyset \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textbf{ReadHydraulicCircuit_DNF_12_TCASE} \\
\hline
ReadHydraulicCircuit_DNF_12 \\
\hline
hc = y \\
lgsfl = on \\
sHC = \emptyset \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}
\end{array}$$

ReadDoorsClosing_FT_7_TCASE

$ReadDoorsClosing\_FT\_7$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadDoorsClosing_FT_8_TCASE

$ReadDoorsClosing\_FT\_8$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadDoorsClosing_FT_9_TCASE

$ReadDoorsClosing\_FT\_9$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

ReadDoorsClosing_FT_13_TCASE

$ReadDoorsClosing\_FT\_13$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

ReadDoorsClosing_FT_14_TCASE

$ReadDoorsClosing\_FT\_14$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$ReadDoorsClosing\_FT\_15\_TCASE$

$ReadDoorsClosing\_FT\_15$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$ReadDoorsClosing\_FT\_16\_TCASE$

$ReadDoorsClosing\_FT\_16$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$

$ReadDoorsClosing\_FT\_17\_TCASE$

$ReadDoorsClosing\_FT\_17$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$

$ReadDoorsClosing\_FT\_18\_TCASE$

$ReadDoorsClosing\_FT\_18$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$

$ReadDoorsClosing\_FT\_19\_TCASE$

$ReadDoorsClosing\_FT\_19$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$ReadDoorsClosing\_FT\_20\_TCASE$

$ReadDoorsClosing\_FT\_20$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$ReadDoorsClosing\_FT\_21\_TCASE$

$ReadDoorsClosing\_FT\_21$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$

$ReadDoorsClosing\_FT\_28\_TCASE$

$ReadDoorsClosing\_FT\_28$

$lgsfl = on$
$sDCl = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$ReadDoorsClosing\_FT\_29\_TCASE$

$ReadDoorsClosing\_FT\_29$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$ReadDoorsClosing\_FT\_30\_TCASE$

$ReadDoorsClosing\_FT\_30$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$ReadDoorsClosing\_FT\_31\_TCASE$

$ReadDoorsClosing\_FT\_31$

$lgsfl = on$
$sDCl = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$ReadDoorsClosing\_FT\_32\_TCASE$

$ReadDoorsClosing\_FT\_32$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$ReadDoorsClosing\_FT\_33\_TCASE$

$ReadDoorsClosing\_FT\_33$

$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$

$Down7\_SP\_17\_TCASE$

$Down7\_SP\_17$

$l20 = 0$
$hPos = down$
$st = d6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 1001$
$dcEV = pressing$
$lHPCh = 0$

$Down7\_SP\_18\_TCASE$

$Down7\_SP\_18$

$l20 = 0$
$hPos = down$
$st = d6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 1000$
$dcEV = pressing$
$lHPCh = 0$

$Down7\_SP\_29\_TCASE$

$Down7\_SP\_29$

$l20 = 0$
$hPos = down$
$st = d6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 1$
$stDCEV = 0$
$now = 1002$
$dcEV = pressing$
$lHPCh = 0$

$Down7\_SP\_30\_TCASE$

$Down7\_SP\_30$

$l20 = 0$
$hPos = down$
$st = d6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 1$
$stDCEV = 0$
$now = 1001$
$dcEV = pressing$
$lHPCh = 0$

Down7_DNF_2_TCASE

$Down7\_DNF\_2$

$l20 = 0$
$hPos = up$
$st = d6$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$lHPCh = 0$

Down7_DNF_3_TCASE

$Down7\_DNF\_3$

$l20 = 0$
$hPos = down$
$st = init$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$lHPCh = 0$

Down7_DNF_4_TCASE

$Down7\_DNF\_4$

$l20 = 0$
$hPos = down$
$st = init$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$spEV = 0$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$lHPCh = 0$

Down7_DNF_5_TCASE

$Down7\_DNF\_5$

$$
\begin{array}{l}
l20 = 0 \\
hPos = down \\
st = init \\
sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
spEV = 0 \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
lHPCh = 0
\end{array}
$$

Down6_SP_335_TCASE

$Down6\_SP\_335$

$$
\begin{array}{l}
l20 = 0 \\
doEV = pressing \\
stEV = 0 \\
hPos = down \\
st = d5 \\
stDCEV = 0 \\
now = 201 \\
dcEV = pressing \\
stDOEV = 0 \\
lHPCh = 0
\end{array}
$$

Down6_SP_347_TCASE

$Down6\_SP\_347$

$$
\begin{array}{l}
l20 = 0 \\
doEV = pressing \\
stEV = 0 \\
hPos = down \\
st = d5 \\
stDCEV = 0 \\
now = 201 \\
dcEV = pressing \\
stDOEV = 1 \\
lHPCh = 0
\end{array}
$$

$Down6\_SP\_348\_TCASE$

$Down6\_SP\_348$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 101$
$lHPCh = 0$

$Down6\_SP\_365\_TCASE$

$Down6\_SP\_365$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down6\_SP\_377\_TCASE$

$Down6\_SP\_377$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 1$
$lHPCh = 0$

$Down6\_SP\_378\_TCASE$

$Down6\_SP\_378$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 200$
$dcEV = pressing$
$stDOEV = 100$
$lHPCh = 0$

$Down6\_SP\_707\_TCASE$

$Down6\_SP\_707$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 202$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down6\_SP\_719\_TCASE$

$Down6\_SP\_719$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 202$
$dcEV = pressing$
$stDOEV = 1$
$lHPCh = 0$

$Down6\_SP\_720\_TCASE$

$Down6\_SP\_720$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 202$
$dcEV = pressing$
$stDOEV = 102$
$lHPCh = 0$

$Down6\_SP\_737\_TCASE$

$Down6\_SP\_737$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down6\_SP\_749\_TCASE$

$Down6\_SP\_749$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 1$
$lHPCh = 0$

$Down6\_SP\_750\_TCASE$

$Down6\_SP\_750$

$l20 = 0$
$doEV = pressing$
$stEV = 1$
$hPos = down$
$st = d5$
$stDCEV = 0$
$now = 201$
$dcEV = pressing$
$stDOEV = 101$
$lHPCh = 0$

$Down6\_DNF\_2\_TCASE$

$Down6\_DNF\_2$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = up$
$st = d5$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down6\_DNF\_3\_TCASE$

$Down6\_DNF\_3$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down6\_DNF\_4\_TCASE$

$Down6\_DNF\_4$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$Down6\_DNF\_5\_TCASE$

$Down6\_DNF\_5$

$l20 = 0$
$doEV = pressing$
$stEV = 0$
$hPos = down$
$st = init$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$lHPCh = 0$

$GearsMotionM\_SP\_47\_TCASE$

$GearsMotionM\_SP\_47$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 7001$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_SP_48_TCASE

$GearsMotionM\_SP\_48$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 7000$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_SP_59_TCASE

$GearsMotionM\_SP\_59$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 1$
$geEV = pressing$
$now = 7002$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_SP_60_TCASE

$GearsMotionM\_SP\_60$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 1$
$geEV = pressing$
$now = 7001$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_SP_17_TCASE

GearsMotionM_SP_17

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 10001$
$gRec = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_SP_18_TCASE

GearsMotionM_SP_18

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 10000$
$gRec = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_SP_29_TCASE

GearsMotionM_SP_29

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 1$
$geEV = pressing$
$now = 10002$
$gRec = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_SP\_30\_TCASE$

$GearsMotionM\_SP\_30$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 1$
$geEV = pressing$
$now = 10001$
$gRec = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_SP\_107\_TCASE$

$GearsMotionM\_SP\_107$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 7001$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_SP\_108\_TCASE$

$GearsMotionM\_SP\_108$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 7000$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_SP\_119\_TCASE$

$GearsMotionM\_SP\_119$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 7002$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 1$

$GearsMotionM\_SP\_120\_TCASE$

$GearsMotionM\_SP\_120$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 7001$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 1$

$GearsMotionM\_SP\_77\_TCASE$

$GearsMotionM\_SP\_77$

$grEV = pressing$
$gExt = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 10001$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_SP\_78\_TCASE$

$GearsMotionM\_SP\_78$

$grEV = pressing$
$gExt = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 10000$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_SP\_89\_TCASE$

$GearsMotionM\_SP\_89$

$grEV = pressing$
$gExt = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 10002$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 1$

$GearsMotionM\_SP\_90\_TCASE$

$GearsMotionM\_SP\_90$

$grEV = pressing$
$gExt = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 10001$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 1$

$GearsMotionM\_DNF\_5\_TCASE$

$GearsMotionM\_DNF\_5$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$gRec = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$
$stGEEV = 0$

$GearsMotionM\_DNF\_6\_TCASE$

$GearsMotionM\_DNF\_6$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_DNF\_7\_TCASE$

$GearsMotionM\_DNF\_7$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_DNF_8_TCASE

$GearsMotionM\_DNF\_8$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_DNF_9_TCASE

$GearsMotionM\_DNF\_9$

$grEV = pressing$
$gExt = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

GearsMotionM_DNF_10_TCASE

$GearsMotionM\_DNF\_10$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_DNF\_11\_TCASE$

$GearsMotionM\_DNF\_11$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$GearsMotionM\_DNF\_12\_TCASE$

$GearsMotionM\_DNF\_12$

$grEV = pressing$
$gExt = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$sGExt = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$sGRec = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$stGREV = 0$
$geEV = pressing$
$now = 0$
$gRec = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stGEEV = 0$

$DoorsMotionM\_SP\_77\_TCASE$

$DoorsMotionM\_SP\_77$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7001$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_78\_TCASE$

$DoorsMotionM\_SP\_78$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7000$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_89\_TCASE$

$DoorsMotionM\_SP\_89$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7002$
$dcEV = pressing$
$stDOEV = 1$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_90\_TCASE$

$DoorsMotionM\_SP\_90$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7001$
$dcEV = pressing$
$stDOEV = 1$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_47\_TCASE$

$DoorsMotionM\_SP\_47$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7001$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_48\_TCASE$

$DoorsMotionM\_SP\_48$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7000$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_59\_TCASE$

$DoorsMotionM\_SP\_59$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7002$
$dcEV = pressing$
$stDOEV = 1$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_60\_TCASE$

$DoorsMotionM\_SP\_60$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7001$
$dcEV = pressing$
$stDOEV = 1$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_17\_TCASE$

$DoorsMotionM\_SP\_17$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7001$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_18\_TCASE$

$DoorsMotionM\_SP\_18$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7000$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_29\_TCASE$

$DoorsMotionM\_SP\_29$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 1$
$now = 7002$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_30\_TCASE$

$DoorsMotionM\_SP\_30$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 1$
$now = 7001$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_107\_TCASE$

$DoorsMotionM\_SP\_107$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7001$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_108\_TCASE$

$DoorsMotionM\_SP\_108$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 7000$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_119\_TCASE$

$DoorsMotionM\_SP\_119$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 1$
$now = 7002$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_SP\_120\_TCASE$

$DoorsMotionM\_SP\_120$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto n), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 1$
$now = 7001$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_DNF\_5\_TCASE$

$$\begin{array}{l}
DoorsMotionM_DNF_5 \\
\hline
doEV = pressing \\
lgsfl = on \\
sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dCl = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\} \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
stDOEV = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$DoorsMotionM\_DNF\_6\_TCASE$

$$\begin{array}{l}
DoorsMotionM_DNF_6 \\
\hline
doEV = pressing \\
lgsfl = on \\
sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
stDOEV = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$DoorsMotionM\_DNF\_7\_TCASE$

$$\begin{array}{l}
DoorsMotionM_DNF_7 \\
\hline
doEV = pressing \\
lgsfl = on \\
sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\} \\
stDCEV = 0 \\
now = 0 \\
dcEV = pressing \\
stDOEV = 0 \\
sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$DoorsMotionM\_DNF\_8\_TCASE$

$DoorsMotionM\_DNF\_8$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto n), (left \mapsto n), (right \mapsto n)\}$

$DoorsMotionM\_DNF\_9\_TCASE$

$DoorsMotionM\_DNF\_9$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$DoorsMotionM\_DNF\_10\_TCASE$

$DoorsMotionM\_DNF\_10$

$doEV = pressing$
$lgsfl = on$
$sDCl = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dCl = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$
$stDCEV = 0$
$now = 0$
$dcEV = pressing$
$stDOEV = 0$
$sDOp = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$dOp = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$$\begin{array}{l}
\textit{ReadShockAbsorbers_FT_7_TCASE} \\
\hline
\textit{ReadShockAbsorbers_FT_7} \\
\hline
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = forward \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadShockAbsorbers_FT_8_TCASE} \\
\hline
\textit{ReadShockAbsorbers_FT_8} \\
\hline
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = left \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadShockAbsorbers_FT_9_TCASE} \\
\hline
\textit{ReadShockAbsorbers_FT_9} \\
\hline
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = right \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\} \\
sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadShockAbsorbers_FT_13_TCASE} \\
\hline
\textit{ReadShockAbsorbers_FT_13} \\
\hline
sSA = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = forward \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\} \\
sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$$\begin{array}{l}
\textit{ReadShockAbsorbers_FT_14_TCASE} \\
\hline
\textit{ReadShockAbsorbers_FT_14} \\
\hline
sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\} \\
lgsfl = on \\
g? = left \\
v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\} \\
sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}
\end{array}$$

$ReadShockAbsorbers\_FT\_15\_TCASE$
$ReadShockAbsorbers\_FT\_15$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$lgsfl = on$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_16\_TCASE$
$ReadShockAbsorbers\_FT\_16$
$sSA = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_17\_TCASE$
$ReadShockAbsorbers\_FT\_17$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_18\_TCASE$
$ReadShockAbsorbers\_FT\_18$
$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$lgsfl = on$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto n), (s3 \mapsto n)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_19\_TCASE$
$ReadShockAbsorbers\_FT\_19$
$sSA = \{(forward \mapsto \{s1, s2, s3\}), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_20\_TCASE$
$ReadShockAbsorbers\_FT\_20$

$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1, s2, s3\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_21\_TCASE$
$ReadShockAbsorbers\_FT\_21$

$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \{s1, s2, s3\})\}$
$lgsfl = on$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto n)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_28\_TCASE$
$ReadShockAbsorbers\_FT\_28$

$sSA = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_29\_TCASE$
$ReadShockAbsorbers\_FT\_29$

$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

$ReadShockAbsorbers\_FT\_30\_TCASE$
$ReadShockAbsorbers\_FT\_30$

$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\}$
$lgsfl = on$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

ReadShockAbsorbers_FT_31_TCASE

$ReadShockAbsorbers\_FT\_31$

$sSA = \{(forward \mapsto \emptyset), (left \mapsto \{s1\}), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = forward$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

ReadShockAbsorbers_FT_32_TCASE

$ReadShockAbsorbers\_FT\_32$

$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \emptyset), (right \mapsto \{s1\})\}$
$lgsfl = on$
$g? = left$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

ReadShockAbsorbers_FT_33_TCASE

$ReadShockAbsorbers\_FT\_33$

$sSA = \{(forward \mapsto \{s1\}), (left \mapsto \{s1\}), (right \mapsto \emptyset)\}$
$lgsfl = on$
$g? = right$
$v? = \{(s1 \mapsto y), (s2 \mapsto y), (s3 \mapsto y)\}$
$sa = \{(forward \mapsto y), (left \mapsto y), (right \mapsto y)\}$

AnalogicalSwitchM_SP_29_TCASE

$AnalogicalSwitchM\_SP\_29$

$l20 = 0$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 1002$
$lHPCh = 1$
$as = y$

$AnalogicalSwitchM\_SP\_30\_TCASE$

$AnalogicalSwitchM\_SP\_30$

$l20 = 0$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 1001$
$lHPCh = 1$
$as = y$

$AnalogicalSwitchM\_SP\_59\_TCASE$

$AnalogicalSwitchM\_SP\_59$

$l20 = 1$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 1502$
$lHPCh = 0$
$as = n$

$AnalogicalSwitchM\_SP\_60\_TCASE$

$AnalogicalSwitchM\_SP\_60$

$l20 = 1$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 1501$
$lHPCh = 0$
$as = n$

$AnalogicalSwitchM\_DNF\_3\_TCASE$

$AnalogicalSwitchM\_DNF\_3$

$l20 = 0$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 0$
$lHPCh = 0$
$as = n$

AnalogicalSwitchM_DNF_4_TCASE

AnalogicalSwitchM_DNF_4

$l20 = 0$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 0$
$lHPCh = 0$
$as = y$

AnalogicalSwitchM_DNF_5_TCASE

AnalogicalSwitchM_DNF_5

$l20 = 0$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 0$
$lHPCh = 0$
$as = y$

AnalogicalSwitchM_DNF_6_TCASE

AnalogicalSwitchM_DNF_6

$l20 = 0$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 0$
$lHPCh = 0$
$as = y$

AnalogicalSwitchM_DNF_7_TCASE

AnalogicalSwitchM_DNF_7

$l20 = 0$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 0$
$lHPCh = 0$
$as = y$

$AnalogicalSwitchM\_DNF\_8\_TCASE$

$AnalogicalSwitchM\_DNF\_8$

$l20 = 0$
$hPos = down$
$lgsfl = on$
$sAS = \emptyset$
$now = 0$
$lHPCh = 0$
$as = y$

$Valid\_SP\_26\_TCASE$

$Valid\_SP\_26$

$i? = \{(s3 \mapsto y)\}$

$Valid\_SP\_64\_TCASE$

$Valid\_SP\_64$

$i? = \emptyset$

$Valid\_SP\_94\_TCASE$

$Valid\_SP\_94$

$i? = \{(s3 \mapsto n)\}$

$Down1\_SP\_17\_TCASE$

$Down1\_SP\_17$

$l20 = 0$
$stEV = 0$
$hPos = down$
$st = d0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 201$
$lHPCh = 0$

$Down1\_SP\_18\_TCASE$

$Down1\_SP\_18$

$l20 = 0$
$stEV = 0$
$hPos = down$
$st = d0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 200$
$lHPCh = 0$

$Down1\_SP\_29\_TCASE$

$Down1\_SP\_29$

$l20 = 0$
$stEV = 1$
$hPos = down$
$st = d0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 202$
$lHPCh = 0$

$Down1\_SP\_30\_TCASE$

$Down1\_SP\_30$

$l20 = 0$
$stEV = 1$
$hPos = down$
$st = d0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 201$
$lHPCh = 0$

$Down1\_DNF\_2\_TCASE$
$Down1\_DNF\_2$
$l20 = 0$
$stEV = 0$
$hPos = down$
$st = d0$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Down1\_DNF\_3\_TCASE$
$Down1\_DNF\_3$
$l20 = 0$
$stEV = 0$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Down1\_DNF\_4\_TCASE$
$Down1\_DNF\_4$
$l20 = 0$
$stEV = 0$
$hPos = up$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$

$Down1\_DNF\_5\_TCASE$

$Down1\_DNF\_5$

$l20 = 0$
$stEV = 1$
$hPos = down$
$st = init$
$gEV = pressing$
$stGEV = 0$
$spGEV = 0$
$now = 0$
$lHPCh = 0$