

DESCRIPTION

Title of Invention

VALIDATION PROCESSING DEVICE, VALIDATION PROCESSING METHOD,
AND PROGRAM

[Technical Field]

[0001]

The present invention relates to a validation processing device, a validation processing method, and a program.

The present application claims priority with respect to Japanese Patent Application No. 2018-206518 filed in Japan on November 1, 2018, the contents of which are incorporated herein by reference.

[Background Art]

[0002]

Patent Document 1 describes that model checking is used to comprehensively validate the operation logic of a data processing system.

[Citation List]

[Patent Document]

[0003]

[Patent Document 1] Japanese Unexamined Patent

Application Publication No. 2008-071135

[Summary of Invention]

[Technical Subject]

[0004]

For example, in a case of validating the operation logic of a relay circuit in model checking, it is not enough to validate the basic operation logic of the relay circuit, and it is necessary to perform validation, considering that troubles may occur in the signal lines and circuit elements included in the relay circuit.

[0005]

Considering that troubles of signal lines and circuit elements (for example, contact (short circuited / interference) of signal lines, disconnection, failure of circuit elements) may occur simultaneously and asynchronously regardless of the basic operation logic of the relay circuit, in model checking, in addition to the state transitions that may occur during the basic operation, it is necessary to comprehensively validate all combinations of troubles that may occur from each state during the basic operation.

[0006]

However, in such a case, even if a counterexample including a combination of a plurality of troubles caused in each signal line and each circuit element included in

the relay circuit is output, some of the combinations of troubles may include troubles that do not necessarily contribute to (that are non-critical to) leading to an unsafe event.

[0007]

In other words, model checking comprehensively checks the conditions (patterns) leading to unsafe events by expressing all possible states of the model to be checked with logical expressions using a binary decision diagram (BDD) or the like. The process leading to the unsafe event may include state transitions that are not necessarily critical. Therefore, there is no choice but to perform counterexample interpretation on counterexamples that may include non-critical troubles regarding unsafe events, which imposes a heavy load on the work of counterexample interpretation of model checking.

[0008]

According to at least one embodiment of the present invention, there is a subject about providing of a validation processing device, a validation processing method, and a program capable of reducing the load required for the work of counterexample interpretation of model checking.

[Subject to be solved]

[0009]

According to a first aspect of the present invention, a validation processing device includes: a processing unit that performs model checking on a model to be checked; and a selection unit that selects, on the basis of a result of the model checking, one element among elements that have undergone a state change in a process leading to an unsafe event. The processing unit further performs model re-checking on a model to be checked which is obtained by excluding the one element.

[0010]

According to a second aspect of the present invention, in a case where the model to be checked has not led to the unsafe event in the model re-checking, the processing unit returns the one element to the model to be checked, and performs model re-checking on a model to be checked which is obtained by excluding an element, which is different from the one element, from the model to be checked.

[0011]

According to a third aspect of the present invention, in a case where the model to be checked has led to the unsafe event in the model re-checking, the processing unit does not return the one element to the model to be checked, and performs model re-checking on a model to be checked which is obtained by excluding an element, which is

different from the one element, from the model to be checked.

[0012]

According to a fourth aspect of the present invention, the processing unit performs the model re-checking within a range equal to or less than the number of transition steps leading up to the unsafe event specified in a first model checking for the model to be checked.

[0013]

According to a fifth aspect of the present invention, the above-mentioned validation processing device further includes a specification unit that specifies a shortest path from an element extracted on the basis of a result of the model re-checking to an element included in a definition of the unsafe event.

[0014]

According to a sixth aspect of the present invention, the above-mentioned validation processing device further includes a calculation unit that calculates a probability of leading the model to be checked to the unsafe event from an initial state, on the basis of a trouble occurrence probability defined for each element.

[0015]

According to the seventh aspect of the present

invention, in a case where the probability is lower than a predetermined determination threshold value, the selection unit omits processing of selecting the one element on the basis of a result of the model checking.

[0016]

According to a seventh aspect of the present invention, the validation processing method includes: a step of performing model checking on a model to be checked; a step of selecting, on the basis of a result of the model checking, one element among elements that have undergone a state change in a process leading to an unsafe event; and a step of performing model re-checking on a model to be checked which is obtained by excluding the one element.

[0017]

According to an eighth aspect of the present invention, a program causes a computer of a validation processing device to execute: a step of performing model checking on a model to be checked; a step of selecting, on the basis of a result of the model checking, one element among elements that have undergone a state change in a process leading to an unsafe event; and a step of performing model re-checking on a model to be checked which is obtained by excluding the one element.

[Advantageous Effects of Invention]

[0018]

According to each aspect of the invention described above, the load required for the work of counterexample interpretation of model checking can be reduced.

[Brief Description of Drawings]

[0019]

Fig. 1 is a diagram showing a configuration of a validation processing device according to a first embodiment.

Fig. 2 is a diagram showing a functional configuration of a CPU of the validation processing device according to the first embodiment.

Fig. 3 is a diagram showing an example of a model to be checked according to the first embodiment.

Fig. 4 is a diagram showing a processing flow of the validation processing device according to the first embodiment.

Fig. 5 is a diagram for explaining in detail the operation of the validation processing device according to the first embodiment.

Fig. 6 is a diagram for explaining in detail the operation of the validation processing device according to the first embodiment.

Fig. 7 is a diagram showing a functional configuration of a CPU of a validation processing device

according to a second embodiment.

Fig. 8 is a diagram for explaining in detail the operation of the specification unit according to the second embodiment.

Fig. 9 is a diagram for explaining in detail the operation of the specification unit according to the second embodiment.

Fig. 10 is a diagram showing a functional configuration of a CPU of a validation processing device according to a third embodiment.

Fig. 11 is a diagram for explaining in detail the operation of the calculation unit according to the third embodiment.

Fig. 12 is a diagram for explaining in detail the operation of the calculation unit according to the third embodiment.

[Description of Embodiments]

[0020]

<First Embodiment>

Hereinafter, a validation processing device according to a first embodiment will be described with reference to Figs. 1 to 6.

[0021]

(Configuration of Validation Processing Device)

Fig. 1 is a diagram showing a configuration of the

validation processing device according to the first embodiment.

Fig. 2 is a diagram showing a functional configuration of a CPU of the validation processing device according to the first embodiment.

[0022]

As shown in Fig. 1, the validation processing device 1 includes a CPU 10, a memory 11, a display 12, an input device 13, and a storage 14, and is configured as a general computer.

[0023]

The memory 11 is a so-called main storage device, where commands and data for the CPU 10 to operate on the basis of a program are developed.

[0024]

The display 12 is a display device that visually displays information, and may be, for example, a liquid crystal display or an organic EL display.

[0025]

The input device 13 is an input device that receives the operation of the user of the validation processing device 1, and may be, for example, a general mouse, keyboard, touch sensor, or the like.

[0026]

The storage 14 is a so-called auxiliary storage

device, and may be, for example, a hard disk drive (HDD), a solid state drive (SSD), or the like. The storage 14 stores, for example, a model to be checked MOD indicating a relay circuit to be inspected.

[0027]

The CPU 10 is a processor that controls the entire operation of the validation processing device 1. As shown in Fig. 2, the CPU 10 according to the present embodiment exerts functions as a processing unit 100, a selection unit 101, and a reconstruction unit 102.

[0028]

The processing unit 100 performs model checking on the model to be checked MOD. The model checking performed here comprehensively checks the conditions (patterns) leading to unsafe events by expressing all possible states of the model to be checked with a logical expression by binary decision diagram (BDD) or the like. The model checking algorithm performed in this embodiment may be a well-known algorithm.

The model to be checked MOD is information that defines the operation logic of the system to be inspected (for example, a railway security system). In the model checking, the operation of the system is comprehensively validated in accordance with the operation logic defined herein.

The unsafe event is defined as a state in which the system to be inspected does not have to transition under any circumstances. For example, in a railway security system, the following states are defined as unsafe events: a state where "emergency brake does not work during automatic driving control of a vehicle"; a state where "a vehicle is traveling across a railroad but the crossing bar is not down"; and the like.

[0029]

The selection unit 101 selects one element among the elements that have undergone state change in the process leading to an unsafe event, on the basis of the result of the model checking performed by the processing unit 100. The "element" is a minimum unit that defines the operation logic and state of the model to be checked MOD, and is, for example, a signal line or a circuit element mounted on a relay circuit of a security system or the like. As will be described later, the "element" also includes a virtual element defined to simulate the operation of the signal line or circuit element mounted on the actual relay circuit, as well as the operation of the relay circuit in a case where a trouble occurs.

[0030]

In the process of model checking performed in this embodiment, the reconstruction unit 102 reconstructs the

model to be checked MOD in accordance with predetermined conditions.

[0031]

(Example of Model to be Checked)

Fig. 3 is a diagram showing an example of the model to be checked according to the first embodiment.

The model to be checked MOD shown in Fig. 3 simulates the operation logic of the relay circuit constituting the railway security system.

The wiring V and the wiring G shown in Fig. 3 are a power supply line and a ground line (ground), respectively. Further, elements A1, A2, ... are relay switches, and transition to an OFF state or an ON state in accordance with energization (0 (FALSE) = OFF / 1 (TRUE) = ON). Further, elements X1, X2, ... are manual switches, and transition to an OFF state or an ON state through a human operation (0 = OFF / 1 = ON).

Elements E1, E2, ... are virtual elements defined for reproducing troubles (disconnection and contact) that may occur in each signal line. For example, the element E1 is defined on a signal line connecting wiring V (power supply line) and element X1 (manual switch). This element E1 reproduces "occurrence of disconnection" as one of the troubles in the signal line (0 = disconnection / 1 = non-disconnection). Further, the two elements E2 and E3 are

defined on the signal line connecting the element X1 and the element X2 (manual switch). Among the elements, the element E2 reproduces "occurrence of disconnection" in the signal line (0 = disconnection / 1 = non-disconnection), and the element E3 reproduces "occurrence of contact with the power supply line" in the signal line (0 = non-contact / 1 = contact). Similarly, the two elements E4 and E5 are defined on the signal line connecting the element X2 and the element A1 (relay switch). Among the elements, the element E4 reproduces "occurrence of disconnection" in the signal line (0 = disconnection / 1 = non-disconnection), and the element E5 reproduces "occurrence of contact with the power supply line" in the signal line (0 = non-contact / 1 = contact).

[0032]

The actual model to be checked MOD is described by a logical expression (language). For example, the element A1 (relay switch) is described as in Expression (1) in consideration of troubles (disconnection, contact) that may occur in each signal line in addition to the manual switches X1 and X2.

[0033]

$$A1 = (E1 \ \& \ X1 \ \& \ E2 \ \& \ X2 \ \& \ E4) \ \text{or} \ (E3 \ \& \ X2 \ \& \ E4) \ \text{or} \ (E5) \ \dots \ (1)$$

[0034]

Other elements are described by the same logical expression.

[0035]

Expression (1) defines the state transition of the element A1 in a case where the element A1 (relay switch) itself does not have a trouble. In reality, since there is a possibility that a trouble may occur in the element A1 itself, the model to be checked MOD also includes a transition pattern in which the element A1 transitions to the OFF state or the ON state regardless of the logical expression of Expression (1). The same applies to the other elements A2, A3, ..., and the like.

Further, the elements X1, X2, ... which are manual switches, are elements each of which has a state transition according to a human operation. Thus, in model checking, similarly to the elements E1, E2, ... which define the occurrence of a trouble, the elements X1, X2, ... are defined as elements whose possible state transitions, which are simultaneous multiple and asynchronous, may occur at all timings.

[0036]

(Processing Flow of Validation Processing Device)

Fig. 4 is a diagram showing a processing flow of the validation processing device according to the first embodiment.

Figs. 5 and 6 are diagrams for explaining in detail the operation of the validation processing device according to the first embodiment.

The processing flow shown in Fig. 4 shows a series of processing flows of model checking for the model to be checked MOD.

[0037]

First, the processing unit 100 of the CPU 10 performs normal model checking on the model to be checked MOD (step S01). The model checking performed in step S01 will be described with reference to the state transition diagram shown in Fig. 5.

Each of a plurality of states STx shown in Fig. 5 indicates a state of the model to be checked MOD shown in Fig. 3. Each state of the model to be checked MOD is defined by a combination of each state (0/1) of the elements A1, A2, ..., X1, X2, ..., E1, E2, ... included in the model to be checked MOD. Among the elements, the state ST1 is an initial state of the model to be checked MOD (that is, the security system). The processing unit 100 comprehensively validates whether there is a state transition that reaches the initial state (state ST1) from the unsafe event (state ST2) in accordance with the logical expression (refer to Expression (1)) defined by the model to be checked MOD. The unsafe event (state ST2)

is defined as, for example, a state including a logical expression ($S1 \ \& \ T1 \ \& \ U1 = \text{TRUE}$) for other elements $S1$, $T1$, $U1$ included in the model to be checked MOD.

[0038]

In a case where there is a path (condition) from the unsafe event (state $ST2$) to the initial state (state $ST1$), the processing unit 100 lists the elements in which the trouble occurs in the process of reaching the unsafe event, and creates a trouble list (step $S02$). Fig. 6 shows an example of the trouble list created in step $S02$.

[0039]

In a case of randomly repeating state transitions from the initial state (state $ST1$) and resulting in leading to an unsafe event (state $ST2$), the processing unit 100 lists the elements (in other words, the elements in which the trouble occurs) that have undergone state change in the process leading to the unsafe event. Then, the processing unit 100 creates a trouble list L as shown in Fig. 6.

[0040]

The trouble list shown in Fig. 6 shows the list of elements, in which state transition occurs in each transition step from the state $ST1$ to the state $ST2$, and shows which value each element transitions to in which step. For example, the trouble list L shown in Fig. 6

shows leading to the unsafe event ($S1 \ \& \ T1 \ \& \ U1 = \text{TRUE}$) in the state transition from the initial state to the sixth step. Further, according to the trouble list L, it can be read that the element X1 transitions from FALSE to TRUE in the 4th step and transitions from TRUE to FALSE in the 5th step in the process leading to the unsafe event.

[0041]

However, the trouble list L shown in step S02 is just a list of elements that happen to have a state transition in the process of reaching an unsafe event ($S1 \ \& \ T1 \ \& \ U1 = \text{TRUE}$) from the initial state as a result of repeating random state transitions. Therefore, it is presumed that each element listed in the trouble list L includes an element that does not directly contribute to the unsafe event. Therefore, the validation processing device 1 according to the present embodiment further executes the following processing of steps S03 to S08.

[0042]

Specifically, the selection unit 101 of the CPU 10 selects one of the listed elements (step S03). As a simple example, in a case where there are five elements (E1, E2, E3, X1, X2) listed in the trouble list L, the selection unit 101 selects the element E1 as one of them, for example.

[0043]

Next, the reconstruction unit 102 of the CPU 10 creates (reconstructs) a model excluding the elements selected in step S03 from the model to be checked MOD (step S04). Hereinafter, the model created in step S04 will also be described as a "reconstructed model". For example, in a case where the element E1 is selected in step S03, the reconstruction unit 102 creates a reconstructed model in which the element E1 is excluded from the original model to be checked MOD.

[0044]

Next, the processing unit 100 performs model re-checking on the reconstructed model created in step S04 (step S05). In the above example, the reconstructed model does not include the element E1. Therefore, in the model checking performed in step S05, "occurrence of disconnection" on the signal line connecting the wiring V (power supply line, refer to Fig. 3) and the element X1 (manual switch, refer to Fig. 3) is not considered.

[0045]

The processing unit 100 outputs the result of the model checking performed again in step S05, and determines whether or not the same unsafe event ($S1 \ \& \ T1 \ \& \ U1 = \text{TRUE}$) as the unsafe event occurring in the first model checking (step S01) occurs (step S06).

In a case where the same unsafe event did not occur

(step S06; NO), the element selected in step S03 is excluded, and as a result, the unsafe event no longer occurs. Thus, it can be said that the excluded element is a critical element that contributes to the occurrence of unsafe events. Therefore, in this case, the reconstruction unit 102 returns the element selected in step S03 to the model to be checked MOD (step S07).

On the other hand, in a case where the same unsafe event occurs (step S06; YES), the excluded element is excluded because the unsafe event still occurs even though the element selected in step S03 is excluded. Thus, it can be said that the excluded element is an element that does not contribute to the occurrence of unsafe events (the element is not critical). Since it is desirable that such an element is excluded in the counterexample interpretation, the reconstruction unit 102 moves to the next step without returning the element selected in step S03 to the model to be checked MOD.

[0046]

Next, the selection unit 101 determines whether or not all the elements listed in the trouble list L in step S02 are selected (step S08). In a case where all the elements are not selected (step S08; NO), the selection unit 101 returns to step S03 and selects one element different from the previous step S03. Then, the

reconstruction unit 102 and the processing unit 100 repeat the processing of steps S04 to S07.

In a case where all the elements are selected (step S08; YES), the selection unit 101 outputs the elements listed in the trouble list L that remain in the model to be checked MOD (step S09).

[0047]

For example, among the elements (E1, E2, E3, X1, X2) listed in the trouble list L in step S02, the elements (E1, E2, X1) are excluded since the elements are not critical through the processing from step S03 to step S08. Then, the remaining elements (E3, X2) are output in step S09.

[0048]

(Effects and Advantages)

As described above, the validation processing device 1 according to the first embodiment includes: the processing unit 100 that performs model checking on the model to be checked MOD; and the selection unit 101 that selects, on the basis of a result of the model checking, one element among elements that have undergone state change in a process leading to the unsafe event. Then, the processing unit 100 performs model re-checking on the model to be checked (reconstructed model) excluding one element selected by the selection unit 101.

In such a manner, it is possible to extract only the

troubles that directly cause the unsafe event from the plurality of elements (troubles) shown in the model checking (refer to step S09 in Fig. 4). Therefore, the validator of the model to be checked MOD only needs to perform counterexample interpretation for the counterexample including only the critical elements contributing to the occurrence of the unsafe event. As a result, it is possible to reduce the work load of the counterexample interpretation.

[0049]

Further, in a case where the model to be checked has not led to the unsafe event in the model re-checking, the processing unit 100 according to the first embodiment returns the selected one element to the model to be checked MOD, and performs model re-checking on a model to be checked which is obtained by excluding an element, which is different from the one element, from the model to be checked MOD.

Further, in a case where the model to be checked has not led to the unsafe event in the model re-checking, the processing unit 100 according to the first embodiment does not return the selected one element to the model to be checked MOD, but performs model re-checking on a model to be checked which is obtained by excluding an element, which is different from the one element, from the model to

be checked MOD.

In such a manner, it is possible to automate the work of extracting only the critical elements related to unsafe events from the candidate elements (troubles) listed in the first trouble list L. As a result, it is possible to further reduce a load on the validator.

[0050]

(Modification Example of First Embodiment)

Although the validation processing device 1 according to the first embodiment has been described in detail above, the specific embodiment of the validation processing device 1 is not limited to the above, and various modifications in design thereof can be made without departing from the scope. For example, the processing unit 100 according to the modification example of the first embodiment may further have the following functions.

[0051]

The processing unit 100 according to the modification example of the first embodiment performs model re-checking (step S05 in Fig. 4) within the range equal to or less than the number of transition steps leading up to the unsafe event specified in the first model checking (step S01 in Fig. 4) for the model to be checked MOD.

For example, in the example of the trouble list L shown in Fig. 6, as a result of the first model checking, it is identified that the model to be checked MOD reaches an unsafe event ($S1 \ \& \ T1 \ \& \ U1 = \text{TRUE}$) through 6 state transitions (number of transition steps = 6). It should be noted that the first model checking is performed without setting a special upper limit on the number of transition steps for the purpose of covering the possibility of all state transitions leading to unsafe events.

Here, the number of transition steps leading to the unsafe event in the model re-checking (step S05 in Fig. 4) for the model to be checked MOD with the reduced number of elements is not greater than the number of transition steps leading to the unsafe event in model checking (step S01 in Fig. 4) for the model to be checked MOD before the number of elements is reduced. Therefore, in the model re-checking performed in step S05 of Fig. 4, the number of transition steps found in the model checking performed in step S01 of Fig. 4 may be set as an upper limit and then the model re-checking may be performed. In such a manner, the number of transition steps to be validated in the second and subsequent model checkings on the model to be checked MOD is limited. Therefore, the processing load and processing time required for the second and subsequent

model checkings can be reduced.

[0052]

<Second Embodiment>

Next, the validation processing device according to the second embodiment will be described with reference to Figs. 7 to 9.

[0053]

(Configuration of Validation Processing Device)

Fig. 7 is a diagram showing a functional configuration of a CPU of the validation processing device according to the second embodiment.

As shown in Fig. 7, the CPU 10 according to the second embodiment further exerts a function as the specification unit 103.

The specification unit 103 specifies the shortest path from the element extracted on the basis of the result of the model re-checking (refer to step S09 in Fig. 4) to the element included in the definition of the unsafe event.

The processing performed by the specification unit 103 may be, for example, processing which is automatically executed after the series of processing shown in Fig. 4 is completed.

[0054]

(Processing of Shortest Path Specification Unit)

Figs. 8 and 9 are diagrams for explaining in detail

the operation of the shortest path specification unit according to the second embodiment.

[0055]

As a result of the processing flow shown in Fig. 4, it is assumed that elements (E3, X2) are extracted as critical elements leading to the unsafe event (step S09 in Fig. 4). In this case, as shown in Fig. 8, the shortest path specification unit 103 specifies a shortest path until the state transitions (troubles) respectively occurring in the extracted elements E3 and X2 cause the respective state transitions of the elements (elements S1, T1, and U1) included in the definition of the unsafe event.

[0056]

More specifically, as shown in Fig. 9, the shortest path described above is specified with reference to a logic sheet (set of logical expressions) that defines the model to be checked MOD. Here, Fig. 9 shows an example of processing for specifying the shortest path until the state transition occurring in the element X2 causes the state transition in the element S1.

As shown in Fig. 9, first, the shortest path specification unit 103 refers to the logic sheet and specifies a logical expression including the element X2 on the right side. According to the logical expressions LC1 and LC2 specified herein, the elements B1 and B3 are

specified as elements affected by the state transition of the element X2 (step S11).

Next, the shortest path specification unit 103 refers to the logic sheet and specifies a logical expression including the element B1 on the right side (step S12). According to the logical expression LC3 specified herein, the element C1 is specified as an element affected by the state transition of the element B1 (step S13). Similarly, the shortest path specification unit 103 refers to the logic sheet and specifies a logical expression including the element B3 on the right side (step S14). According to the logical expression LC3 specified herein, the same element C1 as the element specified in step S13 is specified as an element affected by the state transition of the element B3 (step S15).

The shortest path specification unit 103 repeatedly executes the above processing from step S11 to step S15 until the element S1 is reached. As a result, the shortest path from the element X2 to the element S1 is specified.

Similarly, the shortest path specification unit 103 specifies the shortest path from the element X2 to the element T1, the shortest path from the element X2 to the element U1, the shortest path from the element E3 to the element S1, the shortest path from the element E3 to the

element T1, and the shortest path from the element E3 to the element U1.

[0057]

(Effects and Advantages)

As described above, according to the validation processing device 1 according to the second embodiment, it is possible to automatically specify the shortest path from a critical trouble (element) that causes an unsafe event to an actual unsafe event. Further, due to the effect obtained in the first embodiment, the shortest path specified by the shortest path specification unit 103 is only the shortest path (critical path) directly connected to the unsafe event, and does not include the shortest path from the element which does not contribute to the occurrence of the unsafe event. As a result, the validator only needs to consider countermeasures for the critical path specified by the shortest path specification unit 103. Therefore, the work load required for counterexample validation can be further reduced.

[0058]

The validation processing device 1 according to the modification example of the second embodiment may further have a function of displaying, as animation, a figure in which a trouble occurring in the relay circuit propagates through the critical path, on a separately provided

circuit diagram (refer to Fig. 3). This allows the validator to more intuitively understand the process through which an unsafe event occurs.

[0059]

<Third Embodiment>

Next, the validation processing device according to a third embodiment will be described with reference to Figs. 10 to 12.

[0060]

(Configuration of Validation Processing Device)

Fig. 10 is a diagram showing a functional configuration of a CPU of the validation processing device according to the third embodiment. As shown in Fig. 10, the CPU 10 according to the third embodiment further exerts a function as a calculation unit 104.

The calculation unit 104 calculates a probability (reaching probability) of leading the model to be checked MOD to the unsafe event from the initial state, on the basis of the trouble occurrence probability λ defined in advance for each element X1, X2, ... E1, E2, ... A1, A2, The trouble occurrence probability λ is a probability that a trouble occurs in each element within a unit time (for example, within 1 hour). It is assumed that such a trouble occurrence probability λ is specified in advance for each element on the basis of past operation

results and simulations.

[0061]

(Processing of Calculation Unit)

Figs. 11 to 12 are diagrams for explaining in detail the operation of the calculation unit according to the third embodiment.

Fig. 11 shows an example of a failure model defined for each element of the model to be checked MOD. Fig. 11 is a failure model showing "occurrence of contact", which is one of the troubles that occur in the signal line. According to the failure model shown in Fig. 11, from a state where no contact occurs (False), the signal line remains in the same state (False) with a probability of " $1-\lambda$ ", and transitions to a state where contact occurs with a probability of " λ " (True). Then, once the contact occurs, the signal line constantly (with a probability of 1) remains in the same state (True). Although not shown, a failure model is individually defined for "occurrence of disconnection" in the signal line. Further, a failure model for the element X1 which is a manual switch, a failure model for the element A1 which is a relay switch, and the like are also defined in the same manner.

[0062]

Fig. 12 is a diagram for specifically explaining the processing of the calculation unit 104. The calculation

unit 104 according to the present embodiment calculates the probability of leading to occurrence of the unsafe event shown in the trouble list L after the trouble list L (Fig. 6) created by the processing unit 100 is created (after step S02 in Fig. 4).

Specifically, the calculation unit 104 calculates the probability that the first state transition shown in the trouble list L occurs (state transition probability λ_1). Here, the calculation unit 104 calculates the state transition probability λ_1 using the failure model of the element changed in the first state transition (refer to Fig. 11). Similarly, the calculation unit 104 calculates the probability that the second state transition shown in the trouble list L occurs (state transition probability λ_2). Here, the calculation unit 104 calculates the state transition probability λ_2 using the failure model of the element changed in the second state transition. Hereinafter, in the same manner, the respective state transition probabilities λ_3 , λ_4 , ..., λ_6 leading to the unsafe event (state ST2) are calculated. Then, the calculation unit 104 calculates the probability that the state transition shown in the trouble list L occurs, that is, the probability ($\lambda_1 \times \lambda_2 \times \dots \times \lambda_6$) of leading to the unsafe event (state ST2) from the initial state (state ST1).

[0063]

(Effects and Advantages)

As described above, the validation processing device 1 according to the third embodiment further includes a calculation unit 104 that calculates a probability of leading the model to be checked MOD to the unsafe event from an initial state, on the basis of a trouble occurrence probability λ defined for each element.

As a result, in a case where the trouble list L is created as a result of the first model checking (step S01 in Fig. 4) for the model to be checked MOD, the sum of all the probabilities leading to the unsafe event can be obtained. Then, the priorities of counterexample interpretation and countermeasures can be appropriately determined in accordance with the probability. For example, in a case where the occurrence probability of the state transition shown in the trouble list created in step S02 of Fig. 4 is extremely low (in a case where the predetermined determination threshold value is not reached), the processing in and after step S03 of Fig. 4 is omitted. As a result, it is possible to immediately determine that another trouble list L is narrowed down or the processing moves to counterexample validation.

From the above, it is possible to further improve the efficiency of the work of counterexample

interpretation of the model to be checked MOD.

[0064]

In the first to third embodiments described above, the various processing processes of the CPU 10 described above are stored in a computer-readable recording medium in the form of a program, and various kinds of processing are performed by reading and executing the program through the computer. The computer-readable recording medium refers to a magnetic disk, a magneto-optical disk, a CD-ROM, a DVD-ROM, a semiconductor memory, or the like. Further, this computer program may be transferred to a computer via a communication line, and the computer receiving the transfer may execute the program.

[0065]

The above program may be for realizing a part of the above-mentioned functions. Further, a so-called difference file (difference program) may be used, which can realize the above-mentioned functions in combination with a program already recorded in the computer system.

[0066]

As described above, some embodiments according to the present invention have been described, but all of these embodiments are presented as examples and are not intended to limit the scope of the invention. These embodiments can be implemented in various other

embodiments, and various omissions, replacements, and modifications can be made without departing from the spirit of the invention. In a case where these embodiments and modifications thereof are included in the scope and spirit of the invention, similarly, the embodiments and modifications are included in the scope of the invention described in the claims and the equivalent scope thereof.

[Industrial Applicability]

[0067]

According to each aspect of the present invention, it is possible to reduce the load required for the work of counterexample interpretation of model checking.

[Reference Signs List]

[0068]

1: validation processing device

10: CPU

100: processing unit

101: selection unit

102: reconstruction unit

103: specification unit

104: calculation unit

11: memory

12: display

13: input device

14: storage

MOD: model to be checked

L: trouble list

Claims

[Claim 1]

A validation processing device comprising:

a processing unit that performs model checking on a model to be checked; and

a selection unit that selects, on the basis of a result of the model checking, one element among elements that have undergone a state change in a process leading to an unsafe event,

wherein the processing unit further performs model re-checking on a model to be checked which is obtained by excluding the one element.

[Claim 2]

The validation processing device according to claim 1, wherein in a case where the model to be checked has not led to the unsafe event in the model re-checking, the processing unit returns the one element to the model to be checked, and performs model re-checking on a model to be checked which is obtained by excluding an element, which is different from the one element, from the model to be checked.

[Claim 3]

The validation processing device according to claim 2, wherein in a case where the model to be checked has led to the unsafe event in the model re-checking, the processing unit does not return the one element to the model to be checked, and performs model re-checking on a model to be checked which is obtained by excluding an element, which is different from the one element, from the model to be checked.

[Claim 4]

The validation processing device according to any one of claims 1 to 3,

wherein the processing unit performs the model re-checking within a range equal to or less than the number of transition steps leading up to the unsafe event specified in a first model checking for the model to be checked.

[Claim 5]

The validation processing device according to any one of claims 1 to 4, further comprising a specification unit that specifies a shortest path from an element extracted on the basis of a result of the model re-checking to an element included in a definition of the unsafe event.

[Claim 6]

The validation processing device according to any one of claims 1 to 5, further comprising a calculation unit that calculates a probability of leading the model to be checked to the unsafe event from an initial state, on the basis of a trouble occurrence probability defined for each element.

[Claim 7]

The validation processing device according to claim 6, wherein in a case where the probability is lower than a predetermined determination threshold value, the selection unit omits processing of selecting the one element on the basis of a result of the model checking.

[Claim 8]

A validation processing method comprising:

a step of performing model checking on a model to be checked;

a step of selecting, on the basis of a result of the model checking, one element among elements that have undergone a state change in a process leading to an unsafe event; and

a step of performing model re-checking on a model to

be checked which is obtained by excluding the one element.

[Claim 9]

A program causing a computer of a validation processing device to execute:

a step of performing model checking on a model to be checked;

a step of selecting, on the basis of a result of the model checking, one element among elements that have undergone a state change in a process leading to an unsafe event; and

a step of performing model re-checking on a model to be checked which is obtained by excluding the one element.