1 Compositional Fault Analysis

1.1 Summary of the Algorithms

Max fault hypothesis for each top level property

(i) Step 1: Verify Nominal Model and Generate Minimal Inductive Validity Cores (MIVCs)

Assuming the nominal model is defined using AADL and AGREE and the nominal behaviors and top level safety properties are verified, the user must also define the faulty behavior using the Safety Annex. Each of the subcomponents of the system have faults defined and each top level property has a Max Fault Hypothesis statement (e.g. max N fault for $N \in \mathbb{N}$).

At this point, the user can select to perform compositional max fault verification and behind the scenes, the verification process is as follows. The verification proceeds in a top-down compositional approach and all properties at all layers are verified valid. In the IVC analysis, the model elements considered depend on the layer of the architecture currently being verified. For a leaf level, only fault activation literals are considered for the IVC analysis. At an intermediate layer, the current layer fault activation literals are considered as well as contracts (if there exists a subcomponent of this layer). During verification, the MIVCs are collected.

The MIVCs collected for each verification layer are the Minimal Unsatisfiable Subsets for the constraint system with the top level property set to false. For the bottom layers, the MIVCs only contain fault activation literals constrained to false. For the upper levels, the MIVCs contain both properties and fault activation literals constrained to false.

(ii) Step 2: Compute Minimal Cut Sets (MinCutSets) from the IVCs

For each of the MIVCs (MUSs) collected at each verification layer, we compute the MinCutSets from the IVCs from the bottom up. Using the Hitting Set Algorithm, the Minimal Correction Sets (MCSs) are collected. Then we transform the MCSs into MinCutSets.

Bottom layer: For each MCS of an upper level property, all fault activation literals in the MCS are constrained to true in order to obtain the MinCutSets for the violation of that upper level property. Since we are looking at only some maximum number of faults, we can eliminate from consideration the MCSs with cardinality greater than N from the Max N Hypothesis.

Upper layer: For each MCS, any element that is not a fault activation literal (i.e. any property violation) is replaced/inlined with the MinCutSet boolean expression obtained at the lower level for the violation of that property. If any of the MinCutSets from the lower level are empty, replace this property with

false (indicating that this property cannot happen). If any of the MinCutSets have cardinality greater than the threshold, we disregard these sets.

What remains is the MinCutSets with cardinality less than or equal to the max fault hypothesis for the violation of each top level property.

(iii) Step 3: Determine Compositional Verification Results for a Property

If no MinCutSets are obtained using Step 2, then the property is valid with this hypothesis.

If some MinCutSets are obtained using Step 2, then the property is invalid with this hypothesis and each of the MinCutSets shows a counterexample indicating that when these faults are active, the property is violated.

Probabilistic fault hypothesis for each top level property

(i) Step 1: Verify Nominal Model and Generate MIVCs

This process is the same as described in Step 1 for Max Fault Hypothesis algorithm.

(ii) Step 2: Compute Fault Combinations based on Fault Probabilities and Hypotheses

Using monolithic analysis, we collect all fault combinations whose probability exceeds or is equal to the specified threshold.

(iii) Step 3: Compute Minimal Cut Sets from the IVCs

For each of the MIVCs (MUSs) collected at each verification layer, we compute the MinCutSets from the IVCs from the bottom up. Using the Hitting Set Algorithm, the Minimal Correction Sets (MCSs) are collected. Then we transform the MCSs into MinCutSets.

Bottom layer: For each MCS of an upper level property, all fault activation literals in the MCS are constrained to true in order to obtain the MinCutSets for the violation of that upper level property. If a MinCutSet is not a subset of any fault combinations computed in the last step, we can eliminate this MinCutSet from consideration.

Upper layer: For each MCS, any element that is not a fault activation literal (i.e. any property violation) is replaced/inlined with the MinCutSet boolean expression obtained at the lower level for the violation of that property. If any of the MinCutSets from the lower level are empty, replace this property with *false* (indicating that this property cannot happen). If any of the MinCutSets have probability greater than the threshold, we disregard these sets.

What remains is the MinCutSets with probability equal to or greater than the probabilistic threshold specified for a given top level property.

(iv) Step 4: Determine Compositional Verification Results for a Property

If no MinCutSets are obtained using Step 2, then the property is valid with this hypothesis.

If some MinCutSets are obtained using Step 2, then the property is invalid with this hypothesis and each of the MinCutSets shows a counterexample indicating that when these faults are active, the property is violated.

1.2 Theory

Definitions:

Given a constraint system C where C is an ordered set of abstract constraints over some set of variables, $\{C_1, ..., C_n\}$. The satisfiability problem is in conjuntive normal form (CNF): $C = \wedge_{i=1,...,n} C_i$; and each C_i is a disjunction of literals $C_i = l_{i1} \vee ... \vee l_{ik_i}$ where each literal $l_{ij}$is either a Boolean variable x or its negation $\neg x$.

Satisfiability (SAT) : A CNF is satisfiable iff there exists an assignment of truth values to its variables such that the formula evaluates to true. If not, it is unsatisfiable (UNSAT).

Given a state space S, a transition system (I, T) consists of the initial state predicate $I : S \to \{0, 1\}$ and a transition step predicate $T : S \times S \to \{0, 1\}$. Reachability for (I, T) is defined as the smallest predicate $R : S \to \{0, 1\}$ which satisfies the following formulas:

$$\forall s.I(s) \Rightarrow R(s)$$
$$\forall s, s'.R \wedge T(s, s') \Rightarrow R(s')$$

A safety property $\mathcal{P} : S \to \{0, 1\}$ is a state predicate. A safety property $\mathcal{P}$ holds on a transition system (I, T) if it holds on all reachable states. More formally, $\forall s.R(s) \Rightarrow \mathcal{P}(s)$. When this is the case, we write $(I, T) \vdash \mathcal{P}$. Following Ghassabani, et. al. [**?**], we formalize IVCs as follows.

Definition 1. *Inductive Validity Core*
Let (I, T) be a transition system and let $\mathcal{P}$ be a safety property with $(I, T) \vdash \mathcal{P}$. Then $S \subseteq T$ is an inductive validity core for $(I, T) \vdash \mathcal{P}$ iff $(I, S) \vdash \mathcal{P}$.

Definition 2. *Minimal Inductive Validity Core*
An inductive validity core S for $(I, T) \vdash \mathcal{P}$ is minimal iff $!\exists S'.S' \subset S \ni (I, S') \vdash \mathcal{P}$.

Intuitively, this can be understood as the minimal set of elements such that the safety property $\mathcal{P}$ is proved.

MUS : Minimal Unsatisfiable Subset (MUS) M of a constraint system C is a subset $M \subseteq C$ such that M is UNSAT and $\forall c \in M$: $M \setminus \{c\}$ is SAT. This is the minimal explaination of the constraint systems infeasability.

The IVC problem is in essence a MUS problem. The original safety predicate $\mathcal{P}$ of a transition system C is negated by the model checker in order to find all MUSs of the system given $\neg\mathcal{P}$. Thus all IVCs found are the MUSs of the transition system with the negation of the safety predicate.

MSS : Maximal Satisfiable Subset (MSS) M of a constraint system C is a subset $M \subseteq C$ such that M is SAT and $\forall C \in C$: $M \cup \{c\}$ is UNSAT. If any element is added to an MSS, we get UNSAT results.

MCS : Minimal Correction Set (MCS) M of a constraint system C is a subset $M \subseteq C$ such that $C \setminus M$ is SAT and $\forall S \subset M : C \setminus M$ is UNSAT. A MCS can be seen to "correct" the infeasability of the constraint system.

MinCutSet: Minimal Cut Set is a minimal collection of faults that lead to the violation of the safety property (or in other words, lead to the top level event).

Hitting Set: Given a collection of sets K, a hitting set for K is a set $H \subseteq \cup_{S \in K} S$ such that $H \cap S \neq \emptyset$ for each $S \in K$. A hitting set for K is minimal if and only if no proper subset of it is a hitting set for K.

Using the algorithm to find all IVCs, we get all MUSs of C: the $UnsatCores$ of C and using the hitting set algorithm described by Reiter and Greiner, et. al., the MCSs of C are generated.

The MCSs describe the minimal set of model elements for which if constraints are removed, the constraint system is satisfied. For C, this corresponds to which faults are not constrained to inactive (and are hence active) and violated contracts which lead to the violation of the safety property. In other words, the minimal set of active faults and/or violated properties that lead to the top level event.

Theorem 1. *The unconstrained model elements found in the Minimal Correction Sets of a constraint system C are equivalent to the faults in the Minimal Cut Sets of the system.*

Proof. Part 1: Leaf level of system

(i) MCS $\subseteq$ MinCutSet:

Let $M \in MCS$. Then $C \setminus M$ is SAT $\wedge$ $\forall S \subset M$, $C \setminus S$ is UNSAT. Since C contains faults constrained to inactive and $\neg P$, then any unconstrained faults in

M cause $\neg P$ to occur. This is the definition of a Minimal Cut Set.

By minimality of the MCS, M is a minimal cut set for $\neg P$.

(ii) $MinCutSet \subseteq$ *MCS:*

Let $M \in MinCutSet$. Then all faults in M cause $\neg P$ to occur by definition. Thus, by removing the constraints of these faults in the constraint system C, we get a satisfiable constraint system with $\neg P$.

By minimality of $MinCutSet$, M is also minimal and thus is a minimal correction set.

Part 2: Intermediate level of system

(i) MCS $\subseteq$ MinCutSet:

Let $M \in MCS$. The elements of M contain unconstrained faults and/or violated contracts from the current level of analysis. In the case that M only contains unconstrained faults, the proof is the same as in the leaf level. In the case that one or more contracts appear in M, we make use of the assumption that the nominal model proves and all contracts hold in the absence of faults. Then if a contract is violated, it is due to the presence of faults in the lower level of the system. In this case, we replace the violated contract with the fault(s) that caused its violation and now M consists only of unconstrained faults. The rest of the proof remains the same as in the leaf level.

By minimality of the MCS, M is a minimal cut set for $\neg P$.

(ii) $MinCutSet \subseteq$ *MCS:*

Let $M \in MinCutSet$. Then all faults in M cause $\neg P$ to occur by definition. If all faults in M are from the current layer of analysis, the proof is done and is identical to the leaf layer. If one or more faults are defined in a lower level of the architecture, we can replace these faults with the contracts they will violate.

NOTE: I need to think more about this part of the proof and showing that any lower level faults that occur in a MinCutSet can be replaced with the contracts they violate AND these will be equivalent to the contracts found in the MCSs... This part needs more work.

By minimality of $MinCutSet$, M is also minimal and thus is a minimal correction set.

1.3 Example

Let P be our top level safety property: the "good" behavior we want to happen.

P = (pressure $>$ threshold) $\implies$ shut down command, i.e. shut down when we should.

Our model elements are: $F = \{f1, f2, f3\}$ corresponding to:
$f1$ = sensor 1 fault (stuck at low)
$f2$ = sensor 2 fault (stuck at low)
$f3$ = sensor 3 fault (stuck at low)

For the *no voting* implementation, each sensor can have a stuck at low fault and the system shuts down when one of the sensors indicates high pressure. The constraint system for this example corresponds to : $C = \{\neg f1, \neg f2, \neg f3, \neg P\}$. This constraint system is given to the SAT solver. Assuming that the nominal model holds (P and model elements are satisfied), we get an UNSAT result with this constraint system. The SAT solver provides all counterexamples in the form of IVCs which are our Minimal Unsatisfiable Subsets (MUSs). Since these are all of the sets that show $\neg P$ is UNSAT, they also are the sets that prove P. For example $IVC_1 = \{\neg f1\}$: if sensor 1 fault is inactive, we can prove P : this is the minimal explanation of infeasibility with respect to C.

In this case, the IVCs generated are:
$IVC_1 = \{\neg f1\}$, sensor 1 fault (stuck at low) = false,
$IVC_2 = \{\neg f2\}$, sensor 2 fault (stuck at low) = false,
$IVC_3 = \{\neg f3\}$, sensor 3 fault (stuck at low) = false.

MUSes are equivalent to the IVCs.
$MUS_1 = \{\neg f1\}$,
$MUS_2 = \{\neg f2\}$,
$MUS_3 = \{\neg f3\}$.

Now we look at Maximal Satisfiable Subsets (MSS). MSSs are the sets for which we have the maximal number of elements that will prove our constraint system. If we add anything to these sets, it becomes UNSAT. Since our original constraint system is of the form $C = \{\neg f1, \neg f2, \neg f3, \neg P\}$, we have:
MUS: P is SAT $\iff$ $\neg P$ is UNSAT
MSS: P is UNSAT $\iff$ $\neg P$ is SAT.

The Minimal Correction Sets (MCSs) are the complement of MSS relative to constraint system C. The MCSs describes the infeasibility of the system and works as a "correction" set to the problem. The MCSs describe the constraints that when removed from the constraint system, provide a satisfiable system. Furthermore, any strict subset of the MCSs, when removed from the constraint system, will provide an unsatisfiable system. It seems that we do not need to actually find the $MSSs$ in order to get their complement because of what is called a *hitting set*. Intuitively, a minimal hitting set has the minimal number of elements in it such that every set in that collection has something in common with the set its "hitting." Every MCS is a minimal hitting set of

its MUSes.

For us in this example, it is: MCS $= \{\neg f1, \neg f2, \neg f3\}$. This is the hitting set because if we take the intersection of every MUS with the MCS, it is nonempty and it is the minimal such set for which this is true. When the constraints on these model elements are removed, we get a constraint system that is satisfiable with regard to $\neg P$. Thus, this is the minimal set of elements for which the top level property occurs. What this means from the models perspective is that all three faults together cause the top level property to fail. Thus, when all three sensors have a fault which causes them to report low temp, we do not shut down when we should. Intuitively, it makes sense that this is the Minimal Cut Set because it is the minimal description of why the top level property fails.