Supporting Materials for Symbolic Repairs for GR(1) Specifications

Shahar Maoz
Tel Aviv University
Tel Aviv, Israel

Jan Oliver Ringert
University of Leicester
Leicester, UK

Rafi Shalom
Tel Aviv University
Tel Aviv, Israel

***Abstract*—This document provides supporting materials for the ICSE'19 paper by the authors titled "Symbolic Repairs for GR(1) Specifications" [5]. We show proofs for the soundness and completeness of the algorithm of GLASS repair (preliminaries, notation, and illustrating examples are in the paper). We further provide additional observations from the evaluation results.**

I. GLASS Repair Soundness and Completeness

A. *Problem Definition*

Intuitively, the repair problem takes as input an unrealizable GR(1) specification $\mathcal{S}$ and produces a set of assumptions that make $\mathcal{S}$ realizable.

A closer look reveals that some (repaired) specifications do not allow assumptions and guarantees to be satisfied together and realizability implies that the system forces the environment to violate assumptions. We say a specification $\mathcal{S}$ is γ-sat iff all existing assumptions and guarantees can be satisfied together, i.e., when $\gamma = \theta^e \wedge \theta^s \wedge \mathbf{G}(\rho^e \wedge \rho^s) \wedge \bigwedge_{i \in 1..n} \mathbf{GF}(J_i^e) \wedge \bigwedge_{j \in 1..m} \mathbf{GF}(J_j^s)$ is satisfiable.

We consider repairs leading to γ-unsat specifications not useful. Adding assumptions to a γ-unsat specification preserves γ-unsatisfiability and we thus consider it as unrepairable.

Formally, we define the repair problem as follows: given an unrealizable and γ-sat specification $\mathcal{S}$, find additional assumptions that will make $\mathcal{S}$ realizable and γ-sat. We call such a set of assumptions a repair.

B. *GLASS Repair Algorithm*

Algorithm 1 symbolically computes up to three kinds of assumptions (in this order): a safety assumption ρ', a set J' of up to $|J^s|$ justice assumptions, and an initial assumption θ'.

First, Alg. 1 symbolically computes all states $collSat$ from which the system and environment player can collaboratively satisfy all safety and justice assumptions and guarantees. Then it computes all transitions $badEnvTrans$ that start from $collSat$ and are legal for the environment (ρ^e) but force all legal system choices (ρ^e) to leave the states $collSat$. The removal of these *bad* environment transitions is the safety assumption ρ'. This part is a symbolic version of safety assumption computation from [2] adapted for satisfying all assumptions and guarantees.

If the assumption ρ' does not make the specification realizable, Alg. 1 computes for every justice guarantee J_j^s the states J from which the environment can prevent J_j^s while satisfying all justice assumptions. The negations of these states are the justice assumptions J'. The new justice assumptions in J' force the environment to always eventually leave the states where each J_j^s can be prevented. Technically, the LTL formula in Alg. 1, l. 8 is a sub-formula of the negation of φ^{sr} evaluated inside algorithms for computing GR(1) CS.

Finally, if the assumptions ρ' and J' are not enough to make the specification realizable, Alg. 1, l. 15 restricts the initial environment states to θ' where a legal system choice exists to reach a system winning state, i.e., a state to realize the repaired specification.

C. *Soundness, Completeness, and Minimality*

We discuss the soundness, completeness, and minimality of the three parts of Alg. 1 in Lemmas 1-3. The algorithm is sound and complete for unrealizable and γ-sat GR(1) specification as formalized in Theorem 1. The minimality of the computed assumptions in the repair is local to each assumption (see Sect. I-C1).

Lemma 1 (Safety). *Given a γ-sat GR(1) specification $\mathcal{S}$ Alg. 1, lines 1-3 compute the unique minimal (in number of transitions) non-restrictive safety assumption ρ' to allow the system to remain in states where γ without initial constraints is satisfiable. Assumption ρ' maintains γ-satisfiability.*

Proof. Note that the set $collSat$ contains states where γ without initial constraints is satisfiable. Thus, $collSat$ is not empty and contains initial states because $\mathcal{S}$ is γ-sat.

Sound: The transitions $badEnvTrans$ either are a deadlock for the system ($\neg\rho^s$) or lead out of $collSat$ and thus either eventually lead to a deadlock or they lead to states where a justice assumption or guarantee cannot be satisfied. By removing environment transitions out of $collSat$ the system can ensure to stay in $collSat$ (γ without initial constraints). Also note that by staying in $collSat$ the safety guarantees ρ^s are realized.

Complete: The system cannot be forced to violate safety guarantees or leave states where γ without initial constraints is satisfiable because all remaining environment transitions stay in $collSat$.

Minimal: Any set of transitions smaller than ρ' would allow an environment transition from a state in $collSat$ to either force the system into a deadlock or to leave $collSat$.

γ-sat: The set $collSat$ includes all states visited in a computation satisfying γ (the LTL formula in l. 1 is γ without initial guarantees and assumptions). The new assumption ρ'

Algorithm 1 GLASS computes a repair given an unrealizable and γ-sat GR(1) specification $\mathcal{S}$.

```
Require: An unrealizable and $\gamma$-sat $GR(1)$ specification $\mathcal{S}$
Ensure: $GR(1)$ assumptions that makes $\mathcal{S}$ realizable and $\gamma$-sat
 1: $collSat \leftarrow$ collWinStates( $\mathbf{G}(\rho^e \wedge \rho^s) \wedge (\bigwedge_{i \in 1..n} \mathbf{GF} J_i^e \wedge \bigwedge_{j \in 1..m} \mathbf{GF} J_j^s)$ )
 2: $badEnvTrans \leftarrow collSat \wedge \rho^e \wedge (\rho^s \Rightarrow \neg prime(collSat))|_{\forall \mathcal{Y}'}$
 3: $\rho' \leftarrow \mathbf{G}\ \neg badEnvTrans$
 4: if isRealizable( addAssumptions($\mathcal{S}, \{\rho'\}$) ) then
 5:     return $\{\rho'\}$
 6: end if
 7: for $J_j^s \in J^s$ do
 8:     $J \leftarrow$ envWinStates( $\mathbf{F}((\mathbf{H}(\rho^e \wedge \neg badEnvTrans)) \wedge \neg \rho^s) \vee \bigwedge_{i \in 1..n} \mathbf{GF} J_i^e \wedge \mathbf{G} \neg J_j^s$ )
 9:     add $\mathbf{GF}\ \neg J$ to $J'$ unless $J = \mathtt{F}$
10: end for
11: if isRealizable( addAssumptions($\mathcal{S}, \{\rho'\} \cup J'$) ) then
12:     return $\{\rho'\} \cup J'$
13: end if
14: $win \leftarrow$ sysWinStates( addAssumptions($\mathcal{S}$ without ini, $\{\rho'\} \cup J'$) )
15: $\theta' \leftarrow (\theta^e \wedge \theta^s \wedge win)|_{\exists \mathcal{Y}}$
16: return $\{\theta', \rho'\} \cup J'$
```

only removes edges $badEnvTrans$ that lead out of $collSat$. Thus, any computation satisfying γ for $\mathcal{S}$ also satisfies γ for **addAssumptions**($\mathcal{S}, \{\rho'\}$). □

Note that this part of Alg. 1 is adapted from the minimal safety assumption computation from [2]. A major difference to [2] is our change to collaboratively satisfying assumptions and guarantees instead of collaboratively satisfying the specification, i.e., an implication relation between assumptions and guarantees. A positive effect of this difference is that the new assumption will also avoid transitions to states where justice assumptions are no longer realizable, i.e., it will avoid a case of non-well-separation [4].

Lemma 2 (Justice)**.** *Given a γ-sat GR(1) specification $\mathcal{S}' =$ **addAssumptions**($\mathcal{S}, \{\rho'\}$) where the system can realize all safety guarantees, Alg. 1, ll. 8-9 compute a justice assumption that allows the system to realize the justice guarantees J_j^s from all states where the system can realize the safety guarantees.*

Proof. It is possible to satisfy all justice guarantees J_j^s because $\mathcal{S}'$ is γ-sat.

Sound: Alg. 1, l. 8 computes the states J where the environment can satisfy all justice assumptions and prevent the system from satisfying justice guarantee J_j^s. Because of determinancy for LTL winning conditions (the LTL formula in l. 8 is an inner part of the negated winning condition φ^{sr} as used in CS computation) in $\neg J$ the system can satisfy J_j^s if all justice assumptions are satisfied. Thus, the assumption $\mathbf{GF}\ \neg J$ ensures that the system can satisfy $\mathbf{GF}\ J_j^s$ or wins the GR(1) game because the environment violates assumptions.

Complete: If J is empty J_j^s can be realized without additional assumptions because $\mathcal{S}'$ is γ-sat and the system can satisfy all safety guarantees. The set J would contain all states only if $J_j^s = \mathtt{F}$, however $\mathcal{S}'$ is γ-sat. Thus, the computation is complete: either J_j^s can be realized to begin with or a repairing assumption $\mathbf{GF}\ \neg J$ is computed.

γ-sat: Every computation π that satisfies γ for $\mathcal{S}'$ clearly satisfies $\mathbf{GF} J_j^s$ (part of γ). The computation π also satisfies the new assumption $\mathbf{GF}\ \neg J$ because $J \subseteq \neg J_j^s$ (right side of l. 8 where left disjunct is empty following Lem. 1), i.e., $J_j^s \subseteq \neg J$ and thus satisfying $\mathbf{GF} J_j^s$ implies satisfying $\mathbf{GF}\ \ \neg J$. □

Note the special case where the environment can always prevent J_j^s and satisfy all justice assumptions. In this case the computed assumption is identical to the guarantee $\mathbf{GF}\ J_j^s$.

On minimality: The set of states J is not minimal for the property described in Lem. 2. Specifically, Alg. 1, l. 8 computes some states that are not necessary to ensure that the system can realize $\mathbf{GF}\ J_j^s$. An example for this are states on the attractor to a cycle where the environment satisfies all justice assumptions but prevents J_j^s.

Lemma 3 (Initial)**.** *Given a γ-sat GR(1) specification $\mathcal{S}$ where the system can realize all safety guarantees according to Lem. 1 and all justice guarantees according to Lem. 2, Alg. 1, ll. 14-15 compute the unique minimal (in number of states) initial assumption to make $\mathcal{S}$ realizable.*

Proof. It is possible to satisfy all guarantees from some initial states because $\mathcal{S}$ is γ-sat. In the given specification **addAssumptions**($\mathcal{S}, \{\rho'\} \cup J'$) the system can further realize all safety guarantees and all justice guarantees from states where γ without initial constraints is satisfiable (see Lem. 1 and Lem. 2).

Sound: Alg. 1, l. 14 computes the states from where the system can realize all guarantees while ignoring initial guarantees (the computation directly follows [1]). These states are intersected in l. 15 with common initial states and system variables are existentially quantified out, i.e., for environment choices θ' the system can choose at least one assignment to variables $\mathcal{Y}$ that constitutes a winning state. By adding θ' as an initial assumption the system wins from all initial environment choices.

Complete: The algorithm makes all specifications realizable: $\theta^e \wedge \theta^s \wedge win$ is never empty because the specification is γ-sat.

Minimal: Any set of states smaller than θ' would leave an initial choice for the environment to force the system outside of its winning states.

γ-sat: The set $\theta^e \wedge \theta' \wedge \theta^s$ is not empty because $\mathcal{S}$ was γ-sat. Any computation satisfying γ for $\mathcal{S}$ also satisfies γ with the addition of θ' because win contains all states where γ without initial constraints is satisfiable (see Lem. 1 and Lem. 2). □

We combine all lemmas in Thm. 1 about the soundness and completeness of GLASS repair from Alg. 1.

Theorem 1 (GLASS Repair is Sound and Complete). *Given an unrealizable and γ-sat GR(1) specification $\mathcal{S}$, Alg. 1 produces a set of assumptions that make $\mathcal{S}$ realizable and γ-sat.*

Proof. We denote the set of assumptions returned by the algorithm A.

Sound and Complete: The algorithm always returns a set of assumptions. If Alg. 1 returns A in line 5 then **addAssumptions**$(\mathcal{S}, A)$ is realizable (see line 4). If Alg. 1 returns A in line 12 then **addAssumptions**$(\mathcal{S}, A)$ is realizable (see line 11). Finally, if Alg. 1 returns A in line 16, then **addAssumptions**$(\mathcal{S}, A)$ is realizable, see Lemma 1-3.

γ-sat: Any set of assumptions returned by Alg. 1 keeps $\mathcal{S}$ γ-sat because all individual parts of the algorithm from Lemma 1-3 keep γ-sat specifications γ-sat. □

1) Minimality Discussion: Note the notion of minimality in Lem. 1 with respect to states satisfying γ without initial constraints ($collSat$) rather than with respect to realizability. First, some of these states might not be reachable in any execution of the system and environment. When looking for minimality for realizability, the transitions from these states would not be necessary to exclude. Second, there might be some transitions in $badEnvTrans$ that lead to states where assumptions have to be violated. Again, for realizability, these transitions could be removed from $badEnvTrans$. Nevertheless, we believe that the assumptions computed for γ might lead to better specifications that prevent the environment from “deciding to violate” assumptions.

Finally, note that minimality for the assumptions of GLASS is defined with respect to realizability from all possible initial states (see Lem 3) whereas minimality for repairs can also be seen in a “local” way as minimal assumptions that allow for realizability from at least one initial state.

II. Evaluation Results: Additional Observations

We present additional observations from our evaluation.

Number of repairs. JVTS-Repair and AMT13 may produce more than one repair for a given specification. However, some of these may be very similar to one another. In this context, we consider that a repair $r1$ is weaker than a repair $r2$ iff the conjunction of the safeties in $r2$ implies the conjunction of safeties in $r1$, and for each justice in $r1$ there is at least one justice or safety in $r2$ that implies it[1]. This weakness relation is a partial order on repairs.

We computed the number of weakest repairs, for repairs found within the timeout. For SYNTECH15-UNREAL and SYNTECH15-1UNREAL, the number of weakest repairs found by JVTS-Repair ranges from 1 to 27 (avg. 5.5, median 3) and from 1 to 49 (avg. 11.06, median 4), resp. For SYNTECH15-UNREAL and SYNTECH15-1UNREAL, the number of weakest repairs found by AMT13 ranges from 1 to 3 (avg. 2.3, median 3) and from 1 to 23 (avg. 3.18, median 1), resp.

We leave the selection between or prioritization of different candidate repairs to future work.

Well-separation. In well-separated specifications [3], [4] the environment cannot be forced to violate its assumptions. Non-well-separation allows for unwanted system implementations. Thus, when a specification is originally well-separated it may be preferable to have a repair that maintains well-separation. In this regard, GLASS and JVTS-Repair have success rates of 68% and 57% respectively. Interestingly, AMT13 only repaired well-separated specifications of our corpus, and maintained well-separation for 50%. Also note that all three algorithms repair the specifications RG1, RG2, and LIFT, yet only JVTS-Repair found a well-separated repair for all three, while the other algorithms found a well-separated repair only for two of them.

[1] JVTS-Repair and AMT13 produce repairs that do not include initial assumptions.

References

[1] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'ar. Synthesis of Reactive(1) Designs. *J. Comput. Syst. Sci.*, 78(3):911–938, 2012.

[2] K. Chatterjee, T. A. Henzinger, and B. Jobstmann. Environment assumptions for synthesis. In F. van Breugel and M. Chechik, editors, *CONCUR 2008 - Concurrency Theory, 19th International Conference, CONCUR 2008, Toronto, Canada, August 19-22, 2008. Proceedings*, volume 5201 of *Lecture Notes in Computer Science*, pages 147–161. Springer, 2008.

[3] U. Klein and A. Pnueli. Revisiting synthesis of GR(1) specifications. In *Haifa Verification Conference (HVC)*, volume 6504 of *LNCS*, pages 161–181. Springer, 2010.

[4] S. Maoz and J. O. Ringert. On well-separation of GR(1) specifications. In *FSE*, pages 362–372. ACM, 2016.

[5] S. Maoz, J. O. Ringert, and R. Shalom. Symbolic repairs for GR(1) specifications. In *ICSE*, pages 1016–1026. IEEE / ACM, 2019.