A Appendix

Proposition 1. *Given a temporal user query ρ and a trained MARL policy π, if Algorithm 1 returns YES, then the query ρ must be feasible under the policy π; otherwise, Algorithm 1 generates correct and complete explanations $\mathcal{E}$.*

Proof. We prove the following two cases.
Case 1: When Algorithm 1 returns YES, the policy abstraction MMDP $\mathcal{M}$ or the updated MMDP $\mathcal{M}'$ satisfies the PCTL* formula φ encoding the user query ρ, indicating that there must exist a path through $\mathcal{M}$ or $\mathcal{M}'$ conforms with ρ. By construction, every abstract MMDP transition $(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ in $\mathcal{M}$ or $\mathcal{M}'$ with non-zero probability maps to at least one sampled decision $(\mathbf{x}, \mathbf{a}, \mathbf{x}')$ of the given MARL policy π. Thus, there must exist an execution of policy π that conforms with the user query ρ. By definition, the user query ρ is feasible under the given MARL policy π.
Case 2: Algorithm 1 returns explanations $\mathcal{E}$ generated via Algorithm 3. As described in Section 4.4, Algorithm 3 terminates when all failures in the user query ρ have been explained and fixed. Given a finite-length temporal query ρ, there is a finite number of failures. For any failure in the query, if the target states set $\mathcal{V}$ is non-empty, then the failure must be fixable using a Quine-McCluskey minterm that represents a target state where the failed task is completed. If $\mathcal{V}$ is empty, then the failure is removed from the query. Thus, the termination of Algorithm 3 is guaranteed. By definition, the generated explanations are *correct* (i.e., identifying the causes of one or more failures in ρ) and *complete* (i.e., finding the reasons behind all failures in ρ). $\square$