Definition 1 (Execution)**.** *Let π be an error trace of length n. An execution of π is a sequence of states $s_0, s_1 ... s_n$ such that $s_i, s_{i+1} \models T$, where T is the transition formula of $\pi[i]$.*

Definition 2 (Blocking Execution)**.** *An execution of a trace π of size n is called a blocking execution, if there exists a sequence of states $s_0, s_1 ... s_j$ where $i < j \leq n$ such that $s_i, s_{i+1} \models T$ where T is the transition formula of $\pi[i]$ and there exists an assume statement in the trace π at position j such that $s_j \not\models guard(\pi[j])$*

Definition 3 (Relevance of an assigning statement)**.** *Let $\pi = \langle st_1, ..., st_n \rangle$ be an error trace of length n where st_i is an assigning statement at position i that assigns a new value to some variable x. The statement st_i is relevant if there exists an execution $s_1, ... s_{n+1}$ of π and some value v such that every execution of the trace $\langle x := v; \pi[i+1, n] \rangle$ starting in s_i has a blocking execution.*

Algorithm 1 Relavance of an assigning statement

procedure RELEVANCE
  $trace \leftarrow$ Error trace π of length n
  $relevantStatements \leftarrow [\ ]$
  for $i = n$ to 1 **do**
    $Q \leftarrow \neg wp(false; trace(i+1, n))$
    $P \leftarrow wp(Q; trace(i))$
    $relevance \leftarrow checkUnsatCore(P, trace(i), Q)$
    if $relevance =$ "$unsat$" and $trace(i)$ in "$unsatCore$" **then**
      $relevantStatements.append(trace(i))$
  return $relevantStatements$

In the algorithm , we check the relevance of a statement by checking if the triple $(P, \pi[i], \neg Q)$ is unsatisfiable and $\pi[i]$ is in the unsatisfiable core. We can do this by checking if $P \not\subseteq WP(Q; havoc(x))$.

Theorem 1 (Equivalence of relevance). *Let $\pi = \langle st_1, ..., st_i, ..., st_n \rangle$ be an error trace of length n and $\pi[i]$ be an assigning statement at position i , which assigns a new value to some variable x . Let $P = \neg WP(False; \pi[i, n]) \cap SP(True; \pi[1, i-1])$ be a set of bireachable states at position i and $Q = \neg WP(False; \pi[i+1, n])$ be the coreachable states at position $i+1$. The statement $\pi[i]$ is relevant iff:*

$$P \not\subseteq WP(Q, havoc(x))$$

Proof. Let $\mathcal{D}$ be the domain of the variable x.
"$\Rightarrow$"
If $\pi[i]$ is relevant, then

$$P \not\subseteq WP(Q; havoc(x))$$

Obviously all the transitions from the states in $WP(Q; havoc(x))$ ends up in Q. Relevancy of $\pi[i]$ implies that there is a state in $s \in P$ such that there is a transition from s to $\neg Q$. That would mean:

$$P \not\subseteq WP(Q; havoc(x))$$

"$\Leftarrow$"
$\pi[i]$ is relevant, if:

$$P \not\subseteq WP(Q; havoc(x))$$

We know that $WP(Q; havoc(x))$ is the set of states from which all transitions end up in Q. The above non implication shows the existence of a state s in P such that $s \notin WP(Q; havoc(x))$ from which there is a transition to $\neg Q$. This shows the existence of a value $v \in \mathcal{D}$ that we can assign to x such that if we replace $\pi[i]$ with $x := v$, then every execution is becoming blocking. Also, from our assumption, it is clear that there exits an execution till P, since P is not empty. $\square$