―――――――――――――――――――― MODULE *Trace* ――――――――――――――――――――

CONSTANT *Module*

CONSTANT *InitState*

CONSTANT *NextState*

$Init \triangleq$
   LET $IOUtils \triangleq$ INSTANCE $IOUtils$
   IN
      LET $ret \triangleq IOUtils!IOExecTemplate(\langle$ "rm", "%s.log"$\rangle, \langle Module \rangle)$
      IN   TRUE

$Log(name, args) \triangleq$
   LET $IOUtils \triangleq$ INSTANCE $IOUtils$
      $Json \quad \triangleq$ INSTANCE $Json$
   IN
      LET $init \quad \triangleq InitState$
           $next \quad \triangleq NextState$
           $trace \triangleq [action \mapsto name,$
                   $args \quad \mapsto args,$
                   $init \quad \mapsto init,$
                   $next \quad \mapsto [k \in \{k \in$ DOMAIN $next : next[k] \neq init[k]\} \mapsto next[k]]]$
           $ret \quad \triangleq IOUtils!IOExecTemplate(\langle$ "/bin/sh", "-c", "echo '%s' >> %s.log"$\rangle, \langle Json!ToJsonObject($
      IN $ret.exitValue = 0$

$Step(name, action, args) \triangleq$
   $\wedge action$
   $\wedge action \Rightarrow Log(name, args)$