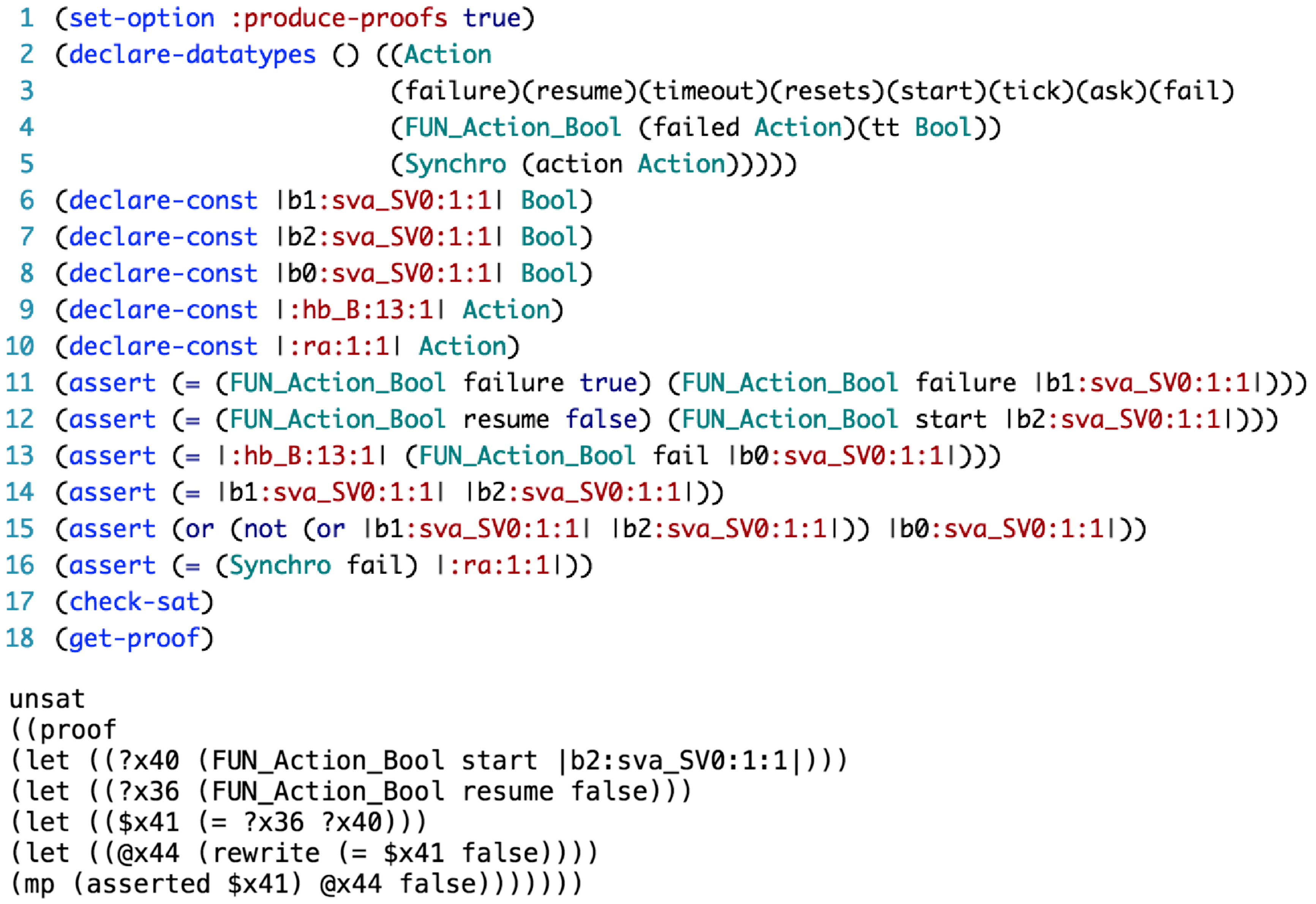

```
(set-option :produce-proofs true)
(declare-datatypes () ((Action
                        (failure)(resume)(timeout)(resets)(start)(tick)(ask)(fail)
                        (FUN_Action_Bool (failed Action)(tt Bool))
                        (Synchro (action Action)))))
(declare-const |b1:sva_SV0:1:1| Bool)
(declare-const |b2:sva_SV0:1:1| Bool)
(declare-const |b0:sva_SV0:1:1| Bool)
(declare-const |:hb_B:13:1| Action)
(declare-const |:ra:1:1| Action)
(assert (= (FUN_Action_Bool failure true) (FUN_Action_Bool failure |b1:sva_SV0:1:1|)))
(assert (= (FUN_Action_Bool resume false) (FUN_Action_Bool start |b2:sva_SV0:1:1|)))
(assert (= |:hb_B:13:1| (FUN_Action_Bool fail |b0:sva_SV0:1:1|)))
(assert (= |b1:sva_SV0:1:1| |b2:sva_SV0:1:1|))
(assert (or (not (or |b1:sva_SV0:1:1| |b2:sva_SV0:1:1|)) |b0:sva_SV0:1:1|))
(assert (= (Synchro fail) |:ra:1:1|))
(check-sat)
(get-proof)
```

```
unsat
((proof
(let ((?x40 (FUN_Action_Bool start |b2:sva_SV0:1:1|)))
(let ((?x36 (FUN_Action_Bool resume false)))
(let (($x41 (= ?x36 ?x40)))
(let ((@x44 (rewrite (= $x41 false))))
(mp (asserted $x41) @x44 false)))))))
```