

HUPYY TEMPORAL - SMT-LIB VERIFICATION REPORT

Generated: 2025-11-04 09:44:50
Query ID: query_1762278290
Status: UNSAT
Execution Time: 19 ms

1. PROBLEM STATEMENT

Based on the company policy document, can Marcus Webb (E-4782) perform WRITE operations on the Production Database at 11:47 PM EST on March 14, 2025 during the P1 production outage?

The additional necessary data is in
/Users/alexanderfedin/Projects/happy/cofounder/hupyy-temporal/data/free-form/adhoc/testfile.txt

2. PHASE ANALYSIS (AI CONVERSION)

PHASE 1: PROBLEM COMPREHENSION

Problem Type: access-control
Domain: Corporate system access control with security policy hierarchy
External References: /Users/alexanderfedin/Projects/happy/cofounder/hupyy-temporal/data/free-form/adhoc/testfile.txt
Reference Status: all-loaded
Complete Problem: Based on the company policy document, can Marcus Webb (E-4782) perform WRITE operations on the Production Database at 11:47 PM EST on March 14, 2025 during the P1 production outage? The policy document provides complete information about access rules, security exceptions, employee records, and security compliance records showing Marcus Webb is under active investigation with ALL SYSTEM ACCESS SUSPENDED.
Complexity: medium
Data Inventory (CRITICAL for verification queries):
- **Data Sources Available:** testfile.txt containing: base access rules, security exceptions policy, employee records for Marcus Webb, security compliance records showing active investigation, access attempt log
- **Query Type:** verification-from-data (asking "Can Marcus perform this specific action?" based on provided facts)
- **Data Extraction Plan:**
 * Marcus Webb's role: Senior Engineer -> FACT (from employee records)
 * Marcus Webb's clearance level: Senior -> FACT (from employee records)
 * Marcus Webb's on-call status: YES (March 10-16, 2025) -> FACT (from employee records)
 * Marcus Webb under investigation: YES (active since March 12, 2025) -> FACT (from security compliance records)
 * Investigation suspends ALL access: YES -> FACT (from restriction field and policy section 4)
 * Time of access attempt: March 14, 2025, 11:47 PM EST -> FACT
 * Requested action: WRITE access to Production Database -> FACT

??
PHASE 2: DOMAIN MODELING
??

PHASE 2: DOMAIN MODEL

Entities
Variables:
- marcus: Employee -- Marcus Webb (E-4782)
- can_write: Bool -- Whether Marcus can perform WRITE operations on Production DB
Constants:
- role_marcus = "Senior Engineer" -- Marcus's role from employee records
- clearance_marcus = "Senior" -- Marcus's clearance level from employee records
- oncall_marcus = true -- Marcus is on-call rotation (March 10-16, 2025)
- under_investigation = true -- Marcus is under active investigation (since March 12, 2025)

- access_time = "2025-03-14 23:47 EST" -- Time of access attempt
- during_p1_outage = true -- P1 production outage is occurring

Functions/Relations:
- None needed (using boolean logic for access rules)

Constraints

- 1. Natural: Employees with clearance "Senior" or higher have WRITE access at any time
Formal: has_senior_clearance -> has_write_access_by_clearance
Entities: clearance_marcus, has_write_access_by_clearance
- 2. Natural: Employees in on-call rotation have EMERGENCY OVERRIDE access 24/7
Formal: oncall_marcus -> has_emergency_override
Entities: oncall_marcus, has_emergency_override
- 3. Natural: Any employee under active investigation has ALL SYSTEM ACCESS SUSPENDED, overriding all other rules
Formal: under_investigation -> notcan_access_any_system
Entities: under_investigation, can_access_any_system
- 4. Natural: Access suspension overrides all permissions including emergency override
Formal: under_investigation -> notcan_write
Entities: under_investigation, can_write

Ground Truth (from provided data files/logs)
CRITICAL: Distinguish FACTS (from data) vs UNKNOWNNS (not provided)

FACTS to Assert (extracted from data):
- role_is_senior_engineer = true (from testfile.txt line 14: "Role: Senior Engineer")
- has_senior_clearance = true (from testfile.txt line 15: "Clearance Level: Senior")
- is_on_call = true (from testfile.txt line 17: "On-Call Rotation: YES (March 10-16, 2025)")
- is_under_investigation = true (from testfile.txt line 23: "Status: Active")
- access_suspended_by_policy = true (from testfile.txt line 25: "ALL SYSTEM ACCESS SUSPENDED per Policy Section 4")
- requesting_write_access = true (from testfile.txt line 29: "Requested Action: WRITE access")
- is_p1_outage = true (from testfile.txt line 31: "Incident: P1 Production Outage")
- access_time_is_2347 = true (from testfile.txt line 27: "March 14, 2025, 11:47 PM EST")

UNKNOWNNS (not in data, will be declared as variables):
- None (all relevant facts are provided in the data)

Data Extraction Notes:
- This is a verification query asking whether Marcus can perform a specific action
- All relevant facts about Marcus's status and the access attempt are provided
- The critical fact is that Marcus is under active investigation with ALL ACCESS SUSPENDED

Query
Question: Can Marcus Webb perform WRITE operations on the Production Database at the specified time?
Approach: negation-based-proof
- We want to prove Marcus CANNOT write (the answer should be NO based on the investigation)
- Encoding: Assert that Marcus CAN write, then check-sat
- If UNSAT -> Marcus cannot write (correct, due to investigation)
- If SAT -> Marcus can write (incorrect, would violate security policy)

Selected Approach: negation-based-proof
Encoding Plan: Assert (can_write = true) to try to find a case where Marcus can write. UNSAT will prove he cannot.

??
PHASE 3: LOGIC SELECTION
??

PHASE 3: LOGIC SELECTION

Theory Analysis
- Quantifiers: NO -- All facts are about specific entities (Marcus Webb)
- Uninterpreted Functions: NO -- Using only boolean propositions and logical connectives
- Arrays: NO -- No array operations needed
- Arithmetic: NO -- No numeric calculations required
- Strings: NO -- String values are used as labels but no string operations
- Datatypes: NO -- Only using booleans

Decision

Selected Logic: `QF_LIA`

Justification:

Since we only need propositional logic (booleans and logical connectives), and no quantifiers, the simplest appropriate logic is QF_LIA (Quantifier-Free Linear Integer Arithmetic). Even though we're not using arithmetic, QF_LIA supports propositional logic and is widely supported by SMT solvers. We could also use QF_UF but QF_LIA is simpler and sufficient.

Alternatives Rejected:

- ALL: Too general, includes unnecessary theories for this simple boolean problem
- QF_UF: Would work but unnecessary since we have no uninterpreted functions
- SAT: Too restrictive, SMT-LIB requires a logic declaration

??

PHASE 4: SMT-LIB ENCODING

??

```smt2

```
;; =====
;; SMT-LIB v2.7 Encoding
;; Logic: QF_LIA
;; Problem: Access control verification for Marcus Webb's write access during investigation
;; =====
```

```
(set-logic QF_LIA)
(set-option :produce-models true)
(set-option :produce-unsat-cores true)
```

```
;; =====
;; SECTION 1: GROUND TRUTH (from data)
;; =====
;; These are FACTS extracted from provided data files/logs.
;; DO NOT leave these as free variables!
;; Each fact should reference its source from Phase 2.
```

```
;; From testfile.txt Employee Records (lines 14-17)
(declare-const role_is_senior_engineer Bool)
(assert role_is_senior_engineer) ; Line 14: "Role: Senior Engineer"
```

```
(declare-const has_senior_clearance Bool)
(assert has_senior_clearance) ; Line 15: "Clearance Level: Senior"
```

```
(declare-const is_on_call Bool)
(assert is_on_call) ; Line 17: "On-Call Rotation: YES (March 10-16, 2025)"
```

```
;; From testfile.txt Security Compliance Records (lines 23-25)
(declare-const is_under_investigation Bool)
(assert is_under_in...
```

### 3. GENERATED SMT-LIB CODE

Logic: QF\_LIA

```
(set-logic QF_LIA)
(set-option :produce-models true)
(set-option :produce-unsat-cores true)

(declare-const role_is_senior_engineer Bool)
(assert role_is_senior_engineer)

(declare-const has_senior_clearance Bool)
(assert has_senior_clearance)

(declare-const is_on_call Bool)
(assert is_on_call)

(declare-const is_under_investigation Bool)
```

```

(assert is_under_investigation)

(declare-const access_suspended_by_policy Bool)
(assert access_suspended_by_policy)

(declare-const requesting_write_access Bool)
(assert requesting_write_access)

(declare-const is_pl_outage Bool)
(assert is_pl_outage)

(declare-const has_write_by_clearance Bool)
(declare-const has_emergency_override Bool)
(declare-const all_access_suspended Bool)
(declare-const can_write Bool)

(assert (=> has_senior_clearance has_write_by_clearance))
(assert (=> is_on_call has_emergency_override))
(assert (=> is_under_investigation all_access_suspended))
(assert (=> access_suspended_by_policy all_access_suspended))
(assert (= can_write
 (and (not all_access_suspended)
 (or has_write_by_clearance has_emergency_override)))))

(assert can_write)
(check-sat)
(get-model)

```

## 4. VERIFICATION RESULTS

Status: UNSAT

Wall Time: 19 ms

## 5. HUMAN-READABLE EXPLANATION

Proof:

- Employee: Marcus Webb (E-4782), Senior Engineer
- Request: WRITE access to Production Database
- Time: March 14, 2025, 11:47 PM EST (P1 outage)
  
- Policy Rule #2: Senior clearance grants WRITE access [x]
- Policy Rule #3: On-call rotation grants emergency override (March 10-16) [x]
  
- Security Investigation: Active (INV-2025-089)
- Investigation Start: March 12, 2025, 2:00 PM EST
- Policy Rule #4: Active investigation suspends ALL access
- Policy Rule #4: Overrides emergency access privileges
  
- Verification: Investigation suspension active [x]
- Verification: ALL access suspended overrides WRITE permission [x]
- Verification: ALL access suspended overrides emergency override [x]
  
- DENIED: Marcus Webb cannot perform WRITE operations due to active security investigation overriding all access privileges

## 6. TECHNICAL DETAILS (APPENDIX)

### cvc5 Standard Output:

```
unsat
(error "cannot get model unless after a SAT or UNKNOWN response.")
```

*Generated by Hupyy Temporal - Hupyy Powered SMT Verification*