HUPYY TEMPORAL - SMT-LIB VERIFICATION REPORT

Generated: 2025-11-04 10:43:41

Query ID: query_1762281821

Status: UNSAT

Execution Time: 23 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/hapyy/cofounder/hupyy-temporal/data/free-form/adhoc/testfile.txt

2. PHASE ANALYSIS (AI CONVERSION)

```
Now let me proceed with the structured 5-phase analysis:

## PHASE 1: PROBLEM COMPREHENSION

**Problem Type:** access-control
**Domain:** Enterprise system access control with security policy enforcement
**External References:** /Users/alexanderfedin/Projects/hapyy/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 testfile.txt contains:
- Base access rules for Production Database access
- Security exceptions overriding all other rules
- Employee record for Marcus Webb (E-4782)
- Security compliance records showing active investigation
- Access attempt log for the specific incident

**Complexity:** medium

**Data Inventory (CRITICAL for verification queries):**
- **Data Sources Available:** testfile.txt containing policy rules, employee records, security compliance records, and access logs
- **Query Type:** verification-from-data - "Can Marcus Webb perform WRITE operations?" requires verifying against factual data
- **Data Extraction Plan:**
  * Marcus Webb's role: Senior Engineer (FACT from employee records)
  * Marcus Webb's clearance: Senior (FACT from employee records)
  * Marcus Webb's on-call status: YES for March 10-16, 2025 (FACT)
  * Investigation status: ACTIVE since March 12, 2025 (FACT from security compliance records)
  * Access suspension: ALL SYSTEM ACCESS SUSPENDED (FACT from security compliance records)
  * Timestamp: March 14, 2025, 11:47 PM EST (FACT from access log)
  * P1 outage status: Active (FACT from access log)

## PHASE 2: DOMAIN MODEL

### Entities
**Variables:**
- employee_id: String -- Marcus Webb's ID (E-4782)
- employee_role: String -- Role of the employee (Senior Engineer)
- clearance_level: String -- Security clearance level (Senior)
- is_on_call: Bool -- Whether employee is on call rotation (true for March 10-16)
- under_investigation: Bool -- Whether employee is under active investigation (true)
- access_suspended: Bool -- Whether all system access is suspended (true)
```

- request_time: Time -- Time of access request (11:47 PM EST, March 14, 2025)
- is_business_hours: Bool -- Whether request is during business hours (false for 11:47 PM)
- has_write_access: Bool -- Whether the employee can perform WRITE operations (to be determined)

Constants:
- business_hours_start = 9:00 AM EST
- business_hours_end = 6:00 PM EST
- business_days = Monday-Friday

Functions/Relations:
- None (all relationships encoded as direct constraints)

### Constraints

1. Natural: Employees with role containing "Engineer" have READ access during business hours
   Formal: (role contains "Engineer") and is_business_hours -> has_read_access
   Entities: employee_role, is_business_hours

2. Natural: Employees with clearance "Senior" or higher have WRITE access at any time
   Formal: (clearance_level = "Senior" or clearance_level > "Senior") -> has_write_access_normally
   Entities: clearance_level, has_write_access_normally

3. Natural: Employees on call rotation have EMERGENCY OVERRIDE access 24/7
   Formal: is_on_call -> has_emergency_override
   Entities: is_on_call, has_emergency_override

4. Natural: Employees under investigation have ALL ACCESS SUSPENDED, overriding all other rules
   Formal: under_investigation -> access_suspended
   Entities: under_investigation, access_suspended

5. Natural: If access is suspended, no operations are allowed
   Formal: access_suspended -> nothas_write_access
   Entities: access_suspended, has_write_access

### Ground Truth (from provided data files/logs)

CRITICAL: Distinguish FACTS (from data) vs UNKNOWNS (not provided)

FACTS to Assert (extracted from data):
- employee_role = "Senior Engineer" (from testfile.txt line 14)
- clearance_level = "Senior" (from testfile.txt line 15)
- is_on_call = true (from testfile.txt line 17, March 10-16 includes March 14)
- under_investigation = true (from testfile.txt line 23, Status: Active)
- access_suspended = true (from testfile.txt line 25, ALL SYSTEM ACCESS SUSPENDED)
- request_date = March 14, 2025 (from testfile.txt line 27)
- request_time = 11:47 PM EST (from testfile.txt line 27)
- is_business_hours = false (11:47 PM is outside 9 AM - 6 PM)

UNKNOWNS (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 where we need to check if Marcus Webb can perform WRITE operations
- All necessary facts are explicitly stated in the provided testfile.txt
- The critical fact is that Marcus Webb is under active investigation with ALL SYSTEM ACCESS SUSPENDED

### Query

Question: Can Marcus Webb (E-4782) perform WRITE operations on the Production Database at 11:47 PM EST on March 14, 2025?
Approach: negation-based-proof
  - We want to prove that Marcus Webb CANNOT perform WRITE operations
  - We'll assert that he CAN perform WRITE operations (has_write_access = true)
  - If UNSAT, it proves he cannot; if SAT, he can

Selected Approach: negation-based-proof
Encoding Plan: Assert (has_write_access = true) and check if it's satisfiable. UNSAT means Marcus cannot perform WRITE operations.

## PHASE 3: LOGIC SELECTION

### Theory Analysis
- Quantifiers: NO -- All constraints are propositional about specific entities
- Uninterpreted Functions: NO -- All relationships are direct boolean constraints
- Arrays: NO -- No array operations needed
- Arithmetic: NO -- While we have time comparisons, they're already evaluated to boolean (is_business_hours)
- Strings: NO -- String values are used only for equality checks, can be abstracted to booleans
- Datatypes: NO -- Simple boolean propositions suffice

### Decision
Selected Logic: `QF_LIA`

Justification:
Since we're dealing with pure propositional logic (all constraints are boolean), we could use QF_UF or even simpler. However, QF_LIA (Quantifier-Free Linear Integer Arithmetic) provides a good foundation if we need to add any numeric comparisons and is well-supported by solvers. Since we have no quantifiers and only boolean constraints, QF_LIA will work perfectly.

Alternatives Rejected:
- ALL: Too general, unnecessarily complex for this simple propositional problem
- QF_UF: Could work but QF_LIA is more standard for access control problems

## PHASE 4: SMT-LIB ENCODING

```smt2
;; =========================================================================
;; SMT-LIB v2.7 Encoding
;; Logic: QF_LIA
;; Problem: Access control verification for Marcus Webb
;; =========================================================================

(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 testfile.txt

;; From testfile.txt line 14: Role = Senior Engineer
(declare-const is_engineer Bool)
(assert (= is_engineer true))  ; Role contains "Engineer"

;; From testfile.txt line 15: Clearance Level = Senior
(declare-const has_senior_clearance Bool)
(assert (= has_senior_clearance true))

;; From testfile.txt line 17: On-Call Rotation = YES (March 10-16, 2025)
(declare-const is_on_call Bool)
(assert (= is_on_call true))  ; March 14 is within March 10-16

;; From testfile.txt line 23: Investigation Status = Active
(declare-const under_investigation Bool)
(assert (= under_investigation true))

;; From testfile.txt line 25: ALL SYSTEM ACCESS SUSPENDED
(declare-const access_suspended Bool)
(assert (= access_suspended true))

;; From testfile.txt line 27: Time = 11:47 PM EST
(declare-const is_business_hours Bool)
(assert (= is_business_hours false))  ; 11:47 PM is outside 9 AM - 6 PM

;; ============================================
;; SECTION 2: DERIVED LOGIC & CONSTRAINTS
;; ===================...
```

3. GENERATED SMT-LIB CODE

Logic: QF_LIA

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

;; GROUND TRUTH from testfile.txt
(declare-const is_engineer Bool)
(assert (= is_engineer true))

(declare-const has_senior_clearance Bool)
(assert (= has_senior_clearance true))

(declare-const is_on_call Bool)
(assert (= is_on_call true))

(declare-const under_investigation Bool)
(assert (= under_investigation true))

(declare-const access_suspended Bool)
(assert (= access_suspended true))

(declare-const is_business_hours Bool)
(assert (= is_business_hours false))

;; DERIVED LOGIC
(declare-const has_write_access Bool)
(declare-const has_read_access_from_role Bool)
(assert (=> (and is_engineer is_business_hours) has_read_access_from_role))

(declare-const has_write_access_from_clearance Bool)
(assert (=> has_senior_clearance has_write_access_from_clearance))

(declare-const has_emergency_override Bool)
(assert (=> is_on_call has_emergency_override))

(assert (=> under_investigation access_suspended))
(assert (=> access_suspended (not has_write_access)))

(assert (=> has_write_access
            (and (or has_write_access_from_clearance has_emergency_override)
                 (not access_suspended))))

;; QUERY: Assert Marcus can write (expect UNSAT to prove he cannot)
(assert has_write_access)
(check-sat)
(get-model)
```

4. VERIFICATION RESULTS

Status: UNSAT

Wall Time: 23 ms

5. HUMAN-READABLE EXPLANATION

Proof:

- Marcus Webb (E-4782) status: Engineer with Senior Clearance
- Request time: 11:47 PM EST (outside business hours)
- Marcus is currently on-call for P1 production outage
- Senior clearance grants: WRITE access to Production Database [x]
- On-call status grants: Emergency override privileges [x]
- Security investigation status: ACTIVE
- Company Policy ?7.3: All access suspended during active investigation
- Access suspension overrides: ALL permissions including emergency access
- Verification: has_write_access requires NOT access_suspended

- Current state: access_suspended = true [ ]
- DENIED: Active security investigation blocks all database access

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