TASK 02

DAY 01:

Action: Transferring Money.

a. Same Action, Same Actor, Different Authority

Case 1: Action is allowed
1. Actor: User linked to a source account
2. Goal: Transfer a specific amount to a destination account
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition:
The user has valid authority to initiate the transfer
Source account balance ≥ transfer amount at request time
5. Result if conditions pass: Amount is deducted from source and amount is added to destination
6. Result if conditions fails: Balance unchanged

Case 2: action is not allowed
1. Actor: User linked to a same source account
2. Goal: Transfer same amount to a destination account
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition:
The user does not have valid authority to initiate the transfer
5. Result if conditions pass: not reached
6. Result if conditions fails: Balance unchanged

b. Authority Without Identity Change
In this case, the user identity remains unchanged:
-That same user had authority to transfer
-The user attempts the same transfer again but authority has been expired or revoked

Decision condition failure:
Authority is no longer valid at time of request.

Outcome:

Transfer refused even though user and action are same

c. Delegation Without Transfer

In this case , execution is delegated but authority is not transferred:
-The user which is Actor A has authority on source account
-Another person Actor B initiates the transfer on behalf of Actor A
-Actor B does not have authority on source account.

Failure:
If Actor B is treated as authority holder then the transfer maybe incorrectly allowed or denied.

d. Reflection
Observation:
- The actor and action remain the same. The refusal occurs only because authority is not present.
- Authority cannot be decided from identity alone. It must be valid at the moment the action is evaluated.
- Execution and authority are separate. Delegation of execution does not imply delegation of authority.
- Authority must be checked every time an action is performed. If it is assumed instead of verified, the system can make wrong decisions even when all other conditions seem right.

DAY 02:

a. Time Window Validity.
(Taking as the maintenance is going on)
Case 1: Action is valid (Before the time window)
1. Actor: User linked to a source account
2. Goal: Transfer a specific amount to a destination account
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition:
The user has valid authority to initiate the transfer
Source account balance ≥ transfer amount at request time
The transfer request is outside the restricted time window

5. Result if conditions pass: Amount is deducted from source and amount is added to destination
6. Result if conditions fails: Balance unchanged

Case 2: Action is Invalid (During the Time Window)
1. Actor: User linked to a source account
2. Goal: Transfer a specific amount to a destination account
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition:
The user has valid authority to initiate the transfer
Source account balance ≥ transfer amount at request time
The transfer request is within the restricted time window
5. Result if conditions pass: not reached
6. Result if conditions fails: Balance unchanged

Case 3: Action is valid again (After the Time Window)
1. Actor: User linked to a source account
2. Goal: Transfer a specific amount to a destination account
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition:
The user has valid authority to initiate the transfer
Source account balance ≥ transfer amount at request time
The transfer request is outside the restricted time window
5. Result if conditions pass: The amount is transferred successfully
6. Result if conditions fails: Balance unchanged

b. Evaluation Timing
this case show how same action becomes invalid depending on when evaluated
Actor: User linked to a source account
Goal: Transfer a specific amount to a destination account
Required setup: Source and destination account and transfer mechanism should exist
Possible Scenarios:
-At time of initiation the balance is sufficient, all condition are valid
-At time of transfer balance changed, balance is not sufficient.

Decision Condition: Source account balance is equal to or greater than the transfer amount
Result:
The transfer is refused because the condition is evaluated at execution time, not request time.

c. State Drift
Actor: User linked to a source account
Goal: Transfer a specific amount to a destination account
Required setup: Source and destination account and transfer mechanism should exist
Scenario
- The user initiates a transfer request.
- At request time, the source account balance is sufficient.
- Before the transfer completes, another transaction reduces the balance.
- The balance is no longer sufficient.

Decision Condition: Source account balance is equal to or greater than the transfer amount
Result if conditions pass:not reached
Result if conditions fail: No money transfer

d. Reflection
- The action becomes invalid and valid again purely due to time, even though the actor, intent, and setup never change.
- Even though the action looked valid initially, evaluation timing changes the outcome.
- The action must be refused because the system state changed during evaluation, even though it was valid at the start.
- That is why, time of checking and time of acting cannot be treated as the same because system state can change between these moments. An action that starts in a valid state may become unsafe or incorrect by the time it is executed. Treating these moments as identical can cause actions to succeed when they should fail, leading to inconsistent and unreliable behaviour.

DAY 03:

a. Valid Intent vs Executable

Scenario: User initiates two transfer from same source account
-Intent A: Transfer amount X to destination A
-Intent B: Transfer amount Y to destination B

Intent A:
1. Actor: User linked to a source account
2. Goal: Transfer amount X to destination A
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition:
Source account balance is sufficient for amount X
5. Result if conditions pass: Amount X is transferred.
6. Result if conditions fails: Balance unchanged.

Intent B:
1. Actor: User linked to a source account
2. Goal: Transfer amount Y to destination B
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition:
Source account balance is sufficient for amount Y
5. Result if conditions pass: Amount Y is transferred.
6. Result if conditions fails: Balance unchanged.

Conflict:
The source account balance is either sufficient for Intent A or Intent B,but not for both.

Outcome:
One Intent is executed, the other one gets refused

b. Composite Intent.

Scenario: A money transfer is treated as a composite intent consisting of multiple dependent steps:
-Deduct amount from source
- Add amount to destination.

Composite Intent:
Actor: User linked to a source account
Goal: Transfer a specific amount to a destination account
Required setup: Source and destination account and transfer mechanism should exist
Decision Condition:
-Source account balance is sufficient
-Both addition and deduction steps are executable

Fail case: Deduction succeed but addition fails

Outcome: The entire intent must be considered as failed, transfer is not successful

c. Irreversibility
Scenario: The user initiates the transfer that is completed , cannot be safely undone
- The transfer moves funds outside the immediate control of the system.
-The reversing of action cannot be guaranteed

1. Actor: User linked to a source account
2. Goal: Transfer a specific amount to a destination account
3. Required setup: Source and destination account and transfer mechanism should exist
4. Decision making condition:
The user has valid authority to initiate the transfer
Source account balance ≥ transfer amount at request time

Risk consideration:
Once executed the transfer cannot be reversed

Outcome: The action must be refused before execution.

d. Reflection
- Both intents are valid in isolation, but only one is executable. Validity of intent does not guarantee executability when resources conflict.
-A composite intent cannot be partially successful. Failure of any required step invalidates the whole intent.

-An action can be valid and still unsafe to execute. Irreversible consequences require refusal before execution, not correction after.
-Correctness alone is not sufficient to decide whether an action should be executed. An action may satisfy all conditions and still create irreversible or conflicting outcomes that the system cannot recover from. Systems must consider consequences, not just validity. Refusal is sometimes the only safe and correct response, even when an action appears valid.

DAY 04:

a. Human Misuse
- Humans may misuse the structure by taking shortcuts or relying on intuition instead of explicitly checking conditions.
-Assuming that a user who transferred money earlier is still allowed to do so, without rechecking authority.
-Skipping decision conditions because the transfer "usually works."
-Treating a sufficient balance check as permanent instead of time-dependent.
-Assuming that if one transfer succeeds, similar transfers will also succeed.

b. Machine Misuse
- Machines may misuse the structure through rigid or incomplete interpretation.
-Evaluating decision conditions only once and reusing the result for later transfers.
-Treating missing or delayed information as approval.
-Executing part of a transfer (such as deduction) without ensuring the full composite intent can complete.
-Applying conditions in an incorrect order, such as performing balance deduction before final validation.

c. Failure Classification
Must be prevented structurally:
-Executing a transfer without valid authority.
-Allowing a transfer when decision conditions are not explicitly satisfied.
-Treating identity as a substitute for authority.
-Allowing partial execution of a composite transfer.

Can only be handled procedurally:
- System interruptions during transfer processing.
-Temporary network or processing delays.
-Concurrent actions that change state during evaluation.

d. Grammar Freeze Reflection

The structure guarantees that actions are evaluated consistently by separating the actor, intent, setup, conditions, and outcomes. It ensures that authority, timing, and state are checked explicitly before execution and that refusal is a valid outcome.

The structure explicitly refuses to guarantee correctness beyond the defined conditions. It does not promise safety, reversibility, or fairness. It also does not protect against misuse outside the defined evaluation boundaries.

The structure must not be extended further to include assumptions about intent, correctness, or recovery. Adding such guarantees would introduce ambiguity and hidden logic, breaking determinism. Freezing the grammar preserves clarity by limiting what the structure is allowed to decide.