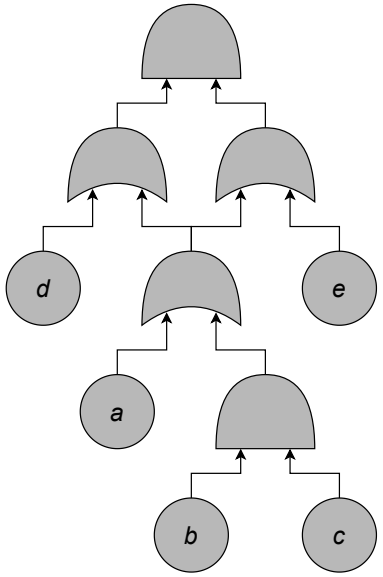
STAR Example Exam 2025

Exercise 1 (Risk management)

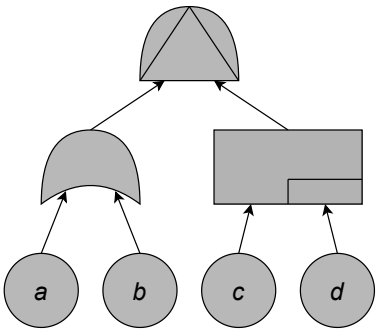
1. (2pt) Name a disadvantage of using "probability times impact" as a definition for risk.
2. (6pt) Suppose your objective is "get a passing grade for the STAR exam". Give three risks associated with this objective, and for each risk a strategy to handle that risk. State under which of the main risk handling strategies your strategy falls, and sketch the risks and their strategies on the risk matrix.

Exercise 2 (Static fault trees) Consider the fault tree below.



1. (2 point) List all of its minimal cut sets.
2. (2 point) Suppose $p_a = p_b = \frac{1}{2}$ and $p_c = p_d = p_e = \frac{1}{4}$. Approximate the failure probability using the cut set method; explain your answer.
3. (4 points) Give the binary decision diagram of this fault tree, with variable ordering $a < b < c < d < e$.
4. (3 points) Use the binary decision diagram to calculate the failure probability of the fault tree; explain your answer.

Exercise 3 (Dynamic fault trees) Consider the dynamic fault tree below. The BEs a, b, c have failure rates $\lambda_a, \lambda_b, \lambda_c$, respectively, while d has failure rate λ_d until it takes over the function of c, when it will have failure rate λ_d.



1. (4 points) First consider just the OR-gate. Given a time $t \geq 0$, find an expression for the probability that the OR-gate has failed, in terms of λ_a, λ_b, and t. Explain your answer.
2. (1 point) Show that the failure time of the OR-gate is exponentially distributed. What is its failure rate?
3. (6 points) Represent this DFT with a Markov chain with at most 6 states, and give its transition matrix.

Exercise 4 (Integration testing, old SET exam question) Suppose that you have a partial software implementation of the El Dorado game (for the rules of the game, see: `https://www.ultraboardgames.com/the-quest-for-el-dorado/game-rules.php`). You just finished your first implementation of the Hand cards of a Player, the Market Board, and that a Player can buy a card from the Market Board.
In Figure 1 some of the implementation's classes with the headers of some of its methods and constructors are shown in the Java language. Above each method a short description of its functionality is given in comments.
Now you want to write an integration test for buying a card with hand cards from the market board. It should be checked that the gold spend from the hand of the Player is the same as the cost of the card from the market board.

(a) To implement the integration test, (3pts)

   (i) which classes should to be replaced by stubs? Shortly motivate your answer.

   (ii) which classes should be replaced by drivers? Shortly motivate your answer.

   (iii) which classes should be used in the integration test as-is, i.e. with their actual implementation? Shortly motivate your answer.

(b) Write Java code for the integration test, using methods/constructors of Figure 1, and any methods/-constructors for stubs or drivers that you need to introduce. If you include a line of code with a call to a stub or driver method/constructor, you need to include a comment line above that describes what that line of code achieves. (6pts)

Exercise 5 (LTS)

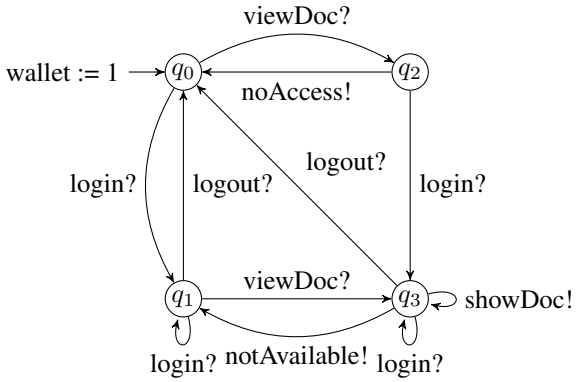
(a) Create an LTS that models an automatic sun screen that closes when sun is shining and opens when sun is not shining.

(b) Which states of your LTS are quiescent?

```
1  class MarketBooard {
2    // Constructor for initializing board with cards at start of game
3    MarketBoard()
4
5    // Iterate over all cards available for buying in the market
6    Iterator<Card> availableCardsIterator()
7
8    // Removes card from market
9    removeCard(Card card)
10 }
11
12 class HandCards {
13   // Constructor for initializing hand with random cards at start of game
14   HandCards()
15
16   // Constructor for initializing hand with t Traveler cards (each worth 1 gold)
17   // and random other cards at start of game
18   HandCards(int t)
19
20   // Removes gold cards and returns true if amount of gold is in hand
21   // returns false otherwise
22   boolean spendGold(int amount)
23 }
24
25 class Card {
26   //Creates a card with given name, power, color, and price
27   Card(String name, int power, Color color, int Price)
28
29   // Returns price of the card in gold
30   int getPrice();
31 }
32
33 class Player {
34   // Initializes player with hand
35   Player(HandCards hand)
36
37   // Returns the total amount of gold present in the hand
38   int goldInHand()
39
40   // Calls HandCards.spendGold MarketBoard.removeCard to buy card from market
41   // Returns true if hand has enough gold and false otherwise
42   boolean buy(Card card, MarketBoard market)
43 }
```

Figure 1: This figure shows some El Dorado classes with the headers of some of its methods and constructors. Above each method/constructor a short description of its functionality is given in comments.

Exercise 6 (ioco) Consider the following LTS as a specification:

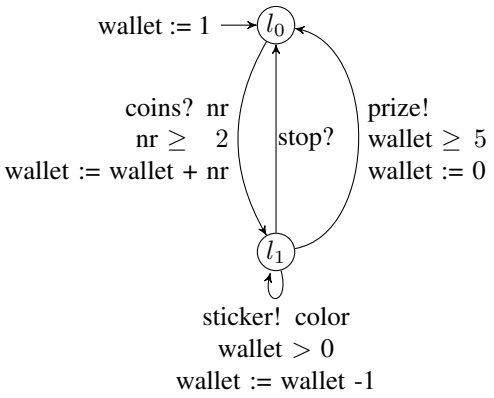


1. Create an implementation, that:
   - does not use one output label, that occurs on transition(s) of the specification, in any of its transition
   - is ioco with the specificaiton

and explain why it is ioco conforming

2. Create a test case of depth 3 (the tree has a trace with 3 subsequent labels), using batch test generation, such that the showDoc! output of the specification in state q_3 is covered by the test case.

Exercise 7 (STS) Consider the following graphical representation of an STS in this exercise:



(a) Write down the formal definition of the switches of the STS

(b) Write down the symbolic execution graph of the STS, up to level 3, i.e. take any valid sequence of three switches from the initial state of the graph.

(c) Write down a test trace such that the sticker switch is taken twice, and such that the prize! switch cannot be taken, according to the STS.

(d) Point out to which branch of the symbolic execution graph your trace corresponds.

(e) If you would choose values for test cases that test the first coins? switch, using boundary value analysis, what values would you then choose?