# Chapter 4 - Part I
# Test design

**Software Testing: IN3240 / IN4240**

# Summary:

**Test development process**

    Analysis / Design / Implementation

**Categories of test design techniques**

    Static / Dynamic

**Specification-based testing (black-box)**

    Equivalence partitioning / Boundary value analysis

    Decision table testing

    State transition testing

# Part I: Close-ended questions

# Question 1

In which **document** described in **IEEE 829** would you find **instructions** for the steps to be taken for a test including **set-up**, **logging**, **environment** and **measurement**?

a.  Test plan
b.  Test design specification
c.  Test case specification
d.  Test procedure specification

# Question 1: Clues

In which **document** described in **IEEE 829** would you find **instructions** for the steps to be taken for a test including **set-up, logging, environment** and **measurement**?

IEEE → Institute of Electrical and Electronics Engineers ("*I triple E*")

IEEE 829

Standard for Software and System Test Documentation

Specifies format of documents used in software / system testing

10 documents in total

# Question 1: Clues

In which **document** described in **IEEE 829** would you find **instructions** for the steps to be taken for a test including **set-up**, **logging**, **environment** and **measurement**?

Master Test Plan (MTP)

Level Test Plan (LTP)

Lest Test Design (LTD)

Level Test Case (LTC)

Level Test Procedure (LTPr)

Level Test Log (LTL)

Anomaly Report (AR)

Level Interim Test Status Report (LITSR)

Level Test Report (LTR)

Master Test Report (MTR)

# Question 1: Clues

In which **document** described in **IEEE 829** would you find **instructions** for the steps to be taken for a test including **set-up, logging, environment** and **measurement**?

| | |
|---|---|
| Master Test Plan (MTP) | Level Test Log (LTL) |
| Level Test Plan (LTP) | Anomaly Report (AR) |
| Lest Test Design (LTD) | Level Interim Test Status Report (LITSR) |
| Level Test Case (LTC) | Level Test Report (LTR) |
| Level Test Procedure (LTPr) | Master Test Report (MTR) |

# Question 1: Clues

In which **document** described in **IEEE 829** would you find **instructions** for the steps to be taken for a test including **set-up**, **logging**, **environment** and **measurement**?

Master Test Plan (MTP)

Scope, system overview, organisation

Responsibilities, tools, techniques, methods

Level Test Plan (LTP)

Like MTP but specific for each level of testing

Scope, resources, schedule of the testing activities

# Question 1: Clues

In which **document** described in **IEEE 829** would you find **instructions** for the steps to be taken for a test including **set-up**, **logging**, **environment** and **measurement**?

Level Test Design (LTD)

Detailing test cases → Identify features to be tested

Expected results

Level Test Case (LTC)

Objectives

Inputs / Outputs

# Question 1: Clues

In which **document** described in **IEEE 829** would you find **instructions** for the steps to be taken for a test including **set-up, logging, environment** and **measurement**?

Level Test Procedure (LTPr)

Detailed account of how to run each test

Description of each step to be taken to execute test cases

Set-up: Sequence of necessary actions to prepare for test execution

Log: List tools / methods for logging results

Environment: Describe environment for test execution

Measurement: Describe how test measurements will be made

# Question 1: Answer

In which **document** described in **IEEE 829** would you find **instructions** for the steps to be taken for a test including **set-up**, **logging**, **environment** and **measurement**?

a. Test plan
b. Test design specification
c. Test case specification
d. **Test procedure specification**

# Question 2

**With a highly experienced tester with a good business background, which approach to defining test procedures would be effective and most efficient for a project under severe time pressure?**

a.  A high-level outline of the test conditions and general steps to be taken
b.  Every step in the test spelled out in detail
c.  A high-level outline of the test conditions with the steps to take discussed in detailed with another experienced tester
d.  Detailed documentation of all test cases and careful records of each step taken in testing

# Question 2: Clues

**With a highly experienced tester with a good business background, which approach to defining test procedures would be effective and most efficient for a project under severe time pressure?**

Test effort under severe time pressure

Not feasible to define test procedures in full detail

Experience-based testing

Take advantage of the experience of the tester

Previous experience → Insights to what could go wrong

Possible solution

High-level outline of test condition + General steps to be taken

# Question 2: Answer

With a highly **experienced tester** with a good **business** background, which **approach** to **defining test** procedures would be **effective** and most efficient for a project under severe **time pressure**?

a. **A high-level outline of the test conditions and general steps to be taken**

b. Every step in the test spelled out in detail

c. A high-level outline of the test conditions with the steps to take discussed in detailed with another experienced tester

d. Detailed documentation of all test cases and careful records of each step taken in testing

# Question 3

Put the **test cases** that implement the following test conditions into the **best order** for the test **execution schedule**, for a test that is **checking modifications** of **customers** on a **database**.

1) Print modified customer record
2) Change customer address: House number and street name
3) Capture and print the on-screen error message
4) Change customer address: Postal code
5) Confirm existing customer is on the database by opening that record
6) Close the customer record and close the database
7) Try to add a new customer with no details at all

a. 5, 4, 2, 1, 3, 7, 6
b. 4, 2, 5, 1, 6, 7, 3
c. 5, 4, 2, 1, 7, 3, 6
d. 5, 1, 2, 3, 4, 7, 6

# Question 3: Clues

Put the **test cases** that implement the following test conditions into the **best order** for the test **execution schedule**, for a test that is **checking modifications** of **customers** on a **database.**

Activities

1. **Print** modified customer record

2. **Change** customer **address**: House number and street name

3. **Capture** and print the on-screen **error message**

4. **Change** customer **address**: Postal code

5. **Confirm** existing **customer** is in the database by **opening** that record

6. **Close** the customer **record** and close the **database**

7. **Try** to **add** a new **customer** with **no details** at all

# Question 3: Clues

Put the **test cases** that implement the following test conditions into the **best order** for the test **execution schedule**, for a test that is **checking modifications** of **customers** on a **database.**

Activities: Simplified

1.      **Print** modified record

2.      **Change address**

3.      **Capture error message**

4.      **Change address**

5.      **Confirm customer** by **opening** record

6.      **Close record** and close **database**

7.      **Try** to **add** new **customer** with **no details**

# Question 3: Clues

Put the **test cases** that implement the following test conditions into the **best order** for the test **execution schedule**, for a test that is **checking modifications** of **customers** on a **database**.

Execution schedule for checking modifications

What is the most intuitive order for customer record modification?

**Find** customer

**Modify** customer record

**Verify** modification

Create blank (provoke **error**)

**Verify** error

**Close** record + database

# Question 3: Clues

Put the **test cases** that implement the following test conditions into the **best order** for the test **execution schedule**, for a test that is **checking modifications** of **customers** on a **database**.

## Execution schedule for checking modifications

Find customer

Modify customer records

Verify modification

Create blank (provoke error)

Verify error

Close record + database

5. Confirm existing customer by opening record

4. Change address: Postal code

2. Change address: House number + Street

1. Print modified record

7. Try to add new customer, no details

3. Capture error message

6. Close record + database

# Question 3: Answer

Put the **test cases** that implement the following test conditions into the **best order** for the test **execution schedule**, for a test that is **checking modifications** of **customers** on a **database.**

1)      Print modified customer record
2)      Change customer address: House number and street name
3)      Capture and print the on-screen error message
4)      Change customer address: Postal code
5)      Confirm existing customer is on the database by opening that record
6)      Close the customer record and close the database
7)      Try to add a new customer with no details at all

a. 5, 4, 2, 1, 3, 7, 6
b. 4, 2, 5, 1, 6, 7, 3
**c. 5, 4, 2, 1, 7, 3, 6**
d. 5, 1, 2, 3, 4, 7, 6

# Question 4

**Why are both specification-based and structure-based testing techniques useful?**

a. They find different types of defects
b. Using more techniques is always better
c. Both find the same types of defect
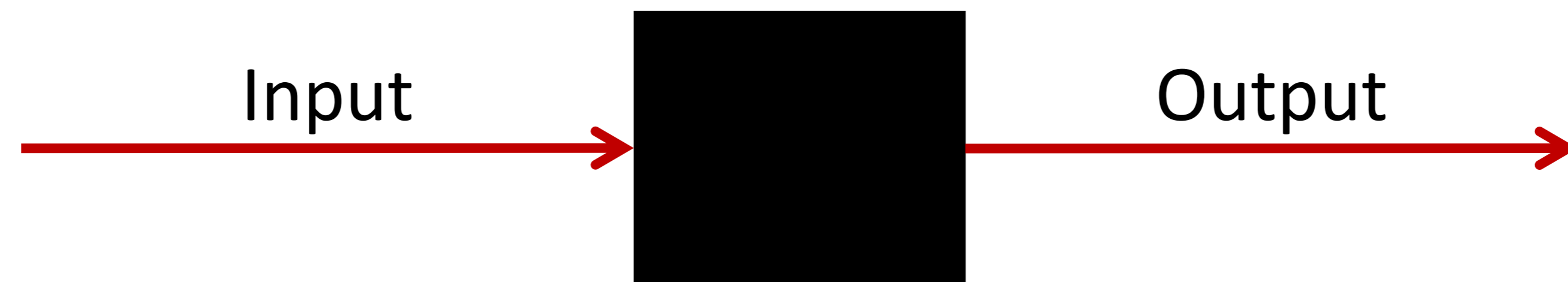d. Because specifications tend to be unstructured

**Why are both specification-based and structure-based testing techniques useful?**

Specification-based testing (Black-box testing)

Views software as a black box with inputs and outputs

Testers have *no knowledge* of how the system looks inside

Examines the functionality without looking into the internal structure

Input

Output

# Question 4: Clues
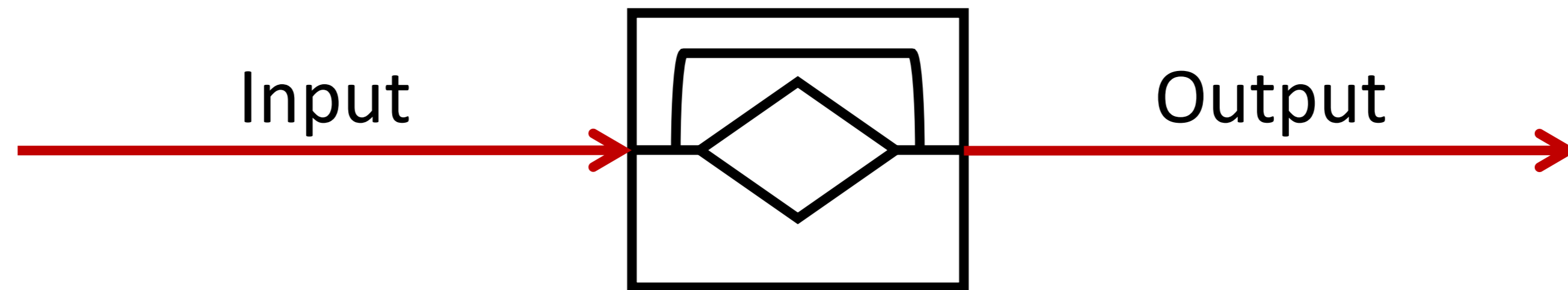
**Why are both specification-based and structure-based testing techniques useful?**

Structure-based testing (White-box testing)

Testers *require knowledge* of how the software is implemented

Testers ask the question: *How* does the software do it?

Examines the structure by looking into the program logic

Input              Output

# Question 4: Answer

Why are both **specification-based** and **structure-based** testing **techniques useful**?

a. **They find different types of defects**
b. Using more techniques is always better
c. Both find the same types of defect
d. Because specifications tend to be unstructured

# Question 5

**What is a key characteristic of structure-based testing techniques?**

a. They are mainly used to assess the structure of a specification
b. They are used both to measure coverage and to design tests to increase coverage
c. They are based on the skills and experience of the tester
d. They use a formal or informal model of the software or component

# Question 5: Clues

**What is a key characteristic of structure-based testing techniques?**

Overall objectives of testing

Find defects / Gain confidence in the system

Question: How?

Testing as much as possible / feasible

Concern

How to assess the thoroughness of the test effort

How much have we tested? How many aspects of the system have been checked?

# Question 5: Clues

**What is a key characteristic of structure-based testing techniques?**

Solution

Assess thoroughness of test effort through coverage

Approach: Structure-based techniques

Advantage: We have access to the code!

Examine code / internal structure of the software

Insight into logic / states / system architecture

# Question 5: Answer

What is a **key characteristic** of **structure-based** testing **techniques**?

a. They are mainly used to assess the structure of a specification
b. **They are used both to measure coverage and to design tests to increase coverage**
c. They are based on the skills and experience of the tester
d. They use a formal or informal model of the software or component

# Question 6

**Should pre-conditions and post-conditions be part of a test case?**

a. Yes
b. No

# Question 6: Clues

**Should pre-conditions and post-conditions be part of a test case?**

Test case (cf. IEEE 829)

Inputs

Execution conditions (pre- and post-conditions)

Expected / Predicted results

Developed for a particular objective

Exercise particular program / functionality

Verify compliance with specific requirement(s)

# Question 6: Clues

**Should pre-conditions and post-conditions be part of a test case?**

The need for test conditions

When can we start a test? / When does a test end?

What conclusions can we derive from a test? / What does the test tell us?

Pre-conditions

Condition(s) that must be in place PRIOR to running the test

Post-conditions

Condition(s) that must be in place AFTER running the test

# Question 6: Answer

Should **pre-conditions** and **post-conditions** be **part** of a **test case?**

a. **Yes**
b. No

_____ is the **analysis** at the **edge** of each **equivalence partition**.

We apply this test design technique because at the **edges** of **equivalence partitions**, the **results** are **more likely** to be **incorrect**.

# Question 7: Clues

_____ is the **analysis** at the **edge** of each **equivalence partition.**

Equivalence partitioning

Idea: Divide test conditions into groups that can be considered the same

These groups are equivalent

Test only one condition from each partition

Assume all conditions in the same partition will be treated the same

Little point in testing all values in the partition

Simplified assumptions → Not always right

**_____** is the **analysis** at the **edge** of each **equivalence partition.**

Example: Public transport ticket prices

| | |
|---|---|
| Children (under the age of 15): | 20 NOK for a single ticket |
| Students (between 15 and 25): | 25 NOK for a single ticket |
| Adults: | 35 NOK for a single ticket |
| Seniors (over the age of 65): | 20 NOK for a single ticket |

## Equivalence partitioning

E.g. We can assume that individuals of age 67, 68, 74, 88 are treated the *same*

Hence, when testing for senior discount → Do not have to test all ages

# Question 7: Clues

_____ is the **analysis** at the **edge** of each **equivalence partition.**

Example: Public transport ticket prices

Question: What is the right price for persons of age:

15 years / 25 years / 65 years?

Specifications may be unclear

Boundary value analysis (BVA)

Testing the boundaries (min. and max. values) / edges of equivalence partitions

High defect-finding capability

# Question 7: Answer

_____ is the **analysis** at the **edge** of each **equivalence partition**.

## BVA – Boundary Value Analysis

# Question 8

**Which of the following would be an example of decision-table testing for a financial application applied at system-test level?**

a. A table containing rules for combination of inputs to two fields on the screen
b. A table containing rules for interfaces between components
c. A table containing rules for mortgage applications
d. A table containing rules for chess

# Question 8: Clues

**Which of the following would be an example of decision-table testing for a financial application applied at system-test level?**

Decision-table testing

Cause-Effect table

Different combinations of input result in different actions

Aids in identifying effective test cases

Can reveal ambiguities in the specification

Explores business rules

| Conditions | R1 | R2 | R3 | R4 |
|---|---|---|---|---|
| Student enrolled in course | T | T | F | F |
| Mandatory exercises passed | T | F | T | F |
| **Actions** | | | | |
| Can write exam | T | F | F | F |

**Which of the following would be an example of decision-table testing for a financial application applied at system-test level?**

System testing

Concerned with the behaviour of the entire system

High-level descriptions of system behaviour

Often final testing phase on behalf of development

Hence:

We are interested in testing an overall / main aspect of the system

# Question 8: Answer

Which of the following would be an **example** of **decision-table** testing for a **financial** application applied at **system-test** level?

a. A table containing rules for combination of inputs to two fields on the screen
b. A table containing rules for interfaces between components
c. **A table containing rules for mortgage applications**
d. A table containing rules for chess

# Question 9

**Which of the following could be a coverage measure for state transition testing?**

| | |
|---|---|
| V. | All states have been reached |
| W. | The respond time for each transition is adequate |
| X. | Every transition has been executed |
| Y. | All boundaries have been exercised |
| Z. | Specific sequences of transitions have been exercised |

a. X, Y and Z

b. V, X, Y and Z

c. W, X and Y

d. V, X and Z

# Question 9: Clues

**Which of the following could be a coverage measure for state transition testing?**

Test coverage

Measure of the amount of testing performed by a set of tests

Simplified: How much of the code has been tested?

Aim: Reveal test coverage + Design additional tests to increase coverage

Coverage measure

How can we measure the coverage of the test effort?

What approaches / artefacts can be used to determine coverage?

# Question 9: Clues

**Which of the following could be a coverage measure for state transition testing?**

State-transition testing

Some aspect of the system can be described in a "finite state machine"

System can be in a *finite* number of different states

Transitions from one state to another depend on the *rules* of the machine

State diagram

Describes the behaviour of the system

Illustrates the different states a system can be in + Transitions between states

# Question 9: Clues

**Which of the following could be a coverage measure for state transition testing?**

Example: State diagram for PIN entry in ATM



Figure 4.2 in textbook

# Question 9: Clues

**Which of the following could be a coverage measure for state transition testing?**
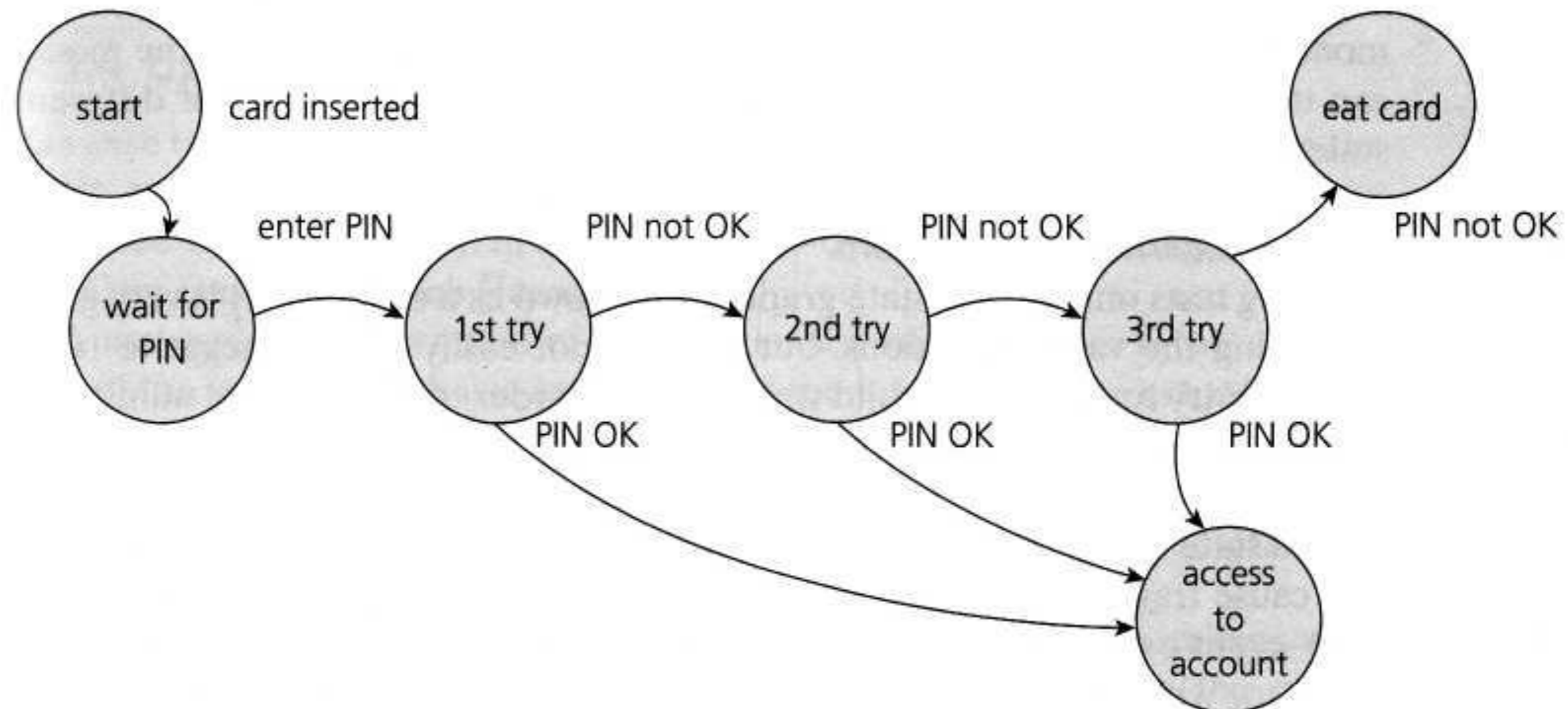
Example: State diagram for PIN entry in ATM
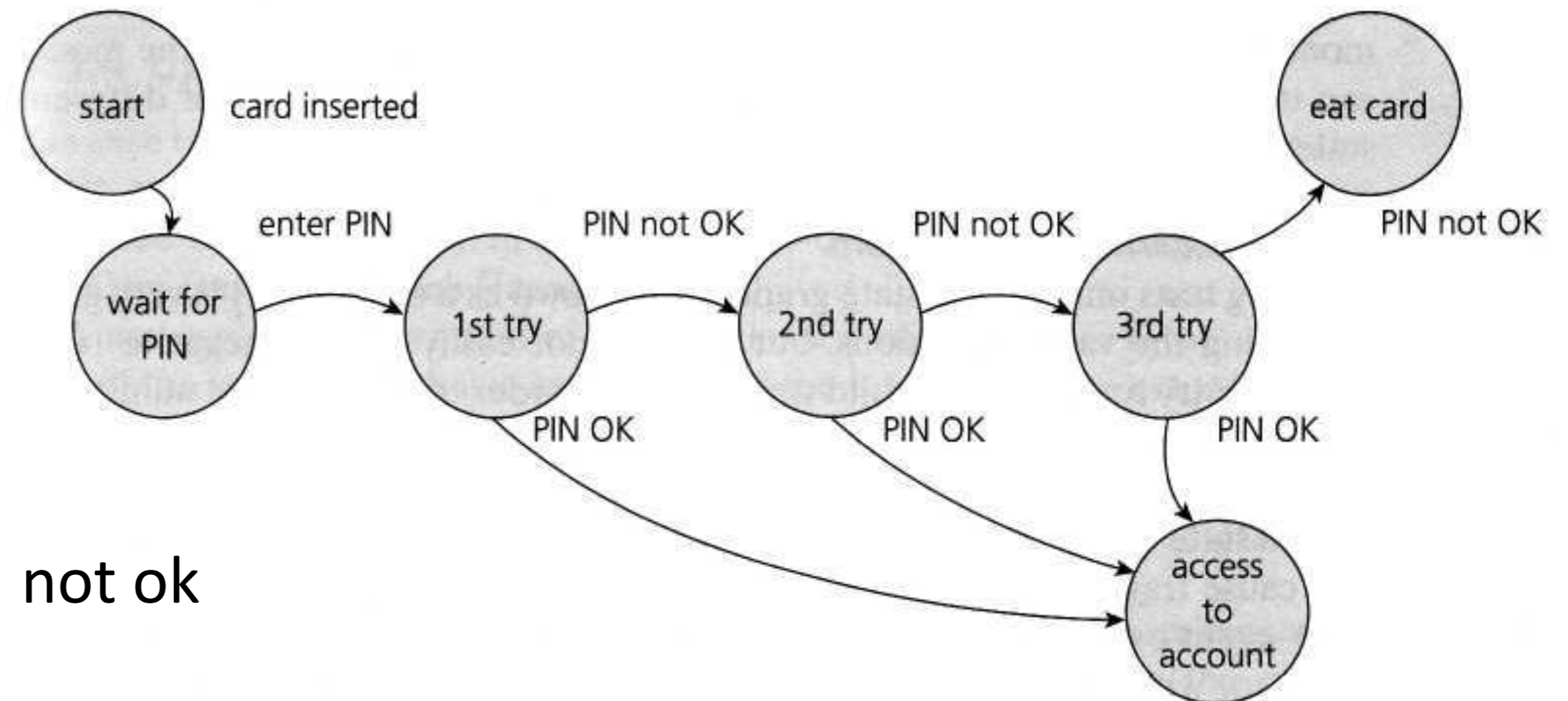
States software may be in

Shown in circles

Transitions from one state to another

Arrows pointing to the next transition

Events causing the transitions

Card inserted / Enter PIN / PIN ok / PIN not ok

Actions resulting from transitions

# Question 9: Clues

**Which of the following could be a coverage measure for state transition testing?**

State transition testing and coverage

When using state transition testing → What can measure coverage?

The number of states reached

Specific sequences of transitions exercised

Every transition has been executed

All of the above tell us about the amount of testing performed through state transition

However: What about testing all boundaries / boundary values?

# Question 9: Clues

**Which of the following could be a coverage measure for state transition testing?**

Testing boundary values

    Indeed a measure of testing coverage

    Tells us about the percentage of boundaries exercised

However: *Not* a coverage measure for state transition testing

    Testing boundary values does not necessarily tell us anything about state transitions

    Boundary values may only be relevant for certain states

    Example: We refer back to the ATM state diagram

# Question 9: Clues

**Which of the following could be a coverage measure for state transition testing?**

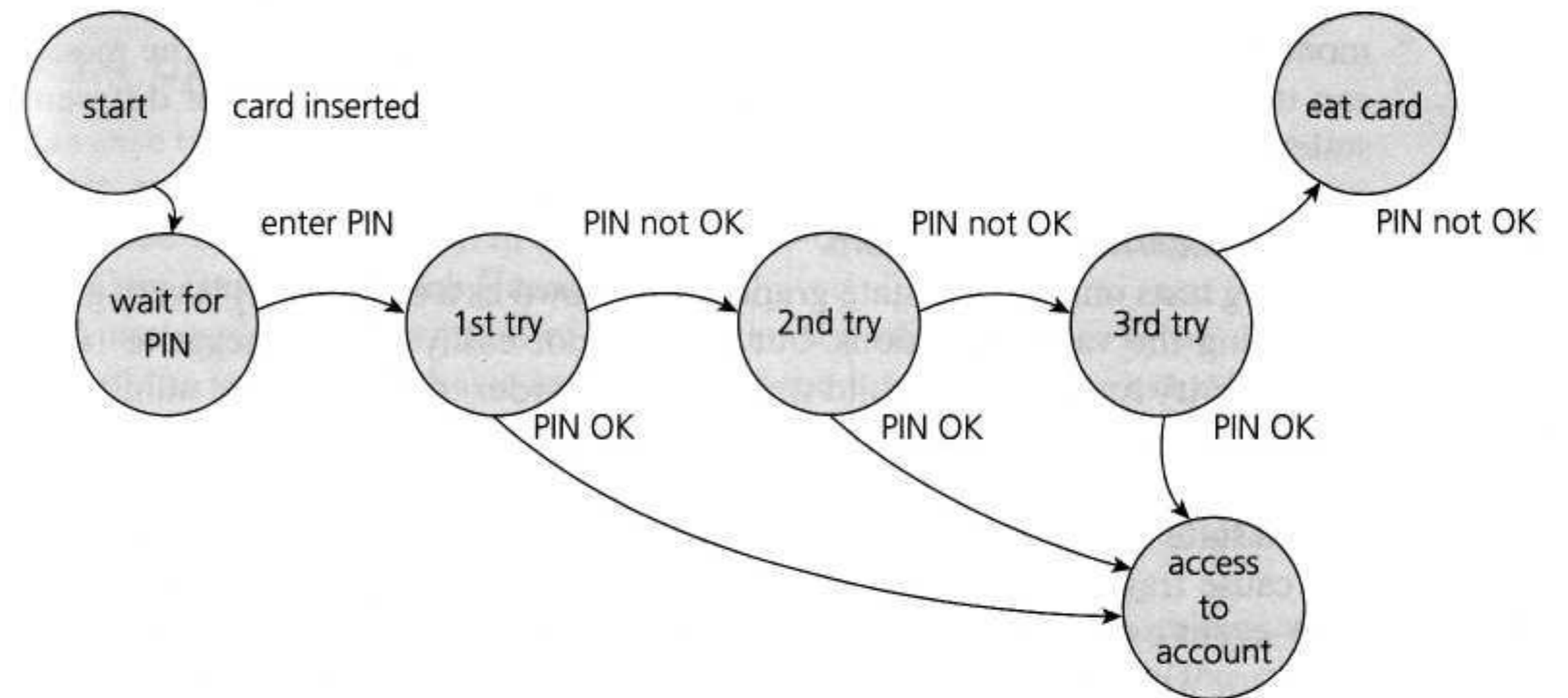Example: State diagram for PIN entry in ATM

Can test all PIN number boundaries

0000 – 9999

Less than four digits

More than four digits

Not directly linked to state transitions

# Question 9: Answer

Which of the following could be a **coverage measure** for **state transition** testing?

V.   All states have been reached
W.   The respond time for each transition is adequate
X.   Every transition has been executed
Y.   All boundaries have been exercised
Z.   Specific sequences of transitions have been exercised

a.   X, Y and Z
b.   V, X, Y and Z
c.   W, X and Y
d.   **V, X and Z**

# Question 10

**Which of the following could be used to assess the coverage achieved for specification-based test techniques?**

| | |
|---|---|
| V. | Decision outcomes exercised |
| W. | Partitions exercised |
| X. | Boundaries exercised |
| Y. | State transitions exercised |
| Z. | Statements exercised |

a. V, W, Y or Z

b. W, X or Y

c. V, X or Z

d. W, X, Y or Z

# Question 10: Clues

**Which of the following could be used to assess the coverage achieved for specification-based test techniques?**

Specification-based test techniques

Views software as a black box

No knowledge of how the system is internally structured

Concern: What the system does, not how it does it

Assessing coverage

Partitions exercised / Boundaries exercised / State transitions exercised

Decisions + Statements exercised → Internal structure
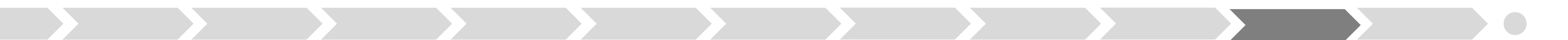
Structure-based techniques (white box)

# Question 10: Answer

Which of the following could be used to **assess** the **coverage** achieved for **specification-based** test **techniques**?

| | | |
|---|---|---|
| V. | Decision outcomes exercised |
| W. | Partitions exercised |
| X. | Boundaries exercised |
| Y. | State transitions exercised |
| Z. | Statements exercised |

a. V, W, Y or Z

**b. W, X or Y**

c. V, X or Z

d. W, X, Y or Z

# Part II: Exercises and Open-ended questions

# Exercise I: Equivalence Partitioning

Postal rates for 'light letters' are 25 NOK up to 10g, 35 NOK up to 50g, plus an extra 10 NOK for each additional 25g up to 100g. Which **test inputs** (in grams) would be **selected** using **equivalence partitioning**?
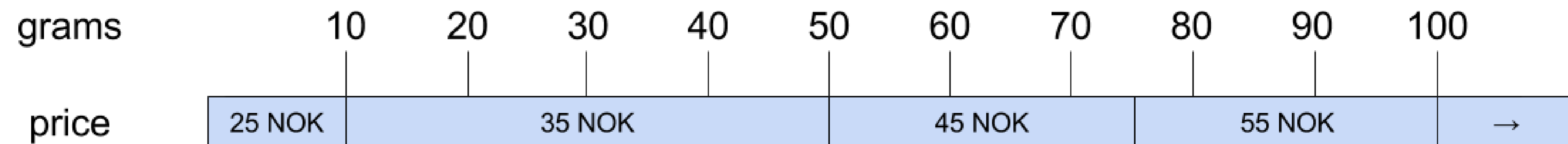
a.   8, 42, 82, 102
b.   4, 15, 65, 92, 159
c.   10, 50, 75, 100
d.   5, 20, 50, 60, 80

# Exercise I: Clues

**Which test inputs (in grams) would be selected using equivalence partitioning?**
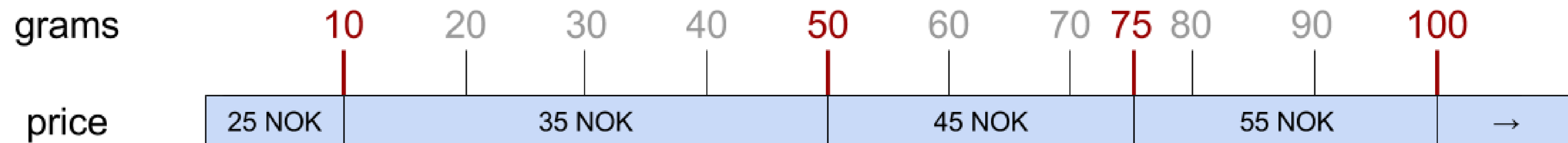
Scenario

How are postal rates calculated?

# Exercise I: Clues

**Which test inputs (in grams) would be selected using equivalence partitioning?**

Questions
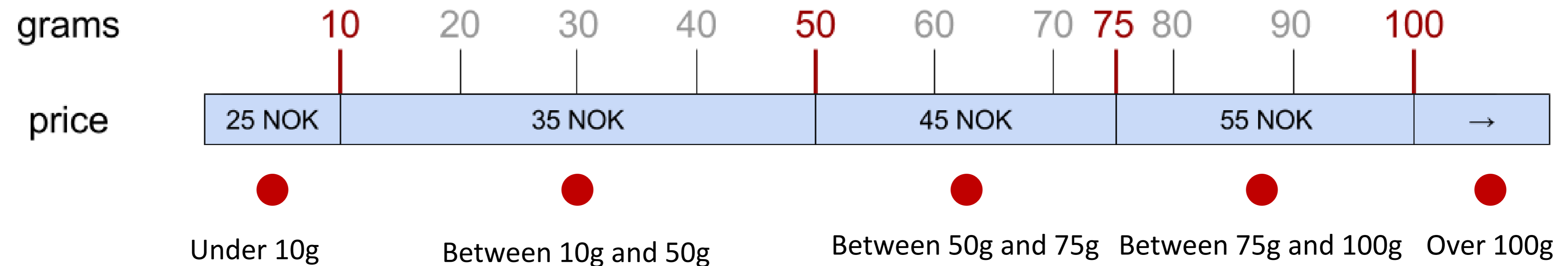
What are the key boundaries? / How many values do we need?

# Exercise I: Clues

**Which test inputs (in grams) would be selected using equivalence partitioning?**

Answer

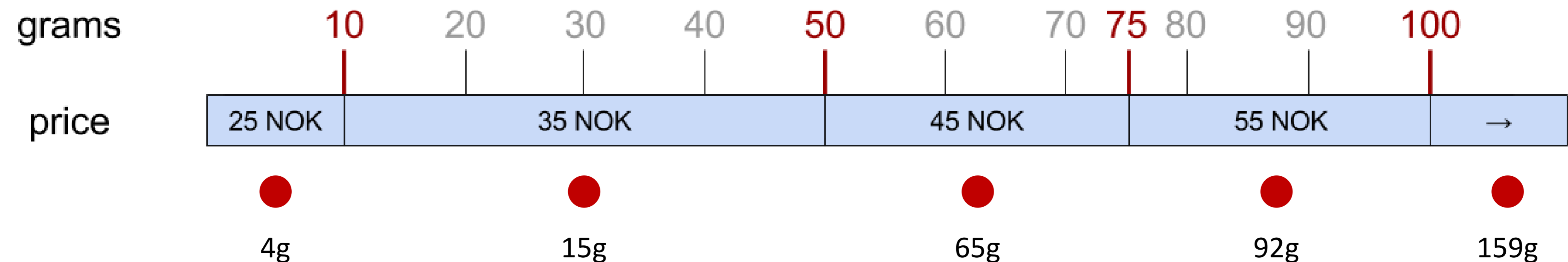We need five test inputs → Each in their own equivalence class

# Exercise I: Clues

**Which test inputs (in grams) would be selected using equivalence partitioning?**

Answer

We choose five arbitrary values for each equivalence class

# Exercise I: Answer

Postal rates for 'light letters' are 25 NOK up to 10g, 35 NOK up to 50g, plus an extra 10 NOK for each additional 25g up to 100g. Which **test inputs** (in grams) would be **selected** using **equivalence partitioning**?

a.   8, 42, 82, 102
b.   **4, 15, 65, 92, 159**
c.   10, 50, 75, 100
d.   5, 20, 50, 60, 80

# Exercise II

**If you take the train before 9:30 AM or in the afternoon after 4:00 PM until 7:30 PM ('rush hour') you must pay full fare. A saver ticket is available for trains between 9:30 AM and 4:00 PM, and after 9:30 PM.**

What are the partitions and boundary values to test the train times for this ticket types?

Which are valid partitions and which are invalid partitions?

What are the boundary values? (A table may be useful)

Derive test cases for the partitions and boundaries.

Do you have any questions about this 'requirement'?
Is anything unclear?

# Exercise II: Clues

Approach

Establish the exact boundaries between full fare and saver fare.

We can use a table to map out the information given:

Departure time of train

Corresponding ticket type for the departure time

Saver ticket

Full fare ticket

| Scheduled Departure time | | | | |
|---|---|---|---|---|
| Ticket type | | | | |

# Exercise II: Clues

Approach

"If you take the train before 9:30 am, or in the afternoon after 4:00 pm until 7:30 pm, you must pay full fare."

| Scheduled Departure time | ≦ 9:29 am | | 4:01 pm - 7:30 pm | |
|---|---|---|---|---|
| Ticket type | FULL | | FULL | |

"A saver ticket is available for trains between 9:30 am and 4:00 pm, and after 7:30 pm."

| Scheduled Departure time | | 9:30 am - 4:00 pm | | ≧ 7:31 pm |
|---|---|---|---|---|
| Ticket type | | SAVER | | SAVER |

# Exercise II: Clues

## Approach

This gives us the following table:

| Scheduled Departure time | ≤ 9:29 am | 9:30 am - 4:00 pm | 4:01 pm - 7:30 pm | ≥ 7:31 pm |
|---|---|---|---|---|
| Ticket type | FULL | SAVER | FULL | SAVER |

We assume that the boundary values are:

9:29 am, 9:30 am

4:00 pm, 4:01 pm

7:30 pm, 7:31 pm

## Benefit of this approach

Our exact interpretation of the specification can reveal ambiguities

# Exercise II: Clues

What we have so far:

Saver: Between 9:30 am and 4:00 pm

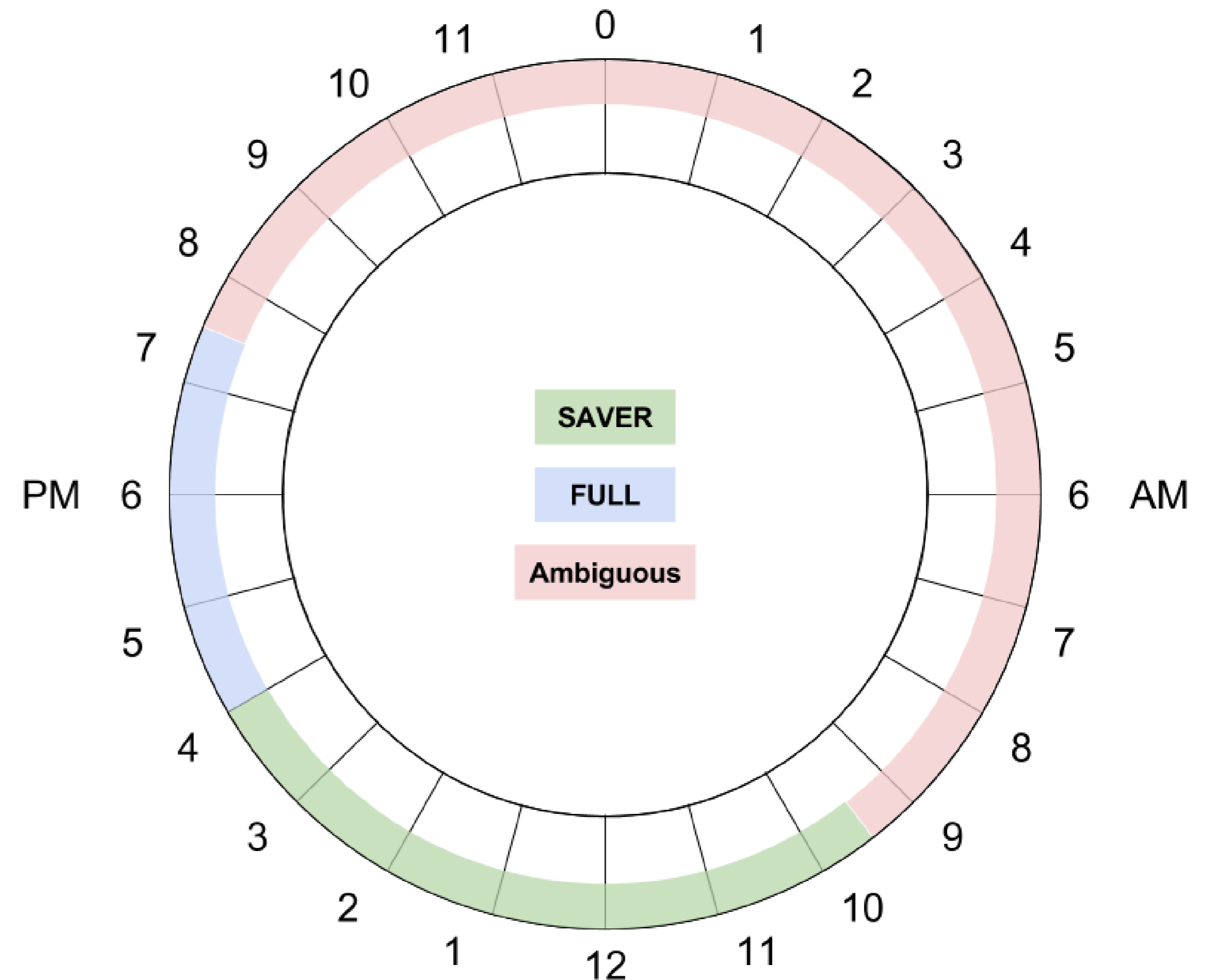Full: After 4:00 pm and until 7:30 pm

Ambiguities / Considerations

When does the morning "rush hour" start?

At midnight?

At 11:30 the previous day?

At the time of the first train of the day?

The specification is unclear!

# Exercise II: Clues

Other considerations

    If a train is scheduled to leave at exactly 4:00 pm

        Is a saver ticket still valid?

    If a train is scheduled to leave before 4:00 pm, but delayed until after 4:00 pm:

        Is a saver ticket still valid?


We can make assumptions, but we prefer the "correct" specification!

However, let us work with the information we have.

# Exercise II: Clues

Test cases for partitions and boundaries

| Test Case ID | Input | Expected outcome |
|---|---|---|
| 1 | Depart 4:30 am | Pay full fare |
| 2 | Depart 9:29 am | Pay full fare |
| 3 | Depart 9:30 am | Buy saver ticket |
| 4 | Depart 11:37 am | Buy saver ticket |
| 5 | Depart 4:00 pm | Buy saver ticket |
| 6 | Depart 4:01 pm | Pay full fare |
| 7 | Depart 5:55 pm | Pay full fare |
| 8 | Depart 7:30 pm | Pay full fare |
| 9 | Depart 7:31 pm | Buy saver ticket |
| 10 | Depart 10:05 pm | Buy saver ticket |

# Exercise III: Answer

Test cases for partitions and boundaries

Note: All partitions are valid

There may be invalid partitions

At a time no trains are running?

Specification does not mention that!

Equivalence partition values

Test cases: 1, 4, 7, and 10

Boundary values

Test cases: 2, 3, 5, 6, 8, and 9

| Test Case ID | Input | Expected outcome |
|:---:|:---:|:---:|
| 1 | Depart 4:30 am | Pay full fare |
| 2 | Depart 9:29 am | Pay full fare |
| 3 | Depart 9:30 am | Buy saver ticket |
| 4 | Depart 11:37 am | Buy saver ticket |
| 5 | Depart 4:00 pm | Buy saver ticket |
| 6 | Depart 4:01 pm | Pay full fare |
| 7 | Depart 5:55 pm | Pay full fare |
| 8 | Depart 7:30 pm | Pay full fare |
| 9 | Depart 7:31 pm | Buy saver ticket |
| 10 | Depart 10:05 pm | Buy saver ticket |

An informatics education program (study) at a university college in Norway have the following admission requirements:

- To obtain a study place, the applicant must have a general academic qualification, the mathematic course R1 from upper secondary school, and competition points above the limit of the year.

- If the applicant's competition points are below this limit, he/she is placed on the waiting list, assuming that the two first conditions are fulfilled.

- If the applicant has a general qualification, but not the R1 mathematics course, the applicant is offered a preparatory course in mathematics, assuming that the competition points are above the limit of that year.

**a)** Draw a decision table, which shows all the possible combinations of conditions for an applicant. The decision table shall include an action part, which shows whether or not the applicant is offered a study place, is placed on a waiting list, or is placed on a preparatory course in mathematics.

### ANSWER:

Conditions:

| | |
|---|---|
| GAQ: | General academic qualification |
| R1: | Passed the mathematic course R1 from upper secondary school |
| CP: | Competition points above this year's limit |

Actions:

| | |
|---|---|
| SP: | Offer of study place |
| WL: | Put on a waiting list |
| PR1: | Offer of preparation course in R1. |
| R: | Refusal |

| | Regel 1 | Regel 2 | Regel 3 | Regel 4 | Regel 5 | Regel 6 | Regel 7 | Regel 8 |
|---|---|---|---|---|---|---|---|---|
| GAQ | true | true | true | true | false | false | false | false |
| R1 | true | true | false | false | true | true | false | false |
| CP | true | false | true | false | true | false | true | false |
| Action | SP | WL | PR1 | R | R | R | R | R |

(It can be discussed whether you can calculate competition points when you do not have general study competence (rules 5 and 7). However, this problem disappears when the table is simplified in the next subtask.)

**b)** You shall now simplify the decision table and thus reduce the number of rules without losing any of the test cases. Justify the simplification.

Answer: We see from the table that when the applicant does not have general study qualifications, the person in question is rejected, regardless of whether the person in question has R1 or not, or whether the person in question has competition points above or below the year's intake limit. Consequently, rules 5, 6, 7 and 8 can be merged, and we get the following table:

|  | Regel 1 | Regel 2 | Regel 3 | Regel 4 | Regel 5 |
|---|---|---|---|---|---|
| GAQ | true | true | true | true | false |
| R1 | true | true | false | false | ---- |
| CP | true | false | true | false | ---- |

**c)** Consider the following three different test cases:

1. The applicant has general academic qualification, the mathematics course R1 from upper secondary school, and competition points above the limit of the year.

2. The applicant has general academic qualification, the mathematics course R1 from upper secondary school, and competition points below the limit of the year.

3. The applicant has general academic qualification and competition points above the limit of the year, but not the mathematics course R1 from upper secondary school.

Do we need to have more test cases? If yes, which should they be?

**Answer:**
Our goal is to have test cases that cover all the rules. We see from the table that the 3 test cases above cover rules 1, 2 and 3. We therefore need 2 more tests for rules 4 and 5:

1. The applicant has general academic qualification, but competition points below the limit of the year as well as not having the mathematical course R1 from upper secondary school.

2. The applicant do not have general academic qualification.

**Yulai Fjeld, revised by Eva H. Vihovde**

**Next week:**
Test design II