Course: Automata Theory

Lecture 2: Introduction to Sequences, Tuples, Functions and Relations

Lecturer: Martha Gichuki

Course description

- The course begins with an introduction to logic and formal grammar where learners will do a recap on sets, logic and truth tables, sequences, relations and functions
- A coverage of finite state machines, Push Down automata and Turing Machines (The Church’s thesis) will culminate the study of various models of computation.
- Formal language and grammar will then follow to enable learners differentiate regular and context free languages.
- An evaluation of the computability and complexity of practical computational problems which are the foundations of automata theory will then be done and the outcome will be problem description.

Learning outcomes:

Lecture 2: Introduction to Sequences, Tuples, Functions and Relations

At the end of the lecture, you will be able to:

i. Define a sequence, tuple, function and relation
ii. Explain the importance of these data structures to computing.
iii. Differentiate functions and relations as they are used in automata theory
iv. Solve problems involving sequences, functions and relations

2.1 Introduction to Sequences

Definition of a sequence:

- A sequence is **an ordered list of objects.**
- A list **of elements** arranged in some order either increasing order or decreasing order (ascending order or descending order).

Set versus Sequence

Unlike in sets:

- **Repetition is allowed** in sequences.
  - Example {1, 1, 2, 3, 3}
- **Order is also important** in sequences.
  - Example {1, 5, 7, 8}

2.2 Tuples

Definition of a tuple:

- A tuple is a sequence of elements.

Examples:

- A sequence of **k elements is a k-tuple. (Length property** has been added **(length = k)**).
- A **2-tuple** is a sequence of two elements (pair) example: {1, 3}
- A **3-tuple** is a sequence of three elements (triple) example: {1, 1, 2}
- A 4-tuple is a sequence of four elements (quadruple) example: {1, 2, 3, 3}

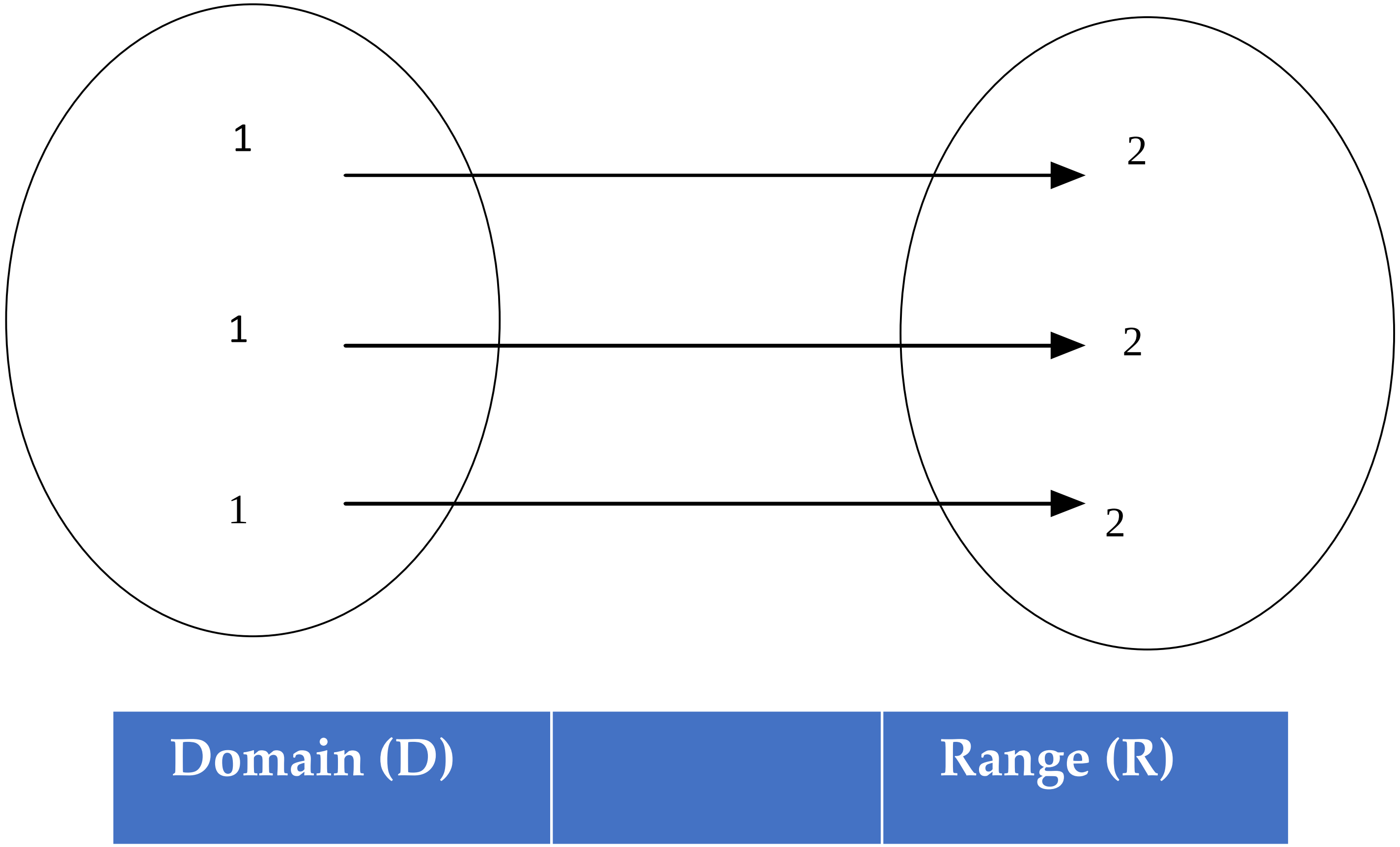
2.3 Functions

Definition of a function:

- A function is a **special kind of relationship** that maps one element from the input elements **(domain)** to only one other element of the output elements **(range) (with none included).**
- **Domain** – a collection of the input values
- **Range** – a collection of the output values

Functions….Example:

- Suppose domain **D** is the set of possible **inputs of the integer one (1)** and Range R is the set of possible **outputs of the integer two (2);**



Functions continued…

- A “***k-ary***” function is a function with **k arguments**.
- A ***unary*** function is a function with one element i.e. k=1
- A ***binary*** function is a function with two elements i.e. k=2
- A function can be a partial function, total function or onto function.

Note: One item in the domain cannot be mapped to more than one item in the range unless if it is a **relation**.

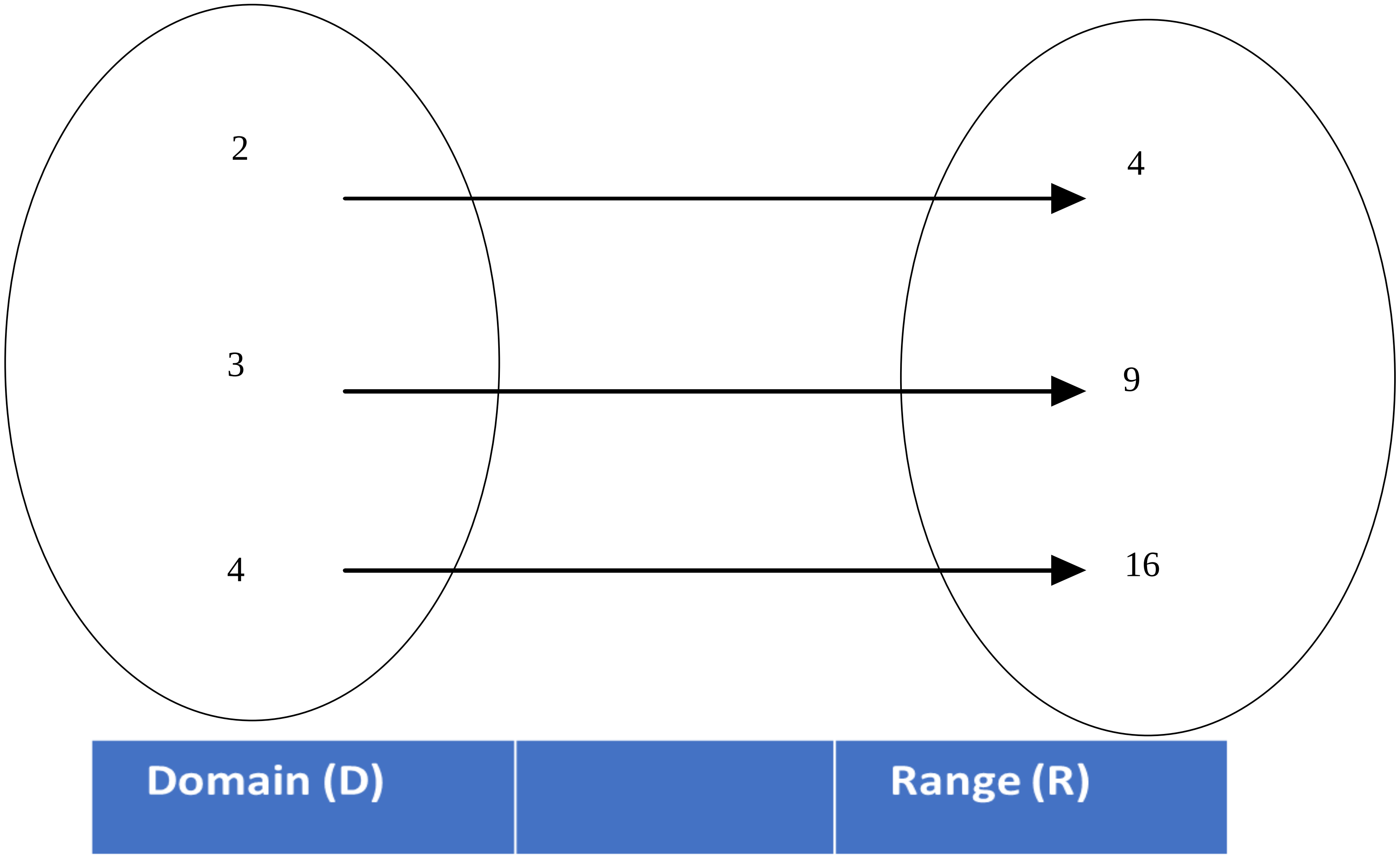
Functions…

- A function is a special kind of relation in which an element is mapped **onto only one other element at most**. (None included.
- This means that we cannot have an item in the range missing a mapping in the domain or an item in the domain missing a mapping from the range).

Example One of a Function Mapping:

- If a function *f* on *a* is described as: - f $(a) = b \wedge c$
- *Or* $f(a) = c \Rightarrow b = c$.
- A function square *f(sqr),* is a function that generates the square of a given number and can be demonstrated diagrammatically as follows: -

Functions Square mapping



2.4 Relations

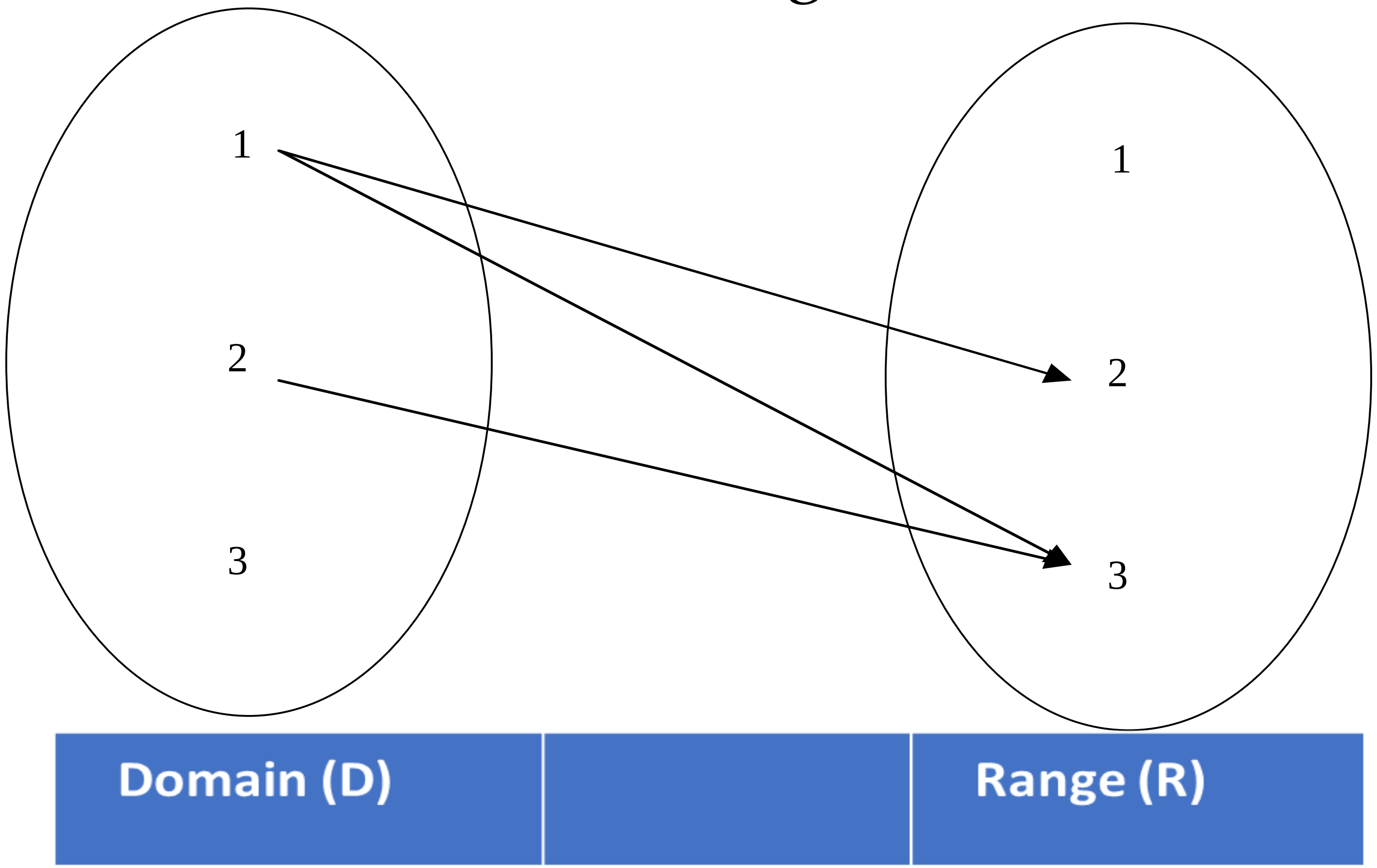
Definition of a relation:

- A relation describes the ***link between elements***.
- Unlike a function, a relation can map ***one*** domain element ***to many*** other elements in the range.
- A relation between two elements is said to be a ***binary relation*** and when a relation is binary, we use the infix notation aRb meaning *a relation to b*

- The infix notation may contain various connectives to link up elements
- Recall that connectives can also be the mathematical operators e,g,
    - < less than *relation (2,3)*
    - = equals *relation (2,2)*
    - < = less than or equal to *relation*
    - > greater than *relation (3,2)*
    - >= greater than or equal to *relation*
    - ≠ not equal to *relation (3,2)*

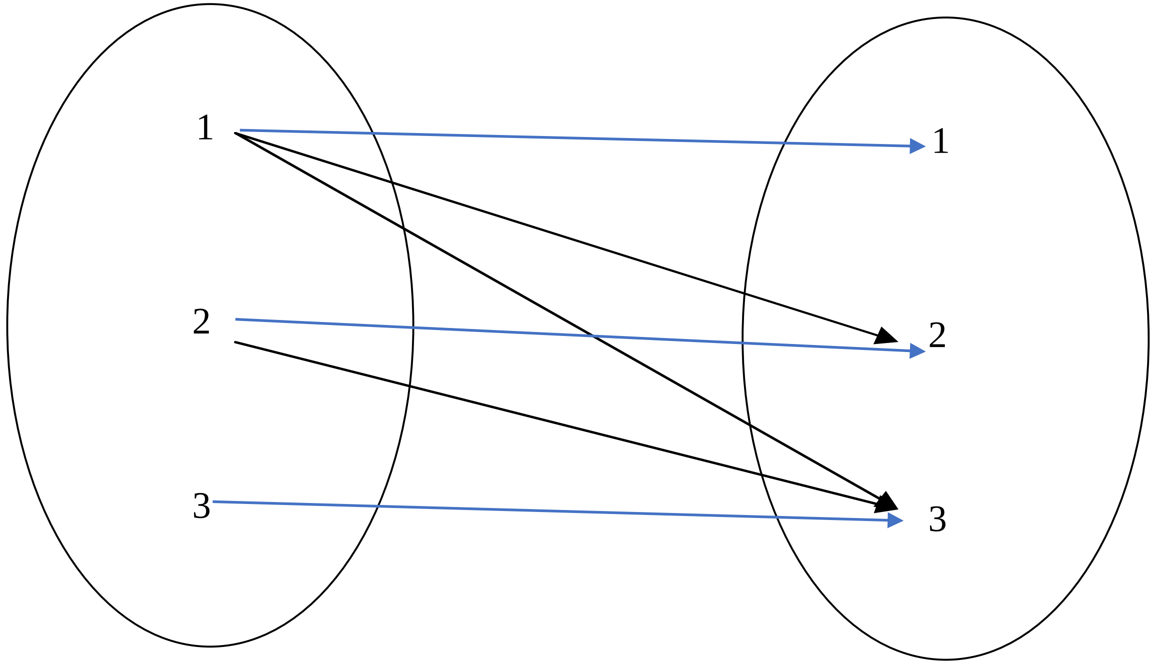
Example One:
The ***less than relation*** between element *a* and element *b* denoted as a<b means that a ***less than*** *b*.

Possible domain and range values include {(1, 2), (1, 3), (2, 3)}



Example 2

The ***less than or equal to relation*** between elements *a* and *b* denoted as a<=b means *element a* is ***less than or equal to*** element *b*. Possible domain and range values include {(1, 2), (1,1) (1, 3), (2,2), (2, 3), (3,3)}



Domain (D)		Range (R)

Equivalence Relation

- An *equivalence relation* describes two elements (objects) that are equal.
- There are four ways of looking at equivalence relations: -

  i. Reflexive Relation
  ii. Symmetric Relation,
  iii. Transitive Relation
  iv. Equivalence Relation

a) Reflexive Relation

Each element is related to itself or maps onto itself. Example, 1+1, 1=1…the *less than relation is not reflexive.*

- In terms of set theory, the relation R on a set A is denoted as: for every $x \in A$, $(x, x) \in R$.

$$\forall x \in A, (x, x) \in R.$$

b) Symmetric Relation

- ∀xy, xRy iff yRx - *For all x* and *y* elements, *x* is related to *y* ***if and only if*** *y* is related to *x*.
- An example of a symmetric relation is the less than relation that can only mean that the not equal relation holds e.g. ∀ 2, 3, 2R3 ∧ 3R2; 2<3∧ 3≠2;
- From set theory we realize that this is a relation R on a set A if (x, y) ∈ R then (y, x) ∈ R, for all x & y ∈ A.

c) Transitive Relation

- For all *x, y* and *z* elements, *x* is related to *y* ***and*** y is related to *z* meaning that *x* is related to *z*.
- ∀xyz, xRy yRz ⇒ xRz -
- The ***less than*** relation is transitive;

$$\forall 1, 2, 3,\ 1R2 \wedge 2R3 \Rightarrow 1R3$$

- If (x, y) ∈ R, (y, z) ∈ R, then (x, z) ∈ R,
- For all x, y, z ∈ A and this relation in set A is transitive.

d) Equivalence Relation

Equivalence – When all the above three relations are brought together, an equivalence relation is formed i.e. a saturation of the above three relations.

- A relation that is reflexive, symmetric and transitive is called an **equivalence relation**.

Review Questions

Question One

Using examples, describe the following terms clearly explaining their relevance to the design and operations of hardware and software

- Set
- Function
- Alphabet
- Language
- Relation

Question One - Solution

- **Set:** It is a well-defined collection of objects or elements e.g. set $A = \{a, e, i, o, u\}$ is the set of the English alphabets. A set is a non-repeating, unordered collection of objects (elements, members). E.g. the collection of four letters; a, b, c d is a set which is written as: L = {a, b, c, d}.
- A **function** is a special kind of relationship that maps one element from the input elements (domain) to only one other element of the output elements (range) (with none included).

Question One Solutions

- **Alphabet** – A finite set of symbols. It is frequently denoted by Σ , which is the set of letters in an alphabet.
- **Language** - refers to a set of words formed by symbols in each alphabet.
- A **relation** describes the link between elements. Unlike a function, a relation can map one domain element to many other elements in the range.

Question Two - If A = {0, 1, 2} and B = {a, b}, describe reflexive, transitive and symmetric relations over the sets A and B

Reflexive Relation – Each element is related to itself or maps onto itself.

Examples:

- 1+1, 1-1, 1=1, 1*1,
- 0+0, 0-0, 0=0,
- a*a a=a, a+a, a-a, a/a,
- 2+2, 2-2, 2=2, 2*2, 2-2,
- b+b, b*b, b-b, b=b, b/b,…

Question Two continued…

Symmetric Relation - ∀xy, xRy iff yRx - *For all x* and *y* elements, *x* is related to *y* ***if and only if*** *y* is related to *x*.

A = {0, 1, 2} and B = {a, b},

- $0<1$ $1>0$,
- $0<2$ $2>0$
- $1<2$ $2>1$
- $a<b$, $a>b$, $a \neq b$ etc.

Question Two continued…

Transitive Relation - $\forall xyz$, xRy $yRz \Rightarrow xRz$ - For all *x, y* and *z* elements, *x* is related to *y* ***and*** y is related to *z* meaning that *x* is related to *z*.

- The ***less than*** relation is transitive;

$$\forall 0, 1, 2, 0R1 \land 1R2 \Rightarrow 0R3$$

if $A = \{0, 1, 2\}$

$$\forall 0, 1, 2, 0<1 \land 1<2 \Rightarrow 0<3$$

Question Three - Using examples, describe two differences between a set and an ordered pair

- **A Set is not ordered but for the ordered pairs, order is important:**
- A set is a non-repeating, unordered collection of elements.

Examples:

- The collection of four letters; a, b, c d is a set which is written as: L = {a, b, c, d}.
- The collection of the square of the first three integers *(1,2,3) could form ordered pairs as follows:
  - {1,1}
  - {2,4}
  - {3,9}

References

1. Rowan G. & John T., (2009), *Discrete Mathematics: Proofs, Structures and Applications*, CRC Press, ISBN: 9781439812808.

2. W. D. Wallis (2003), *A Beginners Guide to Discrete Mathematics*, Springer Science & Business Media, ISBN: 978-0817642693.