Modeling and Simulation

Dr. Ilyes Naidji

Computer Science Department,
Mohamed Khider University of Biskra
Academic year: 2023/2024

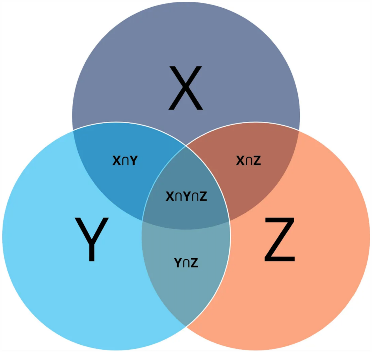
Overview of Sets

- A set is a collection of distinct elements.
- Example: $A = \{1, 2, 3\}$.
- Symbols: $\in$ (element of), $\subseteq$ (subset).
- Example: $2 \in A$ (2 is an element of set A).
- Set operations: $A \cup B$ (union), $A \cap B$ (intersection), A' (complement).

Elements and Subsets

- An element belongs to a set.
- Example: $3 \in A$ (3 is an element of set A).
- A subset contains only elements that are also in the larger set.
- Example: $B = \{1, 2\}$ is a subset of A.
- Power set of A: is the collection of all possible subsets that can be formed by taking elements from the original set. $P(A) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$.

Venn Diagram



- Let $A = \{1, 2, 3, 4\}$ and $B = \{3, 4, 5, 6\}$.
- Find $A \cup B$, $A \cap B$, and $A - B$.

Cardinality and Countability

- The cardinality of a set is the number of elements in the set. Example: $|A| = 3$ for set $A = \{1, 2, 3\}$.
- Countable sets can be put in one-to-one correspondence with the natural numbers. Example: The set of integers $\mathbb{Z}$ is countable.
- Additional Examples:

Relations and Functions

- A relation is a set of ordered pairs. Example: $R = \{(1,2),(2,3),(3,4)\}$.
- A function assigns each element from one set to exactly one element in another set. Example: $f : A \to B$, where $f(x) = x^2$.
- An equivalence relation satisfies reflexivity, symmetry, and transitivity. Example: $R = \{(a,a),(b,b),(c,c)\}$ is an equivalence relation on set $\{a,b,c\}$.
- A partition of a set divides it into disjoint subsets. Example: Partition $\{1,2,3,4,5\}$ into subsets with an even and odd sum.

Relations and Functions

- An equivalence relation is a fundamental concept in set theory and mathematics in general. It's a relation that satisfies three key properties: reflexivity, symmetry, and transitivity.
- 1. Reflexivity: A relation R on a set A is reflexive if every element of A is related to itself. In other words, for all x in A, the pair (x, x) is in R. Symbolically, xRx for all x in A.
- 2. Symmetry: A relation R on a set A is symmetric if whenever x is related to y, then y is related to x as well. In other words, if (x, y) is in R, then (y, x) must also be in R. Symbolically, if xRy then yRx.
- 3. Transitivity: A relation R on a set A is transitive if whenever x is related to y and y is related to z, then x is related to z as well. In other words, if (x, y) and (y, z) are in R, then (x, z) must also be in R. Symbolically, if xRy and yRz, then xRz.

Scenario Representation

Modeling Scenarios with Sets:

- Scenario: Students taking courses.
- Additional Examples: Represent a scenario where students enroll in multiple courses using set relations.

Scenario Representation (1)

Modeling Scenarios with Sets:

- Scenario: Students taking courses. Sets: $S = \{\text{students}\}$, $C = \{\text{courses}\}$. Relations: $R = \{(s, c) \mid s \text{ is taking } c\}$.
- Additional Examples: Represent a scenario where students enroll in multiple courses using set relations.

Scenario Representation (2)

Modeling Scenarios with Sets:

- Scenario: Event Management System
  Imagine you're developing an event management system for organizing conferences. The system needs to keep track of attendees, speakers, and sessions. Each attendee can register for multiple sessions, and each session can have multiple speakers.

Scenario Representation (3): Solution

- Entities:
  - Set of attendees:
  $\text{Attendees} = \{\text{Attendee}_1, \text{Attendee}_2, \text{Attendee}_3, \text{Attendee}_4\}$
  - Set of sessions: $\text{Sessions} = \{\text{Session}_A, \text{Session}_B, \text{Session}_C\}$ -
  Set of speakers: $\text{Speakers} = \{\text{Speaker}_1, \text{Speaker}_2, \text{Speaker}_3\}$
- Relations:
  - $R_{\text{Registration}}$: Represents the registration of attendees for sessions.
  - $R_{\text{Registration}} =$
  $\{(\text{Attendee}_1, \text{Session}_A), (\text{Attendee}_1, \text{Session}_B), (\text{Attendee}_2, \text{Session}_B),$
  - $R_{\text{SessionSpeakers}}$: Contains information about speakers for each session.
  - $R_{\text{SessionSpeakers}} =$
  $\{(\text{Session}_A, \text{Speaker}_1), (\text{Session}_A, \text{Speaker}_2), (\text{Session}_B, \text{Speaker}_2), (\text{Se}$

Scenario Representation (4): Solution

- Visualization:
  A matrix representation can be used to visualize the registration of attendees for sessions and the assignment of speakers to sessions.

	SessionA	SessionB	SessionC
Attendee1	x	x	
Attendee2		x	
Attendee3	x		x
Attendee4	x		

	SessionA	SessionB	SessionC
Speaker1	x		
Speaker2	x	x	
Speaker3		x	

Figure 1: Matrix Visualisation

Scenario Representation (5): Solution

In this scenario, set theory provides a structured way to represent the relationships between entities in the event management system. The relations $R_{\text{Registration}}$ and $R_{\text{SessionSpeakers}}$ capture the registration of attendees for sessions and the assignment of speakers to sessions, respectively. This representation facilitates efficient management and analysis of event data.

Set Theory in Data Structures

- Integration of Set Theory in Data Structures
- Data structure: Set implemented as an array or linked list.
- Example: Representing a set of characters using an array.
- Design a data structure to efficiently perform set union and intersection operations.

Basic Concepts of Logic

- Propositional Logic:
- $p \wedge q$ (p AND q), $p \vee q$ (p OR q), $\neg p$ (NOT p).
- Truth tables and logical equivalences.
- Example: $p \wedge (q \vee \neg p)$ with $p = \text{True}$, $q = \text{False}$.
- Additional Examples: Simplify the logical expression $(p \vee q) \wedge (\neg p \vee \neg q)$.

Basic Concepts of Logic

- Truth Table for the AND Operator ($\wedge$) The AND operator, denoted by $\wedge$, represents logical conjunction. Here's the truth table for the AND operator:

P	Q	$P \wedge Q$
True	True	True
True	False	False
False	True	False
False	False	False

- Example 2: Logical Equivalence - De Morgan's Laws De Morgan's Laws state the following logical equivalences: 1. $\neg(P \wedge Q) \equiv (\neg P) \vee (\neg Q)$ 2. $\neg(P \vee Q) \equiv (\neg P) \wedge (\neg Q)$
- Truth Table for the OR Operator ($\vee$) The OR operator, denoted by $\vee$, represents logical disjunction. Here's the truth table for the OR operator:

Propositional Calculus

- Syntax: Formal structure of logical expressions.
- Semantics: Assigning truth values to logical expressions.
- Tautology: A logical expression that is always true.
- Example: $p \vee \neg p$ is a tautology.
- Additional Examples: Prove the logical equivalence $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$.

Scenario Modeling with Propositional Calculus: Simple example

- Translating Scenarios into Logical Formulas: Scenario: Rain and John takes an umbrella. Logical expression:
- Conversion: Translating English scenarios into logical formulas.
- Additional Examples: Translate the scenario "If it is sunny, then Jane goes for a jog" into a logical expression.

Scenario Modeling with Propositional Calculus: Complex example

Scenario: Building Access Control Simulation
Description: Consider a simulation scenario where an automated building access control system is being designed. The system determines whether individuals are granted access to certain areas of a building based on various conditions.

- 1. Initial Condition: - Is the individual a registered employee? - Yes: Proceed to condition 2. - No: Access denied.
- 2. Time of Day: - Is it during normal working hours (9 AM - 5 PM)? - Yes: - Is the individual's access level set to "Employee" or higher? - Yes: Access granted. - No: Access denied. - No: - Is the individual's access level set to "Manager" or higher? - Yes: Access granted. - No: - Is it a weekend (Saturday or Sunday)? - Yes: Access denied. - No: Proceed to condition 3.

Scenario Modeling with Propositional Calculus: Complex example

- 3. Special Access: - Is the individual attending a scheduled meeting or event? - Yes: Access granted. - No: Access denied. This decision tree outlines the logical flow of decisions for granting access in the building access control simulation scenario. Each decision node represents a condition that needs to be evaluated, and the branches indicate the possible outcomes based on those conditions.

Logical Reasoning in Simulation

- Decision-making with Logic: Using logical reasoning in simulation scenarios. Example: Decision trees based on logical conditions.
- Additional Examples: Create a decision tree for a simulation scenario involving multiple logical conditions.

Decision-making with Logic

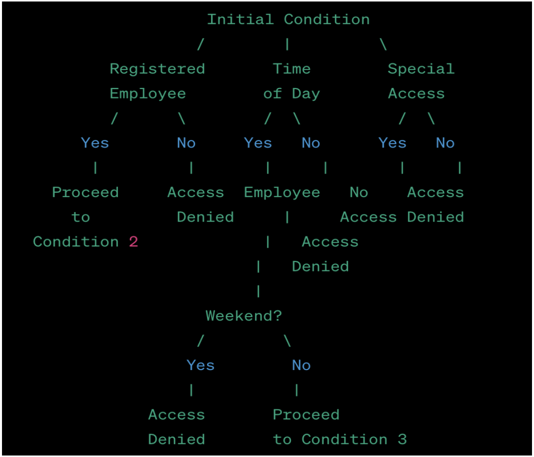


Figure 2: Decision Tree

Complex Example: Transportation Network Simulation

- Scenario: Consider a transportation network involving cities, vehicles, and routes. Vehicles travel between cities, and each route is associated with specific properties such as distance and travel time.
- Entities: Set of cities: $\text{Cities} = \{\text{City}_1, \text{City}_2, \text{City}_3, \text{City}_4\}$ Set of vehicles: $\text{Vehicles} = \{\text{Vehicle}_1, \text{Vehicle}_2, \text{Vehicle}_3\}$ Set of routes: $\text{Routes} = \{\text{Route}_A, \text{Route}_B, \text{Route}_C\}$
- Relations: R_{Travel}: Represents the travel of vehicles between cities.

$R_{\text{Travel}} = \{(\text{Vehicle}_1, \text{City}_1, \text{City}_2), (\text{Vehicle}_2, \text{City}_3, \text{City}_4), (\text{Vehicle}_3, \text{City}_2, \text{City}_3)\}$

$R_{\text{Properties}}$: Contains properties associated with each route.

$R_{\text{Properties}} = \{(\text{Route}_A, \text{Distance}, 150\,\text{km}), (\text{Route}_A, \text{Time}, 2\,\text{hours}), (\text{Route}_B, \text{Distance}, 200\,\text{km}), (\text{Route}_B, \text{Time}, 3\,\text{hours}), (\text{Route}_C, \text{Distance}, 120\,\text{km}), (\text{Route}_C, \text{Time}, 1.5\,\text{hours})\}$

Questions: Transportation Network Simulation

- Find all the cities connected by vehicles.
- Determine the total distance traveled by each vehicle.
- Identify the fastest route based on time.
- Add a new route, Route_D, connecting City_1 to City_3 with associated properties.
- Update the time property for Route_A to 2.5 hours.

Solution: Transportation Network Simulation

- 1- The cities connected by vehicles are $\text{City}_1, \text{City}_2, \text{City}_3, \text{City}_4$.
- 2- Total distance traveled by each vehicle: - Vehicle_1: 150 km (City1 to City2) - Vehicle_2: 200 km (City3 to City4) - Vehicle_3: 120 km (City2 to City3)
- 3- The fastest route based on time is Route_C with a travel time of 1.5 hours.
- Add a new route, Route_D, connecting City_1 to City_3 with associated properties.
- 4- Additions to R_{Travel}: $\{(\text{Vehicle}_4, \text{City}_1, \text{City}_3)\}$
  - Additions to $R_{\text{Properties}}$:
  $\{(\text{Route}_D, \text{Distance}, 180\,\text{km}), (\text{Route}_D, \text{Time}, 2.2\,\text{hours})\}$
- 5- Update to $R_{\text{Properties}}$: $\{(\text{Route}_A, \text{Time}, 2.5\,\text{hours})\}$

1. Set Theory
2. Linear Algebra
3. Calculus
4. Probability
5. Modeling and Simulation of Dynamic Systems