



Theory of Computation - Lecture 2

نظرية الحوسبة

DR. ANNIE IBRAHIM

Automata Theory & Computational Fundamentals

- ▶ نظرية الأوتوماتا والمفاهيم الحسابية الأساسية
- ▶ An introduction to formal languages, computation models, and logical foundations

Automata Theory

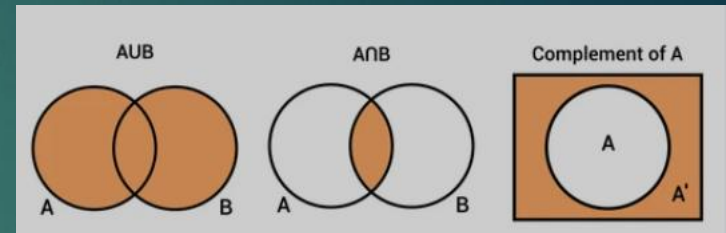
- ▶ What is Automata Theory?
- ▶ Study of abstract machines (automata) and computational problems they can solve
- ▶ Foundation of theoretical computer science and compiler design
- ▶ Classifies machines by their computational power (Finite Automata, Pushdown Automata, Turing Machines)
- ▶ نظرية الأوتوماتا
- ▶ والمشاكل الحسابية التي يمكنها حلها (أوتوماتا) دراسة الآلات المجردة
- ▶ أساس علوم الحاسوب النظرية وتصميم المترجمات
- ▶ تصنف الآلات حسب قدرتها الحسابية

Complexity Theory

- ▶ What is Complexity Theory?
- ▶ Studies the resources required to solve computational problems (time, space)
- ▶ Classifies problems into complexity classes (P, NP, NP-complete, EXP)
- ▶ Key question: P vs NP problem – are problems easy to verify also easy to solve?
- ▶ نظرية التعقيد
- ▶ (الزمن، المساحة) تدرس الموارد المطلوبة لحل المشكلات الحسابية
- ▶ تصنف المشكلات إلى فئات تعقيد
- ▶ هل المشكلات التي يمكن التحقق من حلها بسهولة يمكن حلها بسهولة أيضاً؟: السؤال الأساسي

Sets and Relations

- ▶ Sets
- ▶ Collection of distinct objects: $A = \{1, 2, 3\}$
- ▶ Operations: union ($\cup$), intersection ($\cap$), complement (A')



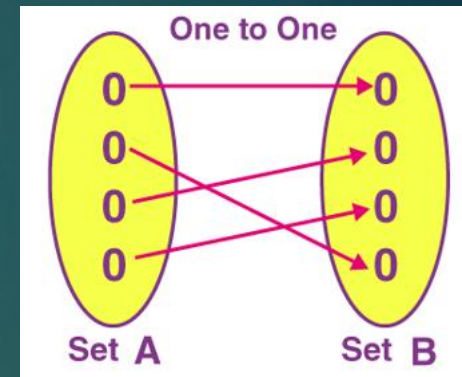
- ▶ Relations
- ▶ Subset of Cartesian product: $R \subseteq A \times B$
- ▶ Properties: reflexive, symmetric, transitive

- ▶ المجموعات والعلاقات
- ▶ مجموعة من العناصر المميزة: المجموعة
- ▶ مجموعة فرعية من الضرب الديكارتي: العلاقة

Property	Symbol
Reflexive	$a = a$
Symmetric	If $a = b$ then $b = a$
Transitive	If $a = b$ and $b = c$ then $a = c$

Functions and Relations

- ▶ Functions
- ▶ Special relation where each input maps to exactly one output
- ▶ $f: A \rightarrow B, \forall a \in A, \exists! b \in B$ such that $f(a) = b$
- ▶ Relations vs Functions
- ▶ Relations can have multiple outputs for same input
- ▶ Functions must have single, unique output



- ▶ الدوال والعلاقات
- ▶ علاقة خاصة حيث كل مدخل يرتبط بمخرج واحد فقط: الدالة
- ▶ العلاقات يمكن أن يكون لها عدة مخرجات لنفس المدخل: الفرق

Predicates

- ▶ What are Predicates?
- ▶ Statements whose truth depends on variables
- ▶ $P(x)$: " $x > 5$ " – true or false depending on x
- ▶ Basis of first-order logic and quantifiers ($\forall, \exists$)
- ▶ (المسندات) الكميات
- ▶ عبارات تحدد صحة أو خطأ بناءً على المتغيرات
- ▶ أساس المنطق من الرتبة الأولى والمحددات الكمية

Concatenation

- ▶ String Concatenation
 - ▶ Operation joining two strings end-to-end
 - ▶ "ab" + "cd" = "abcd"
 - ▶ Identity element: empty string ε
- ▶ Language Concatenation
 - ▶ $L_1L_2 = \{xy \mid x \in L_1, y \in L_2\}$
 - ▶ If $L_1 = \{a, ab\}$, $L_2 = \{b, bb\}$, then $L_1L_2 = \{ab, abb, abb, abbb\}$
- ▶ السلسلة والتوصيل
- ▶ عملية ربط سلسلتين نصيتين
- ▶ تطبيق على السلاسل واللغات الرسمية

Kleene Star

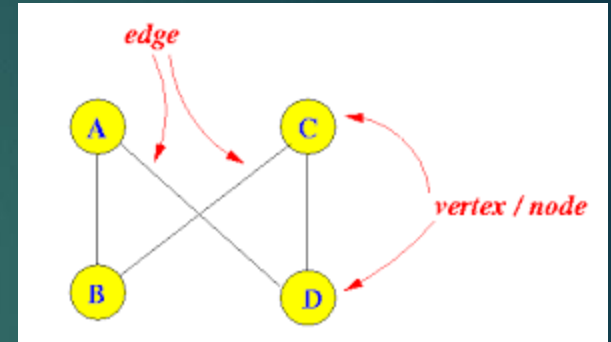
- ▶ Kleene Star Operation (*)
- ▶ $L^* = \{\epsilon\} \cup L \cup LL \cup LLL \cup \dots$ (all concatenations of zero or more strings from L)
- ▶ Example: $\{a\}^* = \{\epsilon, a, aa, aaa, \dots\}$
- ▶ Kleene Plus (+): $L^+ = LL^*$ (one or more concatenations)
- ▶ Example: $\{a\}^+$: Generates $\{a, aa, aaa, \dots\}$.
- ▶ عملية كليني النجمية
- ▶ جميع السلاسل الممكنة بتكرار عناصر اللغة صفر أو أكثر من المرات
- ▶ أساسية في التعابير النمطية واللغات المنتظمة

Regular Expressions

- ▶ What are Regular Expressions?
- ▶ Notation for describing regular languages
- ▶ Built from: symbols (a, b), union (+ or |), concatenation, Kleene star (*)
- ▶ Example: $(a+b)^*abb$ describes all strings ending with "abb"
- ▶ التعابير النمطية
- ▶ تدوين لوصف اللغات المنتظمة
- ▶ تستخدم في البحث عن النصوص ومعالجة اللغات
- ▶ "abb" يصف جميع السلاسل المنتهية بـ $(a+b)^*abb$: مثال

Graphs

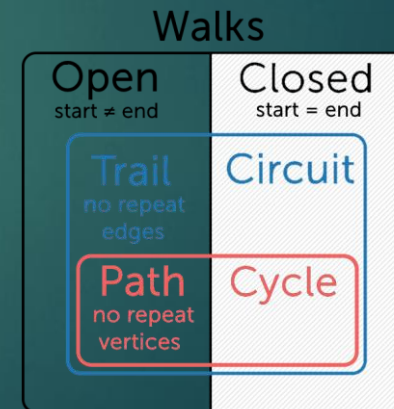
- ▶ Graph Theory Basics
- ▶ $G = (V, E)$ where V = vertices, E = edges
- ▶ Types: directed/undirected, weighted/unweighted
- ▶ Applications: networks, social graphs, routing



- ▶ Graph Terminology
- ▶ Path: sequence of connected vertices
- ▶ Cycle: path that starts and ends at same vertex

- ▶ Tree: acyclic connected graph

- ▶ (الغرافات) الرسوم البيانية
- ▶ هيكل بيانات يتكون من عقد وروابط
- ▶ غير مرجح/غير موجه، مرجح/موجه: أنواع
- ▶ الشبكات، الخوارزميات، تحليل البيانات: تطبيقات



Boolean Logic

- ▶ Boolean Algebra
- ▶ Values: TRUE (1), FALSE (0)
- ▶ Basic operations: AND ($\wedge$), OR ($\vee$), NOT ($\neg$)
- ▶ Laws: De Morgan's, distributive, identity



- ▶ Logic Gates & Circuits
- ▶ Building blocks of digital circuits
- ▶ AND gate, OR gate, NOT gate, XOR gate
- ▶ Foundation of computer hardware

- ▶ المنطق البولياني
- ▶ (خطأ/صحيح) جبر القيم المنطقية
- ▶ العمليات الأساسية: AND, OR, NOT
- ▶ أساس الدوائر الرقمية ومعالجة البيانات

Connections Between Topics

► **How These Concepts Relate**

- Automata recognize languages described by Regular Expressions
- Boolean logic implements computation in automata
- Graphs model state transitions in automata
- Complexity theory classifies automata capabilities
- Sets/relations are the mathematical foundation

- الروابط بين المواضيع
- الأوتوماتا تتعرف على اللغات الموصوفة بالتعابير النمطية
- المنطق البولياني ينفذ الحساب في الأوتوماتا
- نظرية التعقيد تصنف قدرات الآلات الحسابية

Applications

- ▶ Real-World Applications
- ▶ Regular Expressions: Text search, validation, NLP
- ▶ Automata: Compiler design, protocol verification
- ▶ Boolean Logic: Circuit design, database queries
- ▶ Graphs: Social networks, GPS navigation, web crawling
- ▶ Complexity: Algorithm design, cryptography
- ▶ التطبيقات العملية
- ▶ تصميم المترجمات، البحث في النصوص، تحليل الشبكات، تصميم: تستخدم هذه المفاهيم في الخوارزميات