We will cover these parts of the book (8th edition):

8.1.1, 8.1.2
8.2.1, 8.2.2(up to page 544)
8.3
9.1
9.2.1-9.2.4
9.3

Now it's Time for…

Advanced

Counting

Recurrence Relations

- A **recurrence relation** for the sequence $\{a_n\}$ is an equation that expresses a_n is terms of one or more of the previous terms of the sequence, namely, a_0, $a_1, \ldots, a_{n-1}$, for all integers n with $n \geq n_0$, where n_0 is a nonnegative integer.

- A sequence is called a **solution** of a recurrence relation if its terms satisfy the recurrence relation.

- In this section we will show that such relations can be used to study and to solve counting problems.

UMASS BOSTON

Recurrence Relations

- In other words, a recurrence relation is like a recursively defined sequence, but **without specifying any initial values (initial conditions)**.

- Therefore, the same recurrence relation can have (and usually has) **multiple solutions**.

- If **both** the initial conditions and the recurrence relation are specified, then the sequence is **uniquely** determined.

UMASS BOSTON

Recurrence Relations

- **Example:**

Consider the recurrence relation
$a_n = 2a_{n-1} - a_{n-2}$ for $n = 2, 3, 4, \ldots$

- Is the sequence $\{a_n\}$ with $a_n=3n$ a solution of this recurrence relation?
- For $n \geq 2$ we see that

$2a_{n-1} - a_{n-2} = 2(3(n-1)) - 3(n-2) = 3n = a_n$.

- Therefore, $\{a_n\}$ with $a_n=3n$ is a solution of the recurrence relation.

Recurrence Relations

- Is the sequence $\{a_n\}$ with $a_n=5$ a solution of the same recurrence relation?
- For $n \geq 2$ we see that
$2a_{n-1} - a_{n-2} = 2 \cdot 5 - 5 = 5 = a_n$.
- Therefore, $\{a_n\}$ with $a_n=5$ is also a solution of the recurrence relation.

Modeling with Recurrence Relations

- **Example:**
- Someone deposits $10,000 in a savings account at a bank yielding 5% per year with interest compounded annually. How much money will be in the account after 30 years?

- **Solution:**
- Let P_n denote the amount in the account after n years.
- How can we determine P_n on the basis of P_{n-1}?

UMASS BOSTON

Modeling with Recurrence Relations

- We can derive the following **recurrence relation**:
- $P_n = P_{n-1} + 0.05P_{n-1} = 1.05P_{n-1}$.
- The initial condition is $P_0 = 10{,}000$.
- Then we have:
- $P_1 = 1.05P_0$
- $P_2 = 1.05P_1 = (1.05)^2P_0$
- $P_3 = 1.05P_2 = (1.05)^3P_0$
- …
- $P_n = 1.05P_{n-1} = (1.05)^nP_0$

- We now have a **formula** to calculate P_n for any natural number n and can avoid the iteration.

Modeling with Recurrence Relations

- Let us use this formula to find P_{30} under the
- initial condition $P_0 = 10{,}000$:
- $P_{30} = (1.05)^{30} \cdot 10{,}000 = 43{,}219.42$
- After 30 years, the account contains \$43,219.42.

Modeling with Recurrence Relations

- **Another example:**
- Let a_n denote the number of bit strings of length n that do not have two consecutive 0s ("valid strings"). Find a recurrence relation and give initial conditions for the sequence $\{a_n\}$.

- **Solution:**
- Idea: The number of valid strings equals the number of valid strings ending with a 0 plus the number of valid strings ending with a 1.

Modeling with Recurrence Relations

- Let us assume that $n \geq 3$, so that the string contains at least 3 bits.
- Let us further assume that we know the number a_{n-1} of valid strings of length $(n – 1)$ and the number a_{n-2} of valid strings of length $(n – 2)$.
- Then how many valid strings of length n are there, if the string ends with a 1?
- There are a_{n-1} such strings, namely the set of valid strings of length $(n – 1)$ with a 1 appended to them.
- **Note:** Whenever we append a 1 to a valid string, that string remains valid.

Modeling with Recurrence Relations

- Now we need to know: How many valid strings of length n are there, if the string ends with a **0**?
- Valid strings of length n ending with a 0 **must have a 1 as their (n – 1)st bit** (otherwise they would end with 00 and would not be valid).
- And what is the number of valid strings of length (n – 1) that end with a 1?
- We already know that there are a_{n-1} strings of length n that end with a 1.
- Therefore, there are a_{n-2} strings of length (n – 1) that end with a 1.

UMASS BOSTON

Modeling with Recurrence Relations

- So there are a_{n-2} valid strings of length n that end with a 0 (all valid strings of length (n – 2) with 10 appended to them).

- As we said before, the number of valid strings is the number of valid strings ending with a 0 plus the number of valid strings ending with a 1.

- That gives us the following **recurrence relation**:
- $a_n = a_{n-1} + a_{n-2}$

Modeling with Recurrence Relations

- What are the **initial conditions**?

- $a_1 = 2$ (0 and 1)
- $a_2 = 3$ (01, 10, and 11)
- $a_3 = a_2 + a_1 = 3 + 2 = 5$
- $a_4 = a_3 + a_2 = 5 + 3 = 8$
- $a_5 = a_4 + a_3 = 8 + 5 = 13$
- …

- This sequence satisfies the same recurrence relation as the **Fibonacci sequence**.
- Since $a_1 = f_3$ and $a_2 = f_4$, we have $a_n = f_{n+2}$.

Solving Recurrence Relations

- In general, we would prefer to have an **explicit formula** to compute the value of a_n rather than conducting n iterations.

- For one class of recurrence relations, we can obtain such formulas in a systematic way.

- Those are the recurrence relations that express the terms of a sequence as **linear combinations** of previous terms.

Solving Recurrence Relations

- **Definition:** A linear homogeneous recurrence relation of degree k with constant coefficients is a recurrence relation of the form:
- $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}$,
- Where $c_1, c_2, \ldots, c_k$ are real numbers, and $c_k \neq 0$.
- A sequence satisfying such a recurrence relation is uniquely determined by the recurrence relation and the k initial conditions
- $a_0 = C_0, a_1 = C_1, a_2 = C_2, \ldots, a_{k-1} = C_{k-1}$.

Solving Recurrence Relations

- **Examples:**
- The recurrence relation $P_n = (1.05)P_{n-1}$
- is a linear homogeneous recurrence relation of **degree one**.
- The recurrence relation $f_n = f_{n-1} + f_{n-2}$
- is a linear homogeneous recurrence relation of **degree two**.
- The recurrence relation $a_n = a_{n-5}$
- is a linear homogeneous recurrence relation of **degree five**.

Solving Recurrence Relations

- Basically, when solving such recurrence relations, we try to find solutions of the form $a_n = r^n$, where r is a constant.
- $a_n = r^n$ is a solution of the recurrence relation $a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k}$ if and only if
- $r^n = c_1r^{n-1} + c_2r^{n-2} + \ldots + c_kr^{n-k}$.
- Divide this equation by r^{n-k} and subtract the right-hand side from the left:
- $r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_{k-1}r - c_k = 0$
- This is called the **characteristic equation** of the recurrence relation.

Solving Recurrence Relations

- The solutions of this equation are called the **characteristic roots** of the recurrence relation.

- Let us consider linear homogeneous recurrence relations of **degree two**.

- **Theorem:** Let c_1 and c_2 be real numbers. Suppose that $r^2 - c_1 r - c_2 = 0$ has two distinct roots r_1 and r_2.
- Then the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ if and only if $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ for $n = 0, 1, 2, \ldots$, where α_1 and α_2 are constants.

- The proof is shown on page 542/543 (8th Edition), 515/516 (7th Edition)

Solving Recurrence Relations

- **Example:** What is the solution of the recurrence relation $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$ and $a_1 = 7$?

- **Solution:** The characteristic equation of the recurrence relation is $r^2 - r - 2 = 0$. (because C_1=1, C_2=2)
- Its roots are $r = 2$ and $r = -1$.
- Hence, the sequence $\{a_n\}$ is a solution to the recurrence relation if and only if:
- $a_n = \alpha_1 2^n + \alpha_2(-1)^n$ for some constants α_1 and α_2.

UMASS BOSTON

Solving Recurrence Relations

- Given the equation $a_n = \alpha_1 2^n + \alpha_2(-1)^n$ and the initial conditions $a_0 = 2$ and $a_1 = 7$, it follows that
- $a_0 = 2 = \alpha_1 + \alpha_2$
- $a_1 = 7 = \alpha_1 \cdot 2 + \alpha_2 \cdot (-1)$

- Solving these two equations gives us $\alpha_1 = 3$ and $\alpha_2 = -1$.

- Therefore, the solution to the recurrence relation and initial conditions is the sequence $\{a_n\}$ with
- $a_n = 3 \cdot 2^n - (-1)^n$.

Solving Recurrence Relations

- **Another Example:** Give an explicit formula for the Fibonacci numbers.
- **Solution:** The Fibonacci numbers satisfy the recurrence relation $f_n = f_{n-1} + f_{n-2}$ with initial conditions $f_0 = 0$ and $f_1 = 1$.
- The characteristic equation is $r^2 – r – 1 = 0$.
- Its roots are

$$r_1 = \frac{1+\sqrt{5}}{2}, \quad r_2 = \frac{1-\sqrt{5}}{2}$$

Solving Recurrence Relations

- Therefore, the Fibonacci numbers are given by

$$f_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2} \right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2} \right)^n$$

for some constants α_1 and α_2.
We can determine values for these constants so that the sequence meets the conditions $f_0 = 0$ and $f_1 = 1$:

$$f_0 = \alpha_1 + \alpha_2 = 0$$
$$f_1 = \alpha_1 \left(\frac{1+\sqrt{5}}{2} \right) + \alpha_2 \left(\frac{1-\sqrt{5}}{2} \right) = 1$$

Solving Recurrence Relations

- The unique solution to this system of two equations and two variables is

$$\alpha_1 = \frac{1}{\sqrt{5}},\ \alpha_2 = -\frac{1}{\sqrt{5}}$$

So finally we obtained an explicit formula for the Fibonacci numbers:

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Solving Recurrence Relations

- But what happens if the characteristic equation has only one root?
- How can we then match our equation with the initial conditions a_0 and a_1 ?
- **Theorem:** Let c_1 and c_2 be real numbers with $c_2 \neq 0$. Suppose that $r^2 - c_1 r - c_2 = 0$ has only one root r_0. A sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ if and only if $a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$, for $n = 0, 1, 2, \ldots$, where α_1 and α_2 are constants.

Solving Recurrence Relations

- **Example:** What is the solution of the recurrence relation $a_n = 6a_{n-1} - 9a_{n-2}$ with $a_0 = 1$ and $a_1 = 6$?
- **Solution:** The only root of $r^2 - 6r + 9 = 0$ is $r_0 = 3$. Hence, the solution to the recurrence relation is
- $a_n = \alpha_1 3^n + \alpha_2 n3^n$ for some constants α_1 and α_2.
- To match the initial condition, we need
- $a_0 = 1 = \alpha_1$
  $a_1 = 6 = \alpha_1 \cdot 3 + \alpha_2 \cdot 3$
- Solving these equations yields $\alpha_1 = 1$ and $\alpha_2 = 1$.
- Consequently, the overall solution is given by
- $a_n = 3^n + n3^n$.

Divide-and-Conquer Relations

- Some algorithms take a problem and **successively divide** it into one or more smaller problems until there is a **trivial solution** to them.
- For example, the **binary search** algorithm recursively divides the input into two halves and eliminates the irrelevant half until only one relevant element remained.
- This technique is called **"divide and conquer".**
- We can use **recurrence relations** to analyze the complexity of such algorithms.

Divide-and-Conquer Relations

- Suppose that an algorithm divides a problem (input) of size **n** into **a** subproblems, where each subproblem is of size **n/b**. Assume that **g(n)** operations are performed for such a division of a problem.
- Then, if **f(n)** represents the number of operations required to solve the problem, it follows that f satisfies the recurrence relation
- **$f(n) = af(n/b) + g(n)$**.
- This is called a **divide-and-conquer recurrence relation.**

Divide-and-Conquer Relations

- **Example:** The binary search algorithm reduces the search for an element in a search sequence of size **n** to the binary search for this element in a search sequence of size **n/2** (if n is even).
- **Two** comparisons are needed to perform this reduction.
- Hence, if **f(n)** is the number of comparisons required to search for an element in a search sequence of size n, then
- $\mathbf{f(n) = f(n/2) + 2}$ if n is even.

Divide-and-Conquer Relations

- Usually, we do not try to solve such divide-and conquer relations, but we use them to derive a **big-O estimate** for the complexity of an algorithm.
- **Master Theorem:** Let f be an increasing function that satisfies the recurrence relation
- $f(n) = af(n/b) + cn^d$
- whenever $n = b^k$, where k is a positive integer, a, c, and d are real numbers with $a \geq 1$, and b is an integer greater than 1. Then f(n) is

- $O(n^d)$, if $a < b^d$,
- $O(n^d \log n)$ if $a = b^d$,
- $O(n^{\log_b a})$ if $a > b^d$

Divide-and-Conquer Relations

- **Example:**
- For binary search, we have
$f(n) = f(n/2) + 2$, so $a = 1$, $b = 2$, and $d = 0$
- ($d = 0$ because here, $g(n)$ does not depend on n).
- Consequently, $a = b^d$, and therefore,
$f(n)$ is $O(n^d \log n) = O(\log n)$.
- The binary search algorithm has logarithmic time complexity.

How About Some…

- **Relations**

Relations

- If we want to describe a relationship between elements of two sets A and B, we can use **ordered pairs** with their first element taken from A and their second element taken from B.
- Since this is a relation between **two sets**, it is called a **binary relation**.
- **Definition:** Let A and B be sets. A binary relation from A to B is a subset of $A \times B$.
- In other words, for a binary relation R we have $R \subseteq A \times B$. We use the notation aRb to denote that $(a, b) \in R$ and $a \not{R} b$ to denote that $(a, b) \notin R$.

Relations

- When (a, b) belongs to R, a is said to be **related** to b by R.
- **Example:** Let P be a set of people, C be a set of cars, and D be the relation describing which person drives which car(s).
- P = {Carl, Suzanne, Peter, Carla},
- C = {Mercedes, BMW, tricycle}
- D = {(Carl, Mercedes), (Suzanne, Mercedes), (Suzanne, BMW), (Peter, tricycle)}
- This means that Carl drives a Mercedes, Suzanne drives a Mercedes and a BMW, Peter drives a tricycle, and Carla does not drive any of these vehicles.

Functions as Relations

- You might remember that a **function** f from a set A to a set B assigns a unique element of B to each element of A.
- The **graph** of f is the set of ordered pairs (a, b) such that b = f(a).
- Since the graph of f is a subset of A×B, it is a **relation** from A to B.
- Moreover, for each element **a** of A, there is exactly one ordered pair in the graph that has **a** as its first element and **b** as its second element.

Functions as Relations

- Conversely, if R is a relation from A to B such that every element in A is the first element of exactly one ordered pair of R, then a function can be defined with R as its graph.

- This is done by assigning to an element $a \in A$ the unique element $b \in B$ such that $(a, b) \in R$.

Relations on a Set

- **Definition:** A relation on the set A is a relation from A to A.

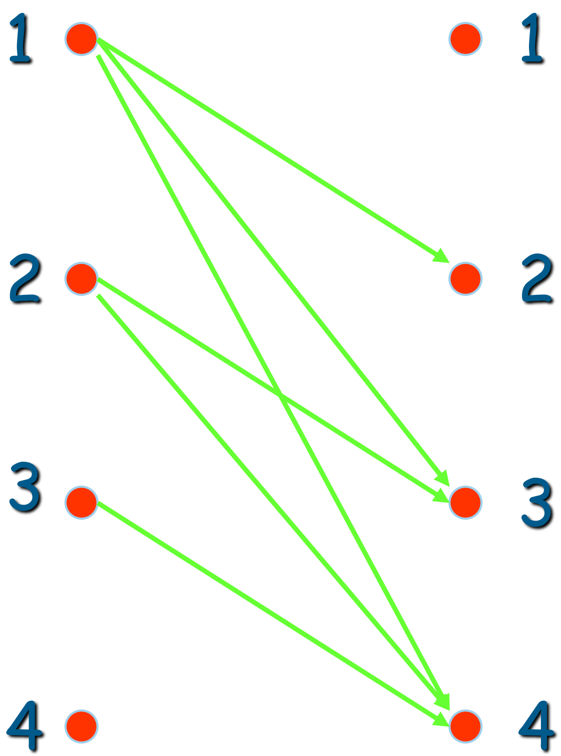
- In other words, a relation on the set A is a subset of $A \times A$.

- **Example:** Let $A = \{1, 2, 3, 4\}$. Which ordered pairs are in the relation $R = \{(a, b) \mid a < b\}$?

UMASS BOSTON

Relations on a Set

- **Solution:** $R = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$



R	1	2	3	4
1		X	X	X
2			X	X
3				X
4				

Relations on a Set

- **How many different relations can we define on a set A with n elements?**

- A relation on a set A is a subset of $A \times A$.
- How many elements are in $A \times A$?

- There are n^2 elements in $A \times A$, so how many subsets (= relations on A) does $A \times A$ have?

- The number of subsets that we can form out of a set with m elements is 2^m. Therefore, 2^{n^2} subsets can be formed out of $A \times A$.

- **Answer:** We can define 2^{n^2} different relations on A.

Properties of Relations

- We will now look at some useful ways to classify relations.
- **Definition:** A relation R on a set A is called **reflexive** if $(a, a) \in R$ for every element $a \in A$.
- Are the following relations on {1, 2, 3, 4} reflexive?

R = {(1, 1), (1, 2), (2, 3), (3, 3), (4, 4)} No

R = {(1, 1), (2, 2), (2, 3), (3, 3), (4, 4)} Yes

R = {(1, 1), (2, 2), (3, 3)} No

Definition: A relation on a set A is called **irreflexive** if $(a, a) \notin R$ for every element $a \in A$.

Properties of Relations

- **Definitions:**
- A relation R on a set A is called **symmetric** if (b, a)$\in$R whenever (a, b)$\in$R for all a, b$\in$A.
- A relation R on a set A is called **antisymmetric** if a = b whenever (a, b)$\in$R and (b, a)$\in$R.
- A relation R on a set A is called **asymmetric** if (a, b)$\in$R implies that (b, a)$\notin$R for all a, b$\in$A.

Properties of Relations

- **Definition:** A relation R on a set A is called **transitive** if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$ for $a, b, c \in A$.

- Are the following relations on {1, 2, 3, 4} transitive?

$R = \{(1, 1), (1, 2), (2, 2), (2, 1), (3, 3)\}$ Yes.

$R = \{(1, 3), (3, 2), (2, 1)\}$ No.

$R = \{(2, 4), (4, 3), (2, 3), (4, 1)\}$ No.

Counting Relations

- **Example:** How many different reflexive relations can be defined on a set A containing n elements?
- **Solution:** Relations on R are subsets of $A \times A$, which contains n^2 elements.
- Therefore, different relations on A can be generated by choosing different subsets out of these n^2 elements, so there are 2^{n^2} relations.
- A **reflexive** relation, however, **must** contain the n elements (a, a) for every $a \in A$.
- Consequently, we can only choose among $n^2 - n = n(n - 1)$ elements to generate reflexive relations, so there are $2^{n(n - 1)}$ of them.

UMASS BOSTON

Combining Relations

- Relations are sets, and therefore, we can apply the usual **set operations** to them.
- If we have two relations R_1 and R_2, and both of them are from a set A to a set B, then we can combine them to $R_1 \cup R_2$, $R_1 \cap R_2$, or $R_1 - R_2$.
- In each case, the result will be **another relation from A to B**.

Combining Relations

- … and there is another important way to combine relations.
- **Definition:** Let R be a relation from a set A to a set B and S a relation from B to a set C. The **composite** of R and S is the relation consisting of ordered pairs (a, c), where $a \in A$, $c \in C$, and for which there exists an element $b \in B$ such that $(a, b) \in R$ and $(b, c) \in S$. We denote the composite of R and S by $\mathbf{S \circ R}$.
- In other words, if relation R contains a pair (a, b) and relation S contains a pair (b, c), then $S \circ R$ contains a pair (a, c).

Combining Relations

- **Example:** Let D and S be relations on A = {1, 2, 3, 4}.
- D = {(a, b) | b = 5 - a} "b equals (5 – a)"
- S = {(a, b) | a < b} "a is smaller than b"
- D = {(1, 4), (2, 3), (3, 2), (4, 1)}
- S = {(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)}
- S∘D = {(2, 4), (3, 3), (3, 4), (4, 2), (4, 3), (4, 4)}

D maps an element a to the element (5 – a), and afterwards S maps (5 – a) to all elements larger than (5 – a), resulting in **S ∘ D = {(a,b) | b > 5 – a}** or **S ∘ D = {(a,b) | a + b > 5}.**

Combining Relations

- We already know that **functions** are just **special cases** of **relations** (namely those that map each element in the domain onto exactly one element in the codomain).

- If we formally convert two functions into relations, that is, write them down as sets of ordered pairs, the composite of these relations will be exactly the same as the composite of the functions (as defined earlier).

Combining Relations

- **Definition:** Let R be a relation on the set A. The powers R^n, n = 1, 2, 3, …, are defined inductively by
- $R^1 = R$
- $R^{n+1} = R^n \circ R$

- In other words:
- $R^n = R \circ R \circ \ldots \circ R$ (n times the letter R)

Combining Relations

- **Theorem:** The relation R on a set A is transitive if and only if $R^n \subseteq R$ for all positive integers n.
- Remember the definition of transitivity:
- **Definition:** A relation R on a set A is called transitive if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$ for $a, b, c \in A$.
- The composite of R with itself contains exactly these pairs (a, c).
- Therefore, for a transitive relation R, $R \circ R$ does not contain any pairs that are not in R, so $R \circ R \subseteq R$.
- Since $R \circ R$ does not introduce any pairs that are not already in R, it must also be true that $(R \circ R) \circ R \subseteq R$, and so on, so that $R^n \subseteq R$.

n-ary Relations

- In order to study an interesting application of relations, namely **databases**, we first need to generalize the concept of binary relations to **n-ary relations**.

- **Definition:** Let $A_1, A_2, \ldots, A_n$ be sets. An **n-ary relation** on these sets is a subset of $A_1 \times A_2 \times \ldots \times A_n$.

- The sets $A_1, A_2, \ldots, A_n$ are called the **domains** of the relation, and n is called its **degree**.

n-ary Relations

- **Example:**
- Let $R = \{(a, b, c) \mid a = 2b \wedge b = 2c \text{ with } a, b, c \in \mathbf{Z}\}$
- What is the degree of R?
- The degree of R is 3, so its elements are triples.
- What are its domains?
- Its domains are all equal to the set of integers.
- Is (2, 4, 8) in R?
- No.
- Is (4, 2, 1) in R?
- Yes.

Databases and Relations

- Let us take a look at a type of database representation that is based on relations, namely the **relational data model.**

- A database consists of n-tuples called **records**, which are made up of **fields**.

- These fields are the **entries** of the n-tuples.

- The relational data model represents a database as an n-ary relation, that is, a set of records.

Databases and Relations

- **Example:** Consider a database of students, whose records are represented as 4-tuples with the fields **Student Name**, **ID Number**, **Major**, and **GPA**:

- R = {(Ackermann, 00231455, CS, 3.88),
  (Adams, 00888323, Physics, 3.45),
  (Chou, 00102147, CS, 3.79),
  (Goodfriend, 00453876, Math, 3.45),
  (Rao, 00678543, Math, 3.90),
  (Stevens, 00786576, Psych, 2.99)}

- Relations that represent databases are also called **tables**, since they are often displayed as tables.

Databases and Relations

- A domain of an n-ary relation is called a **primary key** if the n-tuples are uniquely determined by their values from this domain.
- This means that no two records have the same value from the same primary key.
- In our example, which of the fields **Student Name**, **ID Number**, **Major**, and **GPA** are primary keys?
- **Student Name** and **ID Number** are primary keys, because no two students have identical values in these fields.
- In a real student database, only **ID Number** would be a primary key.

Databases and Relations

- In a database, a primary key should remain a primary key even if new records are added.
- Therefore, we should use a primary key of the **intension** of the database, containing all the n-tuples that can ever be included in our database.

- **Combinations of domains** can also uniquely identify n-tuples in an n-ary relation.
- When the values of a **set of domains** determine an n-tuple in a relation, the **Cartesian product** of these domains is called a **composite key**.

Databases and Relations

- We can apply a variety of **operations** on n-ary relations to form new relations.

- **Definition:** Let *R* be an *n*-ary relation and *C* a condition that elements in *R* may satisfy. Then the **selection operator** s_C maps the *n*-ary relation *R* to the *n*-ary relation of all *n*-tuples from *R* that satisfy the condition *C*.

- **Example:** What is the result when we apply the selection C_1 to the student records, Where C_1 is the condition Major = “Computer Science”?

- **Solution:** The result is the two 4-tuples (Ackermann, 231455, Computer Science, 3.88) and (Chou, 102147, Computer Science,3.49)

UMASS BOSTON

Databases and Relations

- **Definition:** The **projection** $P_{i_1, i_2, \ldots, i_m}$ maps the n-tuple $(a_1, a_2, \ldots, a_n)$ to the m-tuple $(a_{i_1}, a_{i_2}, \ldots, a_{i_m})$, where $m \leq n$.
- In other words, a projection $P_{i_1, i_2, \ldots, i_m}$ keeps the m components $a_{i_1}, a_{i_2}, \ldots, a_{i_m}$ of an n-tuple and deletes its (n – m) other components.
- **Example:** What is the result when we apply the projection $P_{2,4}$ to the student record (Stevens, 00786576, Psych, 2.99) ?
- **Solution:** It is the pair (00786576, 2.99).

UMASS BOSTON

Databases and Relations

- In some cases, applying a projection to an entire table may not only result in fewer columns, but also in **fewer rows**.

- Why is that?

- Some records may only have differed in those fields that were deleted, so they become **identical**, and there is no need to list identical records more than once.

Databases and Relations

- We can use the **join** operation to combine two tables into one if they share some identical fields.

- **Definition:** Let R be a relation of degree m and S a relation of degree n. The **join** $J_p(R, S)$, where $p \leq m$ and $p \leq n$, is a relation of degree $m + n - p$ that consists of all $(m + n - p)$-tuples $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$, where the m-tuple $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p)$ belongs to R and the n-tuple $(c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$ belongs to S.

Databases and Relations

- In other words, to generate $J_p(R, S)$, we have to find all the elements in R whose p last components match the p first components of an element in S.

- The new relation contains exactly these matches, which are combined to tuples that contain each matching field only once.

UMASS BOSTON

Databases and Relations

- **Example:** What is $J_1(Y, R)$, where Y contains the fields **Student Name** and **Year of Birth**,
- Y = {(1978, Ackermann),
  (1972, Adams),
  (1917, Chou),
  (1984, Goodfriend),
  (1982, Rao),
  (1970, Stevens)},
- and R contains the student records as defined before?

Databases and Relations

- **Solution:** The resulting relation is:
- {(1978, Ackermann, 00231455, CS, 3.88),
  (1972, Adams, 00888323, Physics, 3.45),
  (1917, Chou, 00102147, CS, 3.79),
  (1984, Goodfriend, 00453876, Math, 3.45),
  (1982, Rao, 00678543, Math, 3.90),
  (1970, Stevens, 00786576, Psych, 2.99)}

- Since Y has two fields and R has four, the relation $J_1(Y, R)$ has 2 + 4 – 1 = 5 fields.

Representing Relations

- We already know different ways of representing relations. We will now take a closer look at two ways of representation: **Zero-one matrices** and **directed graphs**.

- If R is a relation from $A = \{a_1, a_2, \ldots, a_m\}$ to $B = \{b_1, b_2, \ldots, b_n\}$, then R can be represented by the zero-one matrix $M_R = [m_{ij}]$ with
- $m_{ij} = 1$, if $(a_i, b_j) \in R$, and
- $m_{ij} = 0$, if $(a_i, b_j) \notin R$.

- Note that for creating this matrix we first need to list the elements in A and B in a **particular, but arbitrary order**.

Representing Relations

- **Example:** How can we represent the relation R from the set A = {1, 2, 3} to the set B = {1, 2} with R = {(2, 1), (3, 1), (3, 2)} as a zero-one matrix?

- **Solution:** The matrix M_R is given by

$$M_R = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Representing Relations

- What do we know about the matrices representing a **relation on a set** (a relation from A to A) ?
- They are **square** matrices.
- What do we know about matrices representing **reflexive** relations?
- All the elements on the **diagonal** of such matrices M_{ref} must be **1s**.

$$M_{ref} = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & 1 \end{bmatrix}$$

Representing Relations

- What do we know about the matrices representing **symmetric relations**?
- These matrices are symmetric, that is, $M_R = (M_R)^t$.

$$M_R = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

symmetric matrix,
symmetric relation.

$$M_R = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

non-symmetric matrix,
non-symmetric relation.

Representing Relations

- The Boolean operations **join** and **meet** (you remember?) can be used to determine the matrices representing the **union** and the **intersection** of two relations, respectively.

- To obtain the **join** of two zero-one matrices, we apply the Boolean "or" function to all corresponding elements in the matrices.

- To obtain the **meet** of two zero-one matrices, we apply the Boolean "and" function to all corresponding elements in the matrices.

Representing Relations

- **Example:** Let the relations R and S be represented by the matrices

$$M_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

What are the matrices representing R∪S and R∩S?

Solution: These matrices are given by

$$M_{R\cup S} = M_R \vee M_S = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad M_{R\cap S} = M_R \wedge M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

UMASS BOSTON

Representing Relations Using Matrices

Do you remember the **Boolean product** of two zero-one matrices?

Let $A = [a_{ij}]$ be an $m \times k$ zero-one matrix and $B = [b_{ij}]$ be a $k \times n$ zero-one matrix.

Then the **Boolean product** of A and B, denoted by $A \circ B$, is the $m \times n$ matrix with (i, j)th entry $[c_{ij}]$, where

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \ldots \vee (a_{ik} \wedge b_{kj}).$$

$c_{ij} = 1$ if and only if at least one of the terms $(a_{in} \wedge b_{nj}) = 1$ for some n; otherwise $c_{ij} = 0$.

Representing Relations Using Matrices

Let us now assume that the zero-one matrices $M_A = [a_{ij}]$, $M_B = [b_{ij}]$ and $M_C = [c_{ij}]$ represent relations A, B, and C, respectively.

Remember: For $M_C = M_A \circ M_B$ we have:

$c_{ij} = 1$ if and only if at least one of the terms $(a_{in} \wedge b_{nj}) = 1$ for some n; otherwise $c_{ij} = 0$.

In terms of the relations, this means that C contains a pair (x_i, z_j) if and only if there is an element y_n such that (x_i, y_n) is in relation A and (y_n, z_j) is in relation B.

Therefore, $C = B \circ A$ (composite of A and B).

Representing Relations Using Matrices

This gives us the following rule:

$M_{B \circ A} = M_A \odot M_B$

In other words, the matrix representing the **composite** of relations A and B is the **Boolean product** of the matrices representing A and B.

Analogously, we can find matrices representing the **powers of relations**:

$M_{R^n} = M_R^{[n]}$ (n-th **Boolean power**).

Representing Relations Using Matrices

- **Example:** Find the matrix representing R^2, where the matrix representing R is given by

$$M_R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Solution: The matrix for R^2 is given by

$$M_{R^2} = M_R^{[2]} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Representing Relations Using Digraphs

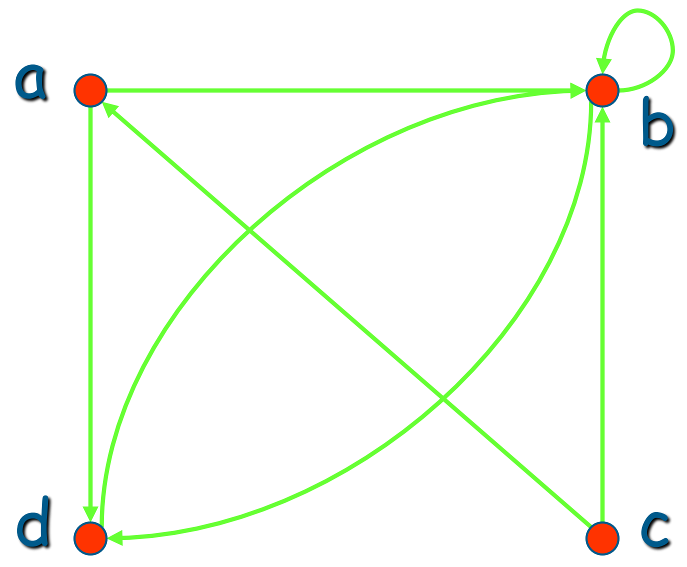
- **Definition:** A **directed graph**, or **digraph**, consists of a set V of **vertices** (or **nodes**) together with a set E of ordered pairs of elements of V called **edges** (or **arcs**).

- The vertex a is called the **initial vertex** of the edge (a, b), and the vertex b is called the **terminal vertex** of this edge.

- We can use arrows to display graphs.

Representing Relations Using Digraphs

- **Example:** Display the digraph with V = {a, b, c, d}, E = {(a, b), (a, d), (b, b), (b, d), (c, a), (c, b), (d, b)}.



An edge of the form (b, b) is called a **loop.**

Representing Relations Using Digraphs

- Obviously, we can represent any relation R on a set A by the digraph with A as its vertices and all pairs (a, b)$\in$R as its edges.

- Vice versa, any digraph with vertices V and edges E can be represented by a relation on V containing all the pairs in E.

- This **one-to-one correspondence** between relations and digraphs means that any statement about relations also applies to digraphs, and vice versa.