CS 2305: Discrete Mathematics for Computing I

Lecture 23

- KP Bhat

Applications of Congruences

For this section only concepts, no computations, on the exam

Section 4.5

Check Digits

- A common method of detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function. If the final digit is not correct, then the string is assumed not to be correct

Parity Check Bits (1)

- Digital information is represented by bit string, split into blocks of a specified size
- Before each block is stored or transmitted, an extra bit, called a **parity check bit**, is appended to each block. The parity check bit x_{n+1} for the bit string $x_1x_2\ldots x_n$ is defined by
  - **$x_{n+1} = x_1 + x_2 + \cdots + x_n \bmod 2$**
- x_{n+1} is 0 if there are an even number of 1 bits in the block of n bits and it is 1 if there are an odd number of 1 bits in the block of n bits
  - Number of 1s, after the addition of the parity bit is even
- When we examine a string that includes a parity check bit, we know that there is an error in it if the parity check bit (the least significant bit) is wrong. However, when the parity check bit is correct, there still may be an error since a parity check can detect an odd number of errors in the previous bits, but not an even number of errors

Parity Check Bits (2)

Example: Suppose we receive in a transmission the bit strings 01100101 and 11010110, each ending with a parity check bit. Should we accept these bit strings as correct?

Solution:

01100101

0 + 1 + 1 + 0 + 0 + 1 + 0

3 ≡ 1 (mod 2)

The expected parity bit is 1. Hence the parity check bit is correct

11010110

1 + 1 + 0 + 1 + 0 + 1 + 1

5 ≡ 1 (mod 2)

The expected parity bit is 1. Hence the parity check bit is incorrect

Check Digits: UPCs

Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit is determined by the congruence:

$$3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \pmod{10}.$$

Example:

Is 793573431042 a well-formed UPC code?

Solution:

The first 11 digits of the UPC are 79357343104. In this case the check digit should be:

$3\cdot7 + 9 + 3\cdot3 + 5 + 3\cdot7 + 3 + 3\cdot4 + 3 + 3\cdot1 + 0 + 3\cdot4 + x_{12} \equiv 0 \pmod{10}$

$98 + x_{12} \equiv 0 \pmod{10}$

The solution of this congruence (beyond the scope of this course) is

$x_{12} \equiv 2 \pmod{10}$

So, the check digit is 2.

Hence 793573431042 is a well-formed UPC code

Check Digits:ISBNs

Books are identified by an *International Standard Book Number* (ISBN-10), a 10 digit code. The first 9 digits identify the language, the publisher, and the book. The tenth digit is a check digit, which is determined by the following congruence.

$$x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11}.$$

X is used for the digit 10.

Example:

Is 0072880082 a well-formed ISBN-10?

Solution:

The first 9 digits of the UPC are 007288008. In this case the check digit should be:

$X_{10} \equiv 1·0 + 2·0 + 3·7 + 4·2 + 5·8 + 6·8 + 7·0 + 8·0 + 9·8 \pmod{11}$.
$X_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$.
$X_{10} \equiv 189 \pmod{11}$.

The solution of this congruence (beyond the scope of this course) is $X_{10} = 2$.

Hence 0072880082 is a well-formed ISBN-10 code

Integer Representations ~~and Algorithms~~

Section 4.2

Representations of Integers

In the modern world, we use *decimal,* or *base* 10, *notation* to represent integers. For example when we write 965, we mean $9\cdot10^2 + 6\cdot10^1 + 5\cdot10^0$.

We can represent numbers using any base *b*, where *b* is a positive integer greater than 1.

The bases *b* = 2 (*binary*), *b* = 8 (*octal*) , and *b* = 16 (*hexadecimal*) are important for computing and communications

The ancient Mayans used base 20 and the ancient Babylonians used base 60.

Base *b* Representations

Theorem 1: Let b be a positive integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form:

$$n = a_k b^k + a_{k-1} b^{k-1} + \dots. + a_1 b + a_0$$

where k is a nonnegative integer, $a_0, a_1, \dots. a_k$ are nonnegative integers less than b, and $a_k \neq 0$. The a_j, $j = 0, \dots, k$ are called the base-b digits of the representation.
(This theorem can be proved using mathematical induction, which is discussed in Section 5.1.)

The representation of n given in Theorem 1 is called the *base b expansion of n* and is denoted by $(a_k a_{k-1} \dots. a_1 a_0)_b$.

We usually omit the subscript 10 for base 10 expansions.

Binary Expansions

Most computers represent integers and do arithmetic with binary (base 2) expansions of integers. In these expansions, the only digits used are 0 and 1.

Example: What is the decimal expansion of the integer that has $(1\,0101\,1111)_2$ as its binary expansion?

Solution:

$$(\overset{8}{1}\,\overset{7}{0}\overset{6}{1}\overset{5}{0}\overset{4}{1}\,\overset{3}{1}\overset{2}{1}\overset{1}{1}\overset{0}{1})_2 = 1\cdot 2^8 + 0\cdot 2^7 + 1\cdot 2^6 + 0\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 351.$$

Example: What is the decimal expansion of the integer that has $(11011)_2$ as its binary expansion?

Solution: $(\overset{4}{1}\overset{3}{1}\overset{2}{0}\overset{1}{1}\overset{0}{1})_2 = 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 27.$

Octal Expansions

The octal expansion (base 8) uses the digits {0,1,2,3,4,5,6,7}.

Example: What is the decimal expansion of the number with octal expansion $(7016)_8$?

Solution: $(\overset{3}{7}\overset{2}{0}\overset{1}{1}\overset{0}{6})_8 = 7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 6 \cdot 8^0 = 3598$

Example: What is the decimal expansion of the number with octal expansion $(111)_8$?

Solution: $(\overset{2}{1}\overset{1}{1}\overset{0}{1})_8 = 1 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 = 64 + 8 + 1 = 73$

Hexadecimal Expansions

The hexadecimal expansion needs 16 digits, but our decimal system provides only 10. So letters are used for the additional symbols. The hexadecimal system uses the digits {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}. The letters A through F represent the decimal numbers 10 through 15.

Example: What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?

Solution:

$(\overset{4}{2}\overset{3}{A}\overset{2}{E}\overset{1}{0}\overset{0}{B})_{16} = 2\cdot16^4 + 10\cdot16^3 + 14\cdot16^2 + 0\cdot16^1 + 11\cdot16^0$
$=175627$

Example: What is the decimal expansion of the number with hexadecimal expansion $(E5)_{16}$?

Solution: $(\overset{1}{E}\overset{0}{5})_{16} = 14\cdot16^1 + 5\cdot16^0 = 224 + 5 = 229$

Base Conversion[1]

To construct the base *b* expansion of an integer *n*:

- Divide *n* by *b* to obtain a quotient and remainder.
  $n = bq_0 + a_0 \quad 0 \leq a_0 \leq b$
- The remainder, a_0, is the rightmost digit in the base *b* expansion of *n*. Next, divide q_0 by *b*.
  $q_0 = bq_1 + a_1 \quad 0 \leq a_1 \leq b$
- The remainder, a_1, is the second digit from the right in the base *b* expansion of *n*.
- Continue by successively dividing the quotients by *b*, obtaining the additional base *b* digits as the remainder. The process terminates when the quotient is 0.

Base Conversion$_2$

```
procedure base b expansion($n$, $b$: positive integers with $b > 1$)
$q := n$ {initialize quotient to the number}
$k := 0$ {initialize digit to rightmost digit}
while ($q \neq 0$)
    $a_k := q$ mod $b$ {set  the k$^{th}$ digit from right}
    $q := q$ div $b$ {update the quotient}
    $k := k + 1$ {advance leftward to the next digit position}
return($a_{k-1}$ ,…, $a_1$,$a_0$){($a_{k-1} \ldots a_1a_0)_b$ is base $b$ expansion of $n$}
```

Base Conversion$_3$

Example: Find the octal expansion of $(12345)_{10}$

Solution: Successively dividing by 8 gives:

- $12345 = 8 \cdot 1543 + \boxed{1}$
- $1543 = 8 \cdot 192 + \boxed{7}$
- $192 = 8 \cdot 24 + \boxed{0}$
- $24 = 8 \cdot 3 + \boxed{0}$
- $3 = 8 \cdot 0 + \boxed{3}$

The remainders are the digits from right to left yielding $(\overset{4}{3}\overset{3}{0}\overset{2}{0}\overset{1}{7}\overset{0}{1})_8$.

Conversion Between Binary, Octal, and Hexadecimal Expansions[1]

Conversion between binary and octal and between binary and hexadecimal expansions is extremely easy because each octal digit corresponds to a block of three binary digits and each hexadecimal digit corresponds to a block of four binary digits,

Comparison of Hexadecimal, Octal, and Binary Representations

TABLE 1 Hexadecimal, Octal, and Binary Representation of the Integers 0 through 15.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

Initial 0s are not shown

Each octal digit corresponds to a block of 3 binary digits.

Each hexadecimal digit corresponds to a block of 4 binary digits.

So, conversion between binary, octal, and hexadecimal is easy.

Conversion Between Binary, Octal, and Hexadecimal Expansions$_2$

Example: Find the octal and hexadecimal expansions of $(11111010111100)_2$.

Solution:

- To convert to octal, we group the digits into blocks of three $(011\ 111\ 010\ 111\ 100)_2$, adding initial 0s as needed. The blocks from left to right correspond to the digits 3,7,2,7, and 4. Hence, the solution is $(37274)_8$.
- To convert to hexadecimal, we group the digits into blocks of four $(0011\ 1110\ 1011\ 1100)_2$, adding initial 0s as needed. The blocks from left to right correspond to the digits 3,E,B, and C. Hence, the solution is $(3EBC)_{16}$.

Conversion Between Binary, Octal, and Hexadecimal Expansions[3]

Example: Find the binary expansions of $(765)_8$ and $(A8D)_{16}$.

Solution:

- To convert $(765)_8$ into binary notation, we replace each octal digit by a block of three binary digits. These blocks are 111, 110, and 101. Hence, $(765)_8 = (1\ 1111\ 0101)_2$.
- To convert $(A8D)_{16}$ into binary notation, we replace each hexadecimal digit by a block of four binary digits. These blocks are 1010, 1000, and 1101. Hence, $(A8D)_{16} = (1010\ 1000\ 1101)_2$.

Induction and recursion

Chapter 5

Mathematical Induction

Section 5.1

Mathematical Induction

- A powerful proof technique that is used to check conjectures about the outcomes of processes that occur repeatedly and according to definite patterns
- Based on the rule of inference that if $P(1)$ and $\forall k(P(k) \rightarrow P(k+1))$ are true for the domain of positive integers, then $\forall n P(n)$ is true

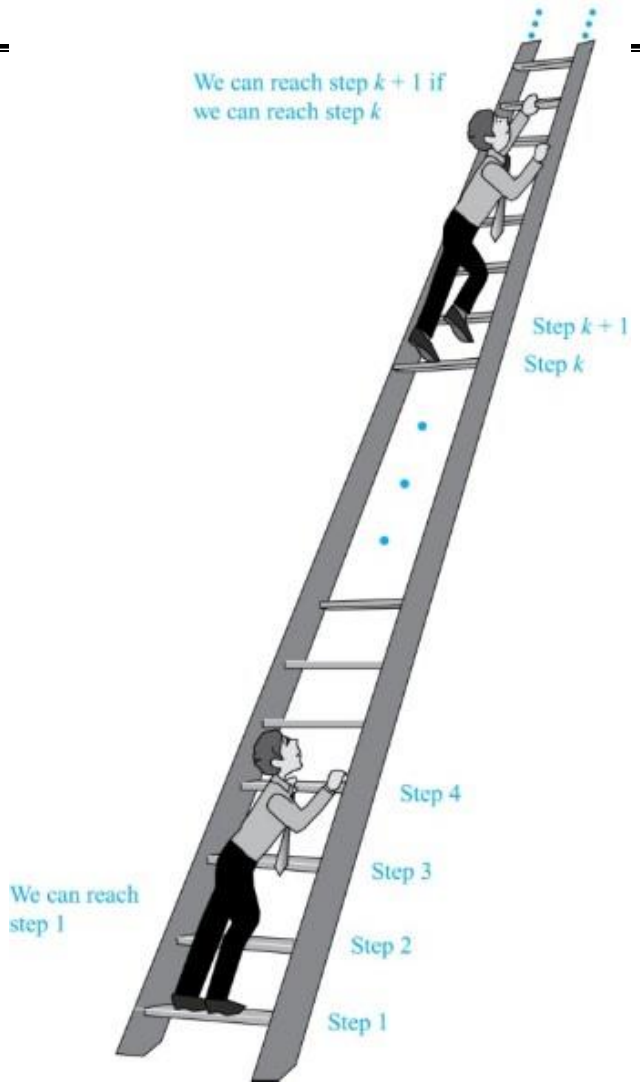
Climbing an Infinite Ladder

Suppose we have an infinite ladder:

1. We can reach the first rung of the ladder.
2. If we can reach a particular rung of the ladder, then we can reach the next rung.

From (1), we can reach the first rung. Then by applying (2), we can reach the second rung. Applying (2) again, the third rung. And so on. We can apply (2) any number of times to reach any particular rung, no matter how high up.

This example illustrates the reasoning employed for proof by mathematical induction.



Principle of Mathematical Induction

Principle of Mathematical Induction: To prove that the propositional function $P(n)$ is true for all positive integers n, we complete these steps:

- *Basis Step*: Show that $P(1)$ is true.
- *Inductive Step*: Show that $P(k) \to P(k + 1)$ is true for all positive integers k.

To complete the inductive step, assuming the *inductive hypothesis* that $P(k)$ holds for an arbitrary integer k, show that $P(k + 1)$ must be true.

Climbing an Infinite Ladder Example:

1. We can reach the first rung of the ladder.
2. If we can reach a particular rung of the ladder, then we can reach the next rung.

- BASIS STEP: By (1), we can reach rung 1.
- INDUCTIVE STEP: Assume the inductive hypothesis that we can reach rung k. Then by (2), we can reach rung $k + 1$.

Hence, $P(k) \to P(k + 1)$ is true for all positive integers *k.* We can reach every rung on the ladder.

Note:- To prove that $P(n)$ is true for all positive integers $n \geq a$, in the Basis Step we show that P(a) is true.

Important Points About Using Mathematical Induction

Mathematical induction can be expressed as the rule of inference

$$\left(P(1) \wedge \forall k\left(P(k) \rightarrow P(k+1)\right)\right) \rightarrow \forall n\, P(n),$$

where the domain is the set of positive integers.

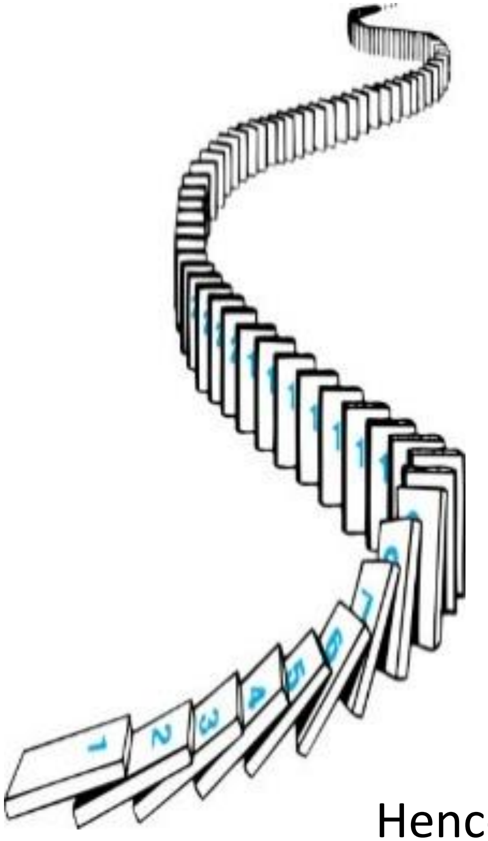
In a proof by mathematical induction, we don’t assume that *P*(*k*) is true for all positive integers! We show that if we assume that *P*(*k*) is true, then *P*(*k* + 1) must also be true. It is therefore not a case of begging the question, or circular reasoning

Proofs by mathematical induction do not always start at the integer 1. In such a case, the basis step begins at a starting point *b* where *b* is an integer. We will see examples of this soon.

Remembering How Mathematical Induction Works

Consider an infinite sequence of dominoes, labeled 1,2,3, …, where each domino is standing.

Let $P(n)$ be the proposition that the nth domino is knocked over.



We know that the first domino is knocked down, i.e., $P(1)$ is true .

We also know that if whenever the kth domino is knocked over, it knocks over the $(k + 1)$st domino, i.e, $P(k) \rightarrow P(k + 1)$ is true for all positive integers k.

Hence, all dominos are knocked over.

$P(n)$ is true for all positive integers n.

Jump to long description

Proving a Summation Formula by Mathematical Induction

Example: Show that: $\displaystyle\sum_{i=1}^{n} = \frac{n(n+1)}{2}$

Solution:

Note: Once we have this conjecture, mathematical induction can be used to prove it correct.

- BASIS STEP: *P*(1) is true since 1(1 + 1)/2 = 1.
- INDUCTIVE STEP: Assume true for *P*(*k*).

The inductive hypothesis is $\displaystyle\sum_{i=1}^{k} = \frac{k(k+1)}{2}$

Under this assumption,

$$\begin{aligned} 1+2+\ldots+k+(k+1) &= \frac{k(k+1)}{2}+(k+1) \\ &= \frac{k(k+1)+2(k+1)}{2} \\ &= \frac{(k+1)(k+2)}{2} \end{aligned}$$

Adding (k+1) to both sides

- Hence, we have shown that *P*(*k* + 1) follows from *P*(*k*). Therefore the sum of the first *n* integers is $\frac{n(n+1)}{2}$