Discrete Structures Spring 2024 – Week 2

Sequence And Sums

Lecture 1

Introduction

- A discrete structure to represent an ordered lists of elements.
  - 1, 2, 3, 5, 8
  - 1, 3, 9, 27, 81, …….
- Sequences arise throughout mathematics, computer science, and in many other disciplines, ranging from botany to music.
- We will introduce the terminology to represent sequences and sums of the terms in the sequences.

Sequences

Definition: A *sequence* is a function from a subset of the integers (usually either the set {0, 1, 2, 3, 4, …..} or {1, 2, 3, 4, ….}) to a set *S*.

- The notation a_n is used to denote the image of the integer *n*.
- We can think of a_n as the equivalent of *f(n)* where *f* is a function from {0,1,2,…..} to *S*.

- We call a_n a *term* of the sequence

Sequences

$$a_n = \frac{1}{n}$$

Example: Consider the sequence $\{a_n\}$ where

- The list of the terms of this sequence, beginning with a_1 would be

$$\{a_n\} = \{a_1, a_2, a_3, \ldots\}$$

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \dots$$

Arithmetic Progression

Definition: A *arithmetic progression* is a sequence of the form:

$$a, a + d, a + 2d, \dots, a + nd, \dots$$

where the *initial term a* and the *common difference d* are real numbers.

Examples:

1. Let $a = -1$ and $d = 4$:

$$\{s_n\} = \{s_0, s_1, s_2, s_3, s_4, \ldots\} = \{-1, 3, 7, 11, 15, \ldots\}$$

2. Let $a = 7$ and $d = -3$:

$$\{t_n\} = \{t_0, t_1, t_2, t_3, t_4, \ldots\} = \{7, 4, 1, -2, -5, \ldots\}$$

3. Let $a = 1$ and d = 2:

$$\{u_n\} = \{u_0, u_1, u_2, u_3, u_4, \ldots\} = \{1, 3, 5, 7, 9, \ldots\}$$

Geometric Progression

Definition: A *geometric progression* is a sequence of the form:

$$a, ar, ar^2, \ldots, ar^n, \ldots$$

where the *initial term a* and the *common ratio r* are real numbers.

Examples:

1. Let $a = 1$ and $r = -1$. Then:

$$\{b_n\} = \{b_0, b_1, b_2, b_3, b_4, \ldots\} = \{1, -1, 1, -1, 1, \ldots\}$$

2. Let $a = 2$ and $r = 5$. Then:

$$\{c_n\} = \{c_0, c_1, c_2, c_3, c_4, \ldots\} = \{2, 10, 50, 250, 1250, \ldots\}$$

3. Let $a = 6$ and $r = 1/3$. Then:

$$\{d_n\} = \{d_0, d_1, d_2, d_3, d_4, \ldots\} = \{6, 2, \frac{2}{3}, \frac{2}{9}, \frac{2}{27}, \ldots\}$$

Recurrence Relations

- In previous examples, sequences are specified by providing explicit formulas for their terms

- Another way to specify a sequence is to provide one or more initial terms together with a rule for determining subsequent terms from those that precede them

Recurrence Relations

Definition: A *recurrence relation* for the sequence $\{a_n\}$ is an equation that expresses a_n in 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.
- The *initial conditions* for a sequence specify the terms that precede the first term where the recurrence relation takes effect.

Questions about Recurrence Relations

- **Example 1**: Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 1,2,3,4,\ldots$. and suppose that $a_0 = 2$. What are a_1, a_2 and a_3?
  [Here $a_0 = 2$ is the initial condition.]

- **Solution**: We see from the recurrence relation that

$$a_1 = a_0 + 3 = 2 + 3 = 5$$
$$a_2 = a_1 + 3 = 5 + 3 = 8$$
$$a_3 = a_2 + 3 = 8 + 3 = 11$$

Questions about Recurrence Relations

- **Example 2**: Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} – a_{n-2}$ for $n = 2,3,4,\ldots.$ and suppose that $a_0 = 3$ and $a_1 = 5$. What are a_2 and a_3?

- [Here the initial conditions are $a_0 = 3$ and $a_1 = 5$]
- **Solution**: We see from the recurrence relation that
- $a_2 = a_1 - a_0 = 5 – 3 = 2$
- $a_3 = a_2 – a_1 = 2 – 5 = –3$

Fibonacci Sequence

Definition: Define the *Fibonacci sequence*, $f_0, f_1, f_2, \ldots$, by: Initial Conditions: $f_0 = 0$, $f_1 = 1$, Recurrence Relation: $f_n = f_{n-1} + f_{n-2}$

Example: Find f_2, f_3, f_4, f_5 and f_6

Answer:

$$f_2 = f_1 + f_0 = 1 + 0 = 1$$
$$f_3 = f_2 + f_1 = 1 + 1 = 2$$
$$f_4 = f_3 + f_2 = 2 + 1 = 3$$

$$f_5 = f_4 + f_3 = 3 + 2 = 5$$
$$f_6 = f_5 + f_4 = 5 + 3 = 8$$

Solving Recurrence Relations

- Finding a formula for the *n*th term of the sequence generated by a recurrence relation is called *solving the recurrence relation*
- Such a formula is called a *closed formula*.

Solving Recurrence Relations

- **Example 1**: Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 1,2,3,4,\ldots.$ and suppose that $a_0 = 2$. Solve the recurrence relation and initial condition.

[Here $a_0 = 2$ is the initial condition.]

Iterative Solution Example

Method 1: Working upward, forward substitution

Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 2,3,4,\ldots$. and suppose that $a_1 = 2$.

$$a_2 = 2 + 3$$
$$a_3 = (2 + 3) + 3 = 2 + 3 \cdot 2$$
$$a_4 = (2 + 2 \cdot 3) + 3 = 2 + 3 \cdot 3$$
$$\vdots$$
$$a_n = a_{n-1} + 3 = (2 + 3 \cdot (n - 2)) + 3 = 2 + 3(n - 1)$$

Iterative Solution Example

Method 2: Working downward, backward substitution

Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} + 3$ for $n = 2,3,4,\ldots$ and suppose that $a_1 = 2$.

$$
\begin{aligned}
a_n &= a_{n-1} + 3 \\
&= (a_{n-2} + 3) + 3 = a_{n-2} + 3 \cdot 2 \\
&= (a_{n-3} + 3) + 3 \cdot 2 = a_{n-3} + 3 \cdot 3 \\
&\quad \vdots \\
&= a_2 + 3(n-2) = (a_1 + 3) + 3(n-2) = 2 + 3(n-1)
\end{aligned}
$$

Summations

$$a_m, a_{m+1}, \ldots, a_n$$

- Sum of the terms from the sequence $\{a_n\}$
- The notation:

$$\sum_{j=m}^{n} a_j$$

represents

- The variable j is called the *index of summation*. It runs through all the integers starting with its *lower limit m* and ending with its *upper limit n*.

Summations

S: • **Examples**:

- More generally for a set

Some Useful Summation Formulae

Geometric Series: We just proved this.

Later we

will prove
some of
these by
induction.

Proof in text
(requires calculus)

Propositional Logic

Lecture 2

Section Summary

- Propositions
- Connectives
  - Negation, Conjunction, Disjunction, Implication
  - Contrapositive, Inverse, Converse
  - Bi-conditional
- Truth Tables

Propositions

• A *proposition* is a declarative sentence that is either true or false. • Examples of propositions:

a) The Moon is made of green cheese.

b) Trenton is the capital of New Jersey.

c) Toronto is the capital of Canada.

d) $1 + 0 = 1$

e) $0 + 0 = 2$

Propositions

- Examples that are not propositions.

a) Sit down!

b) What time is it?

c) $x + 1 = 2$

d) $x + y = z$

Propositional Logic

- Constructing Propositions
- Propositional Variables: p, q, r, s, …
- The proposition that is always true is denoted by **T** and the proposition

that is always false is denoted by **F**.

- Compound Propositions – constructed from logical connectives and other propositions
- Negation ¬, Conjunction ∧, Disjunction ∨, Implication →, Biconditional ↔

Compound Propositions: Negation

- The *negation* of a proposition p is denoted by $\neg p$ and has this truth table:

p		F	T
T			

- **Example**: If p denotes "The earth is round.", then $\neg p$ denotes "It is not the case that the earth is round," or more simply "The earth is not round."

Compound Propositions: Conjunction

- The *conjunction* of propositions p and q is denoted by $p \wedge q$ and has this truth table:

p	q	p ∧ q
T	T	T
T	F	F
F	T	F
F	F	F

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \wedge q$ denotes "I am at home and it is raining."

Compound Propositions: Disjunction

- The

disjunction of propositions p and q is denoted by $p \vee q$ and has this

truth table:

F	T	T
F	F	F

p
T
T

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \vee q$ denotes "I am at home or it is raining."

The Connective Or in English

- In English "or" has two distinct meanings.

- “Inclusive Or” - In the sentence “Students who have taken CS202 or Math120 may take this class,” we assume that students need to have taken one of the prerequisites, but may have taken both. This is the meaning of disjunction. For $p \vee q$ to be true, either one or both of p and q must be true.
- “Exclusive Or” - When reading the sentence “Soup or salad comes with this entrée,” we do not expect to be able to get both soup and salad. This is the meaning of Exclusive Or (Xor). In $p \oplus q$, one of p and q must be true, but not both. The truth table for $\oplus$ is:

The Connective Or in English

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Compound Propositions: Implication

- If p and q are propositions, then $p \rightarrow q$ is a *conditional statement* or *implication* which is read as "if p, then q" and has this truth table:

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Compound Propositions: Implication

- **Example**: If p denotes "I am at home." and q denotes "It is raining." then $p \rightarrow q$ denotes "If I am at home then it is raining."

- In $p \rightarrow q$, p is the *hypothesis* (*antecedent* or *premise*) and q is the *conclusion* (or *consequence*).

Understanding Implication

- In $p \rightarrow q$ there does not need to be any connection between the antecedent or the consequent. The "meaning" of $p \rightarrow q$ depends only on the truth values of p and q.

- These implications are perfectly fine, but would not be used in ordinary English.
- *"If the moon is made of green cheese, then I have more money than Bill Gates."*
- *"If the moon is made of green cheese then I'm on welfare."*
- *"If $1 + 1 = 3$, then your grandma wears combat boots."*

Understanding Implication (cont)

- One way to view the logical conditional is to think of an obligation or contract.
- *"If I am elected, then I will lower taxes."*
- *"If you get 100% on the final, then you will get an A."*

- If the politician is elected and does not lower taxes, then the voters can say that he or she has broken the campaign pledge. Something similar holds for the professor. This corresponds to the case where p is true and q is false.

Different Ways of Expressing $p \rightarrow q$

- if p, then q p implies q
- if p, q p only if q
- q unless ¬p q when p
- q if p p is sufficient for q
- q whenever p q is necessary for p
- q follows from p a necessary condition for p is *q* a sufficient condition for q is p

Converse, Contra-Positive, and Inverse

- From $p \rightarrow q$ we can form new conditional statements
- $q \rightarrow p$ is the **converse** of $p \rightarrow q$
- $\neg q \rightarrow \neg p$ is the **contrapositive** of $p \rightarrow q$

- $\neg p \to \neg q$ is the **inverse** of $p \to q$

Converse, Contra-Positive, and Inverse

- **Example**: Find the converse, inverse, and contra-positive of “It raining is a sufficient condition for my not going to town.”
- **Solution:**

- **converse**: If I do not go to town, then it is raining.
- **inverse**: If it is not raining, then I will go to town.
- **contrapositive**: If I go to town, then it is not raining.

Bi-Conditional

- If p and q are propositions, then we can form the *biconditional* proposition $p \leftrightarrow q$, read as "p if and only if q." The biconditional $p \leftrightarrow q$ denotes the proposition with this truth table:

p	q	p q
T	T	T
T	F	F
F	T	F
F	F	T

Bi-Conditional

• If p denotes “I am at home.” and q denotes “It is raining.” •
Then p ↔q denotes “I am at home if and only if it is raining.”

Expressing the Bi-Conditional

- Some alternative ways "*p* if and only if *q*" is expressed in English:

- **p is necessary and sufficient for q**
- **if p then q , and conversely**
- **p iff q**

Truth Tables For Compound Propositions

- Construction of a truth table:
- Rows – need a row for every possible combination of values for the atomic propositions.
- Columns - need a column for the compound proposition (usually at far right). Need a column for the truth value of each expression that occurs in the compound proposition as it is built up. This includes the atomic propositions

Example Truth Table

- Construct a truth table for

T	T
T	T
T	F

T	F
F	T
F	T
F	F
F	F

F
T
F
T

F
T
F
T

T
T
T
T

T
T
F
F

F
T
F
T
F

T
T
T

Equivalent Propositions

- Two propositions are **e***quivalent* if they always have the same truth value.
- **Example**: Show using a truth table that the conditional is equivalent to the contra-positive.

- **Solution:**

T	T	F	F	T	T
T	F	F	T	F	F
F	T	T	F	T	T
F	F	T	T	T	T

Using a Truth Table to Show Non-Equivalence

- **Example**: Show using truth tables that neither the converse nor inverse of an implication are not equivalent to the implication.

- **Solution:**

T T F F T T T

T	F	F	T	F	T	T
F	T	T	F	T	F	F
F	F	T	T	T	T	T

Problem

- How many rows are there in a truth table with *n* propositional variables? • **Solution**: 2^n

- Note that this means that with n propositional variables, we can construct 2^n distinct (i.e., not equivalent) propositions.

Precedence of Logical Operators

$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

- $p \vee q \rightarrow \neg r$ is equivalent to $(p \vee q) \rightarrow \neg r$
- If the intended meaning is $p \vee (q \rightarrow \neg r)$ then parentheses must be used.

Applications of Propositional Logic

Lecture 3

Applications of Propositional Logic

1. Translating English to Propositional Logic

2. System Specifications
3. Logic Circuits

Translating English Sentences

- Steps to convert an English sentence to a statement in propositional logic ✔Identify atomic propositions and represent using propositional variables. ✔Determine appropriate logical connectives

Translating English Sentences – Example I

- "If I go to Harry's or to the country, I will not go shopping."
- p: I go to Harry's
- q: I go to the country.
- r: I will go shopping.

If p or q then not r.

Translating English Sentences – Example II

- Translate the following sentence into propositional logic:
- “You can access the Internet from campus only if you are a computer science major or you are not a freshman.”
- Let a, c, and f represent respectively “You can access the internet from campus,” “You are a computer science major,” and “You are a freshman.”
- $a \rightarrow (c \lor \neg f)$

System Specifications

- System and Software engineers take requirements in English and express them in a precise specification language based on logic.
- **Example**: Express in propositional logic:

"The automated reply cannot be sent when the file system is full"

- **Solution**: One possible solution: Let p denote "The automated reply can be sent" and q denote "The file system is full."

$q \rightarrow \neg p$

Consistent System Specifications

- **Definition**: A list of propositions is *consistent* if it is possible to assign truth values to the proposition variables so that each proposition is true

Consistent System Specifications – Exercise

- Are these specifications consistent?

- “The diagnostic message is stored in the buffer or it is retransmitted.”
- “The diagnostic message is not stored in the buffer.”
- “If the diagnostic message is stored in the buffer, then it is retransmitted.”

Consistent System Specifications – Solution

- Let p denote "The diagnostic message is stored in the buffer." • Let q denote "The diagnostic message is retransmitted" • The specification can be written as: $p \vee q, \neg p, p \rightarrow q$. • When **p** is false and **q** is true all three statements are true, hence the specification is consistent
- Could come to same conclusion by use of a truth table **(HW)**

Consistent System Specifications

- What if another proposition “The diagnostic message is not retransmitted is added.”
- Added $\neg q$
- There is no satisfying assignment, so the specification is not consistent.

Logic Gates

- Electronic circuits; each input/output signal can be viewed as a 0 or 1
- 0 represents False, 1 represents True
- Circuits are constructed from three basic circuits called gates
- The inverter (NOT gate)takes an input bit and produces the negation of that bit. The OR gate takes two input bits and produces the value equivalent to the disjunction of the two bits. The AND gate takes two input bits and produces the value equivalent to the conjunction of the

two bits.

Logic Gates

Logic Gates

- More complicated digital circuits can be constructed by combining these basic circuits to produce the desired output given the input signals by building a circuit for each piece of the output expression and then combining them. For example:

Practice Exercises

- Section 2.4 Page 177
- Questions 4, 7, 17 (a, b, c), 29

- Section 1.1 Page 13
- Questions 11, 14, 29, 34 (a, b, c)

- Section 1.2 Page
- Questions 5, 10, 47