Fall 2021 - MATH 1101 Discrete Structures
Lecture 7

- **Introduction**
- **PART 1. Well-Ordering Property.**
- **PART 2. Recursive Definitions and Structural Induction.**

Exercises. Set 1 (Solved Problems)

Exercises. Set 2 (Supplementary Problems)

Introduction.

In Lecture 6 we continue to discuss proof technique and analyze the third equivalent form of Mathematical Induction called as **Well-Ordering Property** (WOP) (Part 1). In Part 2 we discuss Structural Induction technique for proving results about recursively defined sets. This technique can be used to proof statements in theoretical Computer Science, Set theory, and Logic.

PART 1. Well-Ordering Property (WOP).

We start our considerations from **axiomatic definition** of the set of Positive Integers. First, remind you that we have two binary operations on the **set of all integers Z**={…, -3, -2, -1, 0 , 1, 2, 3 …}, namely, addition (+) and multiplication (∗) which satisfy several properties.

Definition 1. The **set of Positive Integers N** is defined as the subset of the set of integers **Z** satisfying four key properties (axioms) A1–A4:

A1: The number 1 is a positive number.

A2: If n is a positive number, then n+1, the *successor* of n, is also a positive integer.

A3: Every positive integer other than 1 is the successor of a positive integer.

A4: (The Well-Ordering Property, WOP). Every nonempty subset of the set of positive integers has a least element ($n \in S \subset \mathbf{N}$ is called a least element if n has no predecessor in S, that is, there is no $m \in S$ such that m+1=n).

■

Note. Sometimes, for the special purposes, we use the extended version of WOP: Every **nonempty subset** of the set of *nonnegative* integers has a least element. ■

Theorems 1 and 2 below show that WOP and MI are equivalent each other, WOP ⇔MI.

Theorem 1. WOP implies that **MI is a valid proof technique**. In other word, when WOP is satisfied, the Basis and Inductive steps of MI entail the validity of P(n) for any positive integer.

Proof. Suppose we know that P(1) is true (Basis Step of MI) and the proposition P(k)→P(k+1) is true for all positive integer k (Inductive Step). We must show that P(n) is true for all positive integer n.

Assume the contrary, namely, assume that there is at least one positive integer for which P(n) is false. Then the set S of positive integers for which P(n) is false is nonempty. Thus, by WOP, S has a least element, which will be denoted by m. We know that m cannot be 1, because P(1) is true. Because m is positive and greater than 1, (m-1) is a positive integer. Furthermore, because (m-1) is less than m, it is not in S, so P(m-1) must be true. Because the conditional statement P(m-1)→P(m) is also true, it must be the case that P(m) is true. This contradicts the choice of m. Hence, P(n) must be true for every positive integer n. ■

Conversely, WOP can be proved when the MI is taken as an axiom.

A4′: Mathematical Induction axiom (MI). If S is ***a*** set of positive integers such that 1∈S and for all positive integers n if n∈S, then (n+1)∈S, then S is ***the*** set of **all positive** integers.

Replace **A4** by **A4′** in the system of axioms **A1-A4.**

Theorem 2. The WOP is true when the MI is taken as an axiom, or, in other words, MI implies WOP.

Proof. Suppose that the well-ordering property were false. Let S be a nonempty set of positive integers that has no least element. Let P(n) be the statement "i∉S for i=1, . . . , n". Then P(1) is true (that is, 1∉S) because if 1∈S then S has a least element, namely, 1. Now suppose that P(n) is true. Thus, 1∉S, ..., n∉S. Clearly, (n+1) cannot be in S, for if it were, it would be its least element. Thus P(n+1) is true. So, by the MI, n∉S for all nonnegative integers n. Thus, S=∅, a contradiction. ■

In the Theorem 1, Lecture 6, the equivalence SI ⇔ MI was proved. Therefore, all three proof techniques are equivalent:

SI ⇔ MI ⇔ WOP

The well-ordering property can often be used directly in proofs.

EXAMPLE 1. Use **WOP** to prove the division algorithm:

If ***a*** is an integer and ***d*** is a **positive** integer,

Then there are unique integers ***q*** and ***r*** with $0 \le r < d$ and $a = dq + r$.

Proof.

Existence. We use WOP to prove existence part. Let S be the set of nonnegative integers of the form $a-dq$, where q is an integer, that is, given ***a*** and ***d*** let $S=\{a-dq \mid a-dq \ge 0, q \in \mathbf{Z}\}$. This set is nonempty because $(-dq)$ can be made as large as desired (taking q to be a negative integer with large absolute value). By WOP, S has a least element $r=a-dq_0$ which is also nonnegative, $0 \le r$.

It is also the case that $r<d$. Assume the contrary, assume that $r \ge d$ then $r-d \ge 0$. Substitute the value of r into the last inequality: $(a-dq_0)-d \ge 0 \Rightarrow a-d(q_0+1) \ge 0$. Clear that $a-d(q_0+1) \in S$ and smaller than r. But it is impossible because r is the least element in S. Therefore, our assumption $r \ge d$ is false and $r<d$. Consequently, there are integers q and r with $0 \le r < d$ and $a=dq+r$.

Uniqueness. Assume that $a=dq+r=dq'+r'$ with $0 \le r < d$ and $0 \le r' < d$. Then $d(q-q')=r'-r$. Hence d divides $r'-r$. On the other hand, $0 \le r < d$ and $0 \le r' < d$ imply $-d < r'-r < d$. Therefore, $r'-r=0$. Hence, $r'=r$. It follows that $q=q'$, as claimed. ■

EXAMPLE 2. In a round-robin tournament every player plays every other player exactly once and each match has a winner and a loser. We say that the players $p_1, p_2, ..., p_m$ form a cycle if p_1 beats p_2, p_2 beats p_3, ..., p_{m-1} beats p_m, and p_m beats p_1.

Input: There is a cycle of length m (m≥3) among the players in a round-robin tournament.

Output: There must be a cycle of **three** of these players. Prove it using WOP.

Proof. Let S be a set of positive integers n≥3 such that there is a cycle of the length n. S is nonempty because, by *Input*, there is at least one cycle of length m (m≥3) in the round-robin tournament. By the WOP, **S has a least element k**, which by assumption must be greater or equal than three. Consequently, there exists a cycle of players $p_1, p_2, ..., p_k$ and **no shorter cycle exists**.

Assume the contrary, that is, assume that, under hypothesis of the problem, there is no cycle of three players. Hence, k>3. Consider the first three elements of this cycle, p_1, p_2, and p_3. There are two possible outcomes of the match between p_1 and p_3. If p_3 beats p_1, it follows that p_1, p_2, p_3 is a cycle of length three, contradicting our assumption that there is no cycle of three players. Consequently, it must be the case that p_1 beats p_3. This means that we can omit p_2 from the cycle $p_1, p_2, ..., p_k$ to obtain the cycle $p_1, p_3, p_4, ..., p_k$ of length k−1, contradicting the assumption that the smallest cycle has length k. We conclude that there must be a cycle of length three. ■

EXAMPLE 3. Use WOP to show that **there is a unique** greatest common divisor of two positive integers.

Proof. We suggest a solution consisting of five steps.

Let a and b be positive integers, a>0, b>0, and let S={as+bt | as+bt>0, s and t are integers}.

Step 1. Show that S is nonempty.

Step 2. Use the WOP to show that S has a smallest positive element c.

Step 3. Show that c is a common divisor of a and b, in denotations, c|a and c|b.

Step 4. **Existing** of a greatest common divisor of a and b, gcd(a, b). Show that if d>0 is a common divisor of a and b, then d is a divisor of c, that is, in denotations, **if** d|a and d|b **then** d|c therefore, d≤c and conclude that c=gcd(a, b).

Step 5. **Uniqueness** of gcd(a, b). Finish the proof by showing that c is unique.

Step 1. S is nonempty because at least both a (if s=1 and t=0) and b (if s=0 and t=1) belong to S.

Step 2. By WOP, the set S has a smallest element, say c, c>0, c∈S which means existing of integers x and y such that c=ax+by.

Step 3. Assume the contrary, assume that c does not divide a. Then, by Example 1, there exist q and r such that a=cq+r, 0<r<c (r≠0, otherwise c divides a). Now rewrite last equation as r=a(1)+c(-q). Since r>0, so r∈S. On the other side, r<c which contradicts the property of c to be the smallest element in S. Thus, c divides a. By the similar arguments c divides b. Therefore, c is a common divisor of a and b.

Step 4. Let d>0 is a common divisor of a and b, d|a and d|b. Then a=de and b=df, where e and f are positive integers. Since c=ax+by so c=ax+by=(de)x+(df)y=d(ex+fy). Hence, d divides c, d|c. Since both c and d are positive so (ex+fy) is also the positive integer. Therefore, d≤c and we conclude c=gcd(a, b).

Step 5. Let d>0 is another gcd(a, b), d=gcd(a, b). Hence, d|a and d|b and c≤d. On the other hand, we proved in Step 4 that for any common divisor of a and b (including d) we have $d \le c$. Therefore, $d=c$ and uniqueness is proved. ■

WOP is an existence axiom. It does not tell us which element is the smallest integer, nor does it tell us how to find the smallest element. To find a smallest element we need to attract some other tools.

EXAMPLE 4. Find a smallest element, if any, of the following set S of positive integers

$$S=\{n\in \mathbf{N} \mid n=x^2-8x+12 \text{ for some } x\in \mathbf{Z}\}, \text{ where } \mathbf{Z} \text{ is the set of all integers}$$

Solution. First, S is nonempty because at least for all positive integer $x\ge 7$ all integers $x^2-8x+12$ are positive. By WOP, the set S has a smallest element (existence). To determine the smallest element in S, we need to solve the inequality $x^2-8x+12>0$. Factorization leads to $x^2-8x+12= =(x-2)(x-6)>0$, so we need $x<2$ or $x>6$. Because $x\in Z$, we determine that the minimum value of $x^2-8x+12$ occurs at $x=1$ or $x=7$. Since $1^2-8\cdot 1+12=5$ and $7^2-8\cdot 7+12=5$, then the smallest element in S is 5. ■

The reader may find a few more solved problems and exercises at the end of these Lecture Notes (EXERCISES. SET 1 and 2).

Probably the reader noticed that in all Examples above we used "proof by contradiction" technique to apply WOP. We summarize our observations as a **Template for WOP proofs**.

To prove that "P(n) is true for all n∈N" using the WOP:

- Define the set, C, of counterexamples to P being true. Specifically, define C as:

$$C=\{n\in N \mid \text{NOT } (P(n)) \text{ is true}\}=\{n\in N \mid \neg P(n)\}$$

- Assume for proof by contradiction that C is nonempty.
- By the WOP, there will be a smallest element, n, in C.
- Reach a contradiction somehow—often by showing that P(n) is actually true or by showing that there is another member of C that is smaller than n. This is the open-ended part of the proof task.
- Conclude that C must be empty, that is, no counterexamples exist.

PART 2. Recursive Definitions and Structural Induction.

Recursive data types play a central role in programming, and induction is really all about them.

Recursive data types are specified by *recursive definitions*, which say how to construct new data elements from previous ones. Along with each recursive data type there are recursive definitions of properties (or functions) on the data type. Most importantly, based on a recursive definition, there is a *structural induction* method for proving that all data of the given type have some property.

What is the idea of recursive definitions? Sometimes, it is difficult to define an object explicitly. However, it may be easy to define this object in terms of itself. This process is called *recursion*.

We can use recursion to define **functions** and **sets**. In most beginning mathematics courses, the terms of a sequence (special kind of **function**) are specified using an explicit formula. For instance, the sequence of powers of 2 is given by $a_n=2^n$ for n=0, 1, 2, ... We can also define a sequence *recursively* by specifying how terms of the sequence are found from previous terms. The sequence of powers of 2 can also be defined by giving

1. the **first term** of the sequence, namely, $a_0=1$, and
2. **a rule for finding a term of the sequence from the previous one**, namely, $a_{n+1}=2a_n$ for n=0, 1, 2, ...

Below we provide definition of recursively defined sequences(functions), explain why recursive approach is well-defined, and discuss examples.

Recursively Defined Functions.

Let $\mathbf{N_0}$ be the set of nonnegative integers: $\mathbf{N_0}=\mathbf{N}\cup\{0\}$, here N={1, 2, 3, …} is the set of positive integers.

We know that any **sequence** of numbers: $a_0, a_1, \ldots a_n, \ldots$ is, indeed, an explicitly defined **rule of assignment** (=**function**) f: $\mathbf{N_0}\to\mathbf{R}$. Under such approach to define f(n) we need no to know the values of the function f at the points m<n or m>n. This is exactly the property of explicitly defined function: to evaluate f(n) we do not need to know a value of the function f at other point.

We propose now another, two steps approach, to define a sequence (= function on $\mathbf{N_0}$):

Definition 2. Recursive (or induction) definition of a sequence consists of two steps

- **Basis Step**: Specify the value of a function f at initial point (for instance, at zero if Domain is $\mathbf{N_0}$).
- **Recursive Step**: Give a rule for finding values of the function f at an integer from its values at smaller integers.

EXAMPLE 5. Suppose that f is defined recursively by

Basis Step: f(0)=2

Recursive Step: f(n+1)=3f(n)-2

Find f (1), f (2), and f (4).

Solution. From recursive definition it follows:

f(1)=3f(0)-2=3(2)-2=6-2=4

f(2)=3f(1)-2=3(4)-2=12-2=10

To find f(4) we need to know f(3). Accordingly,

f(3)=3f(2)-2=3(10)-2=30-2=28

f(4)=3f(3)-2=3(28)-2=84-2=82 ■

EXAMPLE 6. Provide a recursive definition of a^n, where a is a nonzero real number and n is a nonnegative integer.

Solution. From recursive definition it follows:

Basis Step: Set $a^0=1$

Recursive Step: Define $a^{n+1}=a\cdot a^n$ for n=0, 1, 2,… .

These two steps (equations) uniquely define a^n for all nonnegative integers n. ■

EXAMPLE 7. Given sequence $\{a_n\}$, n=0, 1, 2, … provide a recursive definition of partial sums function:

$$S_n=S(n)=\sum_{k=0}^{n} a_k$$

Solution.

Basis Step: $S(0)=\sum_{k=0}^{0} a_k=a_0$

Recursive Step: $S(n+1)=\sum_{k=0}^{n+1} a_k=(\sum_{k=0}^{n} a_k)+a_{n+1}$ ■

Note 1. We use ***induction*** **(MI or SI)** to prove results about recursively defined sequences. ■

From mathematical point of view the following important question arises:

Whether or not recursively defined functions are well-defined?

Answer: YES. That is, for every positive integer, the value of the function at this integer is determined in an unambiguous way. This means that given any positive integer, we can use the two parts of the definition to find the value of the function at that integer, and that we obtain the same value no matter how we apply the two parts of the definition. This is a consequence of the principle of mathematical induction. (*Exercise*).

In some recursive definitions of functions, the values of the function at the first k positive integers are specified, and a rule is given for determining the value of the function at larger integers from its values at some or all of the preceding k integers. That recursive definitions defined in this way produce well-defined functions follows from strong induction. Example of such kind of recursive function is Fibonacci sequence (Fibonacci numbers) (see EXERCISE SET 1, exercise 1.8).

Recursively Defined Sets and Structures.

In previous section we have explored **how functions can be defined recursively**. We now turn our attention to **how sets can be defined recursively**.

Recursive definitions of sets have two parts, a basis step and a recursive step, as in the case of recursive definition of functions:

Definition 3. Recursive definition of a set consists of two steps

- **Basis Step:** an initial collection of elements of the set is specified.
- **Recursive Step:** rules for forming new elements in the set from those already known to be in the set are provided.

Recursive definitions may also include an exclusion rule, which specifies that a recursively defined set contains nothing other than those elements specified in the basis step or generated by applications of the recursive step. In our discussions, we assume that the exclusion rule holds and no element belongs to a recursively defined set unless it is in the initial collection specified in the basis step or can be generated using the recursive step one or more times.

EXAMPLE 8. Consider the subset S of the set of integers recursively defined by

Basis Step: $3 \in S$.

Recursive Step: If $x \in S$ and $y \in S$, then $x+y \in S$.

The new elements found to be in S are 3 by the basis step, 3+3=6 at the first application of the recursive step, 3+6=6+3=9 and 6+6=12 at the second application of the recursive step, and so on. We will show later that S is the set of all positive multiples of 3. ■

We continue examples considering new objects: strings and functions(operations) over strings.

Definition 4. Let Σ be a set (no matter finite or infinite). We call Σ as an *alphabet* and elements of it as *letters* (*symbols, bits*). Any finite sequence of symbols from Σ is called a *string* over an alphabet Σ. Set of all strings over Σ is denoted as Σ^*. The *length* of a string (the function $l:\Sigma^* \to \mathbf{N_0}$) is the number of terms in this string. The *empty* string, denoted by λ, is the string that has no terms. The empty string has *length* zero. ■

Sequences of the form $a_1, a_2, ..., a_n$ are often used in computer science. These finite sequences are also called *strings* according to the Definition 4. This string is also denoted by $a_1a_2...a_n$.

EXAMPLE 9. The string abcd is a string of length four. ■

Recursive approach to define <u>strings</u> and <u>length.</u>

Now, we focus our attention **on recursive approach to define <u>strings</u> and <u>length</u> of strings.** Recursive definitions play an important role in the study of strings.

Idea of the recursive definition of strings over the given alphabet Σ is as following. At the basis step of the recursive definition of strings we assert that the empty string belongs to the set of strings Σ^*. At the recursive step we state that new strings are produced by adding a symbol from the alphabet Σ to the end of produced strings in Σ^*. At each application of the recursive step, strings containing one additional symbol are generated.

The strong recursive Definition of Σ^*, the set of strings over Σ, is as following.

Definition 5. The set Σ^* of *strings* over the alphabet Σ is defined recursively by

Basis Step: $\lambda \in \Sigma^*$ (where λ is the empty string containing no symbols).

Recursive Step: If $w \in \Sigma^*$ and $x \in \Sigma$, then $wx \in \Sigma^*$ ■

EXAMPLE 10. If $\Sigma=\{0, 1\}$, the strings found to be in Σ^*, **the set of all bit strings**, are

- λ, specified to be in Σ^* in the basis step,
- 0 and 1 formed during the first application of the recursive step,
- 00, 01, 10, and 11 formed during the second application of the recursive step, and so on. ■

Consider another application of recursive approach to define operations or functions on elements of recursively defined sets. This is illustrated in Definition 6 of the **concatenation** of two strings and Definition 7 concerning the length of a string.

Definition 6. Let Σ be a set of symbols and Σ^* be the set of strings formed from symbols in Σ. We can define the concatenation of two strings, denoted by "·", recursively as follows (Note. We would often omit the "·" symbol and write just ab rather than a·b).

Basis Step: If $w \in \Sigma^*$, then $w\lambda=w$, where λ is the empty string.

Recursive Step: If $w_1 \in \Sigma^*$ and $w_2 \in \Sigma^*$ and $x \in \Sigma$, then $w_1(w_2x)=(w_1w_2)x$. ■

By repeated application of the recursive definition, it follows that the concatenation of two strings w_1 and w_2 consists of the symbols in w_1 followed by the symbols in w_2. For instance, the concatenation of w_1=abra and w_2=cadabra is w_1w_2=abracadabra.

Definition 7. The **length of a string** w, denoted as ***l*(w)**, can be recursively defined by

Basis Step: $l(\lambda)=0$;

Recursive Step: $l(wx)=l(w)+1$ if $w\in\Sigma^*$ and $x\in\Sigma$. ■

Finally, at the end of this Section let us summarize the known information on recursive definitions of functions and sets.

	Recursive definition of a function	**Recursive definitions of a set**
1	**Basis Step**: Specify the *value* of the sequence (function) *at initial point(s).*	**Basis Step:** Specify an initial collection of elements of the **set**.
2	**Recursive Step**: *Rule(s)* for finding values of the function at an integer from its values at smaller integers are provided. Each application of the rules generates new elements in the sequence.	**Recursive Step: Rule(s)** for forming new elements from those already known to be in the set are provided. Each application of the rules generates new elements in the set.
3	***Induction* (MI or SI)** is used to prove results about recursively defined sequences.	***Structural Induction* (StrI)** is used to prove results about recursively defined sets.

Structural Induction.

To prove results about recursively defined sets, we generally use some form of mathematical induction. Example 11 illustrates the connection between recursively defined sets and mathematical induction.

EXAMPLE 11. Show that the set S defined in Example 8 by specifying that $3\in S$ and that if $x\in S$ and $y\in S$, then $x+y\in S$, is the set of all positive integers that are multiples of 3.

Solution: Let A be the set of all positive integers divisible by 3. To prove that A=S, we must show that A is a subset of S and that S is a subset of A. To prove that A is a subset of S, we must show that every positive integer divisible by 3 is in S. We will use mathematical induction to prove this.

Let P(n) be the statement that 3n belongs to S. The basis step holds because by the first part of the recursive definition of S, 3(1)=3 is in S. To establish the inductive step, assume that P(k) is true, namely, that 3k is in S. Because 3k is in S and because 3 is in S, it follows from the second part of the recursive definition of S that 3k+3=3(k+1) is also in S. Therefore, by MI P(n) belongs to S for every positive integer n.

Conversely. To prove that S is a subset of A, we use the recursive definition of S. First, the basis step of the definition specifies that 3 is in S. Because 3=3(1), all elements specified to be in S in this step are divisible by 3 and are therefore in A. To finish the proof, we must show that all integers in S generated using the second part of the recursive definition are in A. This consists of showing that x+y is in A whenever x and y are elements of S also assumed to be in A. Now if x and y are both in A, it follows that 3|x and 3|y. Therefore 3|(x+y), completing the proof. ■

In Example 11 we used mathematical induction over the set of positive integers and a recursive definition to prove a result about a recursively defined set. However, instead of using mathematical induction directly to prove results about recursively defined sets, we can use a more convenient form of induction known as **structural induction**.

Definition 8. A proof by **Structural Induction** (**StrI**) consists of two parts.

Basis Step: Show that the statement holds for all elements specified in the basis step of the recursive definition to be in the set.

Recursive Step: Show that **IF** the statement is true for each of the elements used to construct new elements in the recursive step of the definition, **THEN** the statement holds for these new elements.

Theorem 3. Structural Induction is a valid proof technique.

Proof. It follows from the principle of mathematical induction for the nonnegative integers. Namely, let P(n) state that the claim is true for all elements of the set that are generated by n or fewer applications of the rules in the recursive step of a recursive definition. We will have established that the principle of mathematical induction implies the principle of structural induction if we can show that P(n) is true whenever n is a positive integer.

In the basis step of a proof by structural induction we show that P(0) is true. That is, we show that the result is true of all elements specified to be in the set in the basis step of the definition. A consequence of the recursive step is that if we assume P(k) is true, it follows that P(k+1) is true. When we have completed a proof using structural induction, we have shown that P(0) is true and that P(k) implies P(k+1). By mathematical induction it follows that P(n) is true for all nonnegative integers n. This also shows that the result is true for all elements generated by the recursive definition and shows that structural induction is a valid proof technique. ■

Structural induction can be used to prove that **all members of a set constructed recursively have a particular property** as the Definition 8 asserts**.**

We will illustrate this idea by using structural induction to prove results about strings. First, we have to carry out the appropriate basis step and the appropriate recursive step. Algorithm of applying Structural Induction for the set of strings is:

Suppose that P(w) is a propositional function over the set of strings $w \in \Sigma^*$. To use structural induction to prove that P(w) holds for all strings $w \in \Sigma^*$, we need to complete both basis and recursive steps. These steps are:

Basis Step: Show that P(λ) is true.

Recursive Step: Assume that P(w) is true, where $w \in \Sigma^*$. Show that if $x \in \Sigma$, then P(wx) must also be true.

EXAMPLE 12. Use Structural Induction to prove that $l(xy)=l(x)+l(y)$, where x and y belong to Σ^*, the set of strings over the alphabet Σ. (Here $l(x)$ – length of string x).

Solution: We will base our proof on the recursive definition of the set Σ^* given in Definition 5 and the definition of the length of a string in Definition 7, which specifies that $l(\lambda)=0$ and $l(wx)=l(w)+1$ when $w\in\Sigma^*$ and $x\in\Sigma$. Let P(y) be the statement that $l(xy)=l(x)+l(y)$ whenever x belongs to Σ^*.

Basis Step: To complete the basis step, we must show that P(λ) is true. That is, we must show that $l(x\lambda)=l(x)+l(\lambda)$ for all $x\in\Sigma^*$. Because $l(x\lambda)=l(x)=l(x)+0=l(x)+l(\lambda)$ for every string x, it follows that P(λ) is true.

Recursive Step: To complete the inductive step, we assume that P(y) is true and show that this implies that P(ya) is true whenever $a\in\Sigma$. What we need to show is that $l(xya)=l(x)+l(ya)$ for every $a\in\Sigma$. To show this, note that by the recursive definition of $l(w)$ (given in Definition 7), we have $l(xya)=l(xy)+1$ and $l(ya)=l(y)+1$. And, by the inductive hypothesis, $l(xy)=l(x)+l(y)$. We conclude that $l(xya)=l(x)+l(y)+1=l(x)+l(ya)$. ■

GENERALIZED INDUCTION (Optional Reading).

Remaining part of this Lecture Notes - GENERALIZED INDUCTION (except EXERCISES. SET 1 and SET 2) is for the optional reading.

We can extend mathematical induction to prove results about other sets that have the well-ordering property besides the set of integers.

Definition 9. Define **a lexicographic ordering** on $\mathbf{N_0}\times\mathbf{N_0}$, the ordered pairs of nonnegative integers, by specifying that

(x_1, y_1) is less than or equal to (x_2, y_2) if either $x_1<x_2$, or $x_1=x_2$ and $y_1<y_2$. ■

We denote "less or equal" as symbol $\preccurlyeq$.

Example.

(a) $(3, 4)\preccurlyeq(5, 2)$ because $3<5$
(b) $(3, 4)\preccurlyeq(3, 5)$ because $4<5$

Theorem 4. The set $\mathbf{N_0}\times\mathbf{N_0}$ with this ordering has the property that every subset of $\mathbf{N_0}\times\mathbf{N_0}$ has a least element.

Proof. (Exercise). ■

Theorem 4 implies that we can recursively define the terms $a_{m,n}$, with $m\in\mathbf{N_0}$ and $n\in\mathbf{N_0}$, and prove results about them using a variant of mathematical induction, as illustrated in example below.

EXAMPLE 13. Suppose that $a_{m,n}$ is defined recursively for $(m, n)\in\mathbf{N_0}\times\mathbf{N_0}$ by

Basis Step: $a_{0,0}=0$.

Recursive Step:

$$a_{m,n} = \begin{cases} a_{m-1,n} + 1, & if\ n = 0, m > 0 \\ a_{m,n-1} + n, & if\ n > 0 \end{cases} \qquad (1)$$

Prove that for all $(m, n)\in\mathbf{N_0}\times\mathbf{N_0}$ we have

$$a_{m,n}=m+\frac{n(n+1)}{2} \qquad (2)$$

In other words, the sequence $a_{m,n}$ can be defined both recursively and explicitly.

Solution: We interpret values of the sequence $a_{m,n}$ as values of the function $a_{m,n}$: $\mathbf{N_0}\times\mathbf{N_0}\to\mathbf{N_0}$. Domain of $a_{m,n}$ is the set Dom ($a_{m,n}$)=$\mathbf{N_0}\times\mathbf{N_0}$={(m, n)|m≥0, n≥0, m and n are integers}. We prove that $a_{m,n}=m+\frac{n(n+1)}{2}$ using a generalized version of mathematical induction for which:

- The basis step requires that we show that this formula is valid when (m, n)=(0, 0).
- The induction step requires that we show that:

  IF the formula holds for all pairs smaller than (m, n) in the lexicographic ordering of $\mathbf{N_0}\times\mathbf{N_0}$,

  THEN it also holds for (m, n).

Basis Step: Let (m, n)=(0, 0). Then by the basis case of the recursive definition of $a_{m,n}$ we have $a_{0,0}=0$. Furthermore, when m=n=0, $m+n(n+1)/2=0+(0\cdot 1)/2=0$. This completes the basis step.

Inductive Step: Suppose that $a_{m',n'}=m'+n'(n'+1)/2$ whenever $(m', n')\preccurlyeq(m, n)$ in the lexicographic ordering of $\mathbf{N_0}\times\mathbf{N_0}$. By the recursive definition of $a_{m,n}$ we must consider two cases:

- if n=0, then $a_{m,n}=a_{m-1,n}+1$. Because (m−1, n) is smaller than (m, n), the inductive hypothesis tells us that $a_{m-1,n}=(m-1)+\frac{n(n+1)}{2}$, so that $\boldsymbol{a_{m,n}}=\left[(m-1)+\frac{n(n+1)}{2}\right]+1=m+\frac{n(n+1)}{2}$, giving us the desired equality.
- If n>0, so $a_{m,n}=a_{m,n-1}+n$. Because (m, n−1) is smaller than (m, n), the inductive hypothesis tells us that $a_{m,n-1}=m+\frac{(n-1)n}{2}$, so $\boldsymbol{a_{m,n}}=\left(m+\frac{(n-1)n}{2}\right)+n=m+\frac{(n^2-n+2n)}{2}=m+\frac{n(n+1)}{2}$

This finishes the inductive step. ■

EXERCISES. SET 1 (Solved Problems).

Well-Ordering Property

1.1. Every positive integer more than 1 has a prime divisor.

Solution. **Let S be the set of positive integers >1 with no prime divisor**. We are going to prove that S is empty. Assume the contrary. Suppose S is nonempty. By WOP, the set S has a smallest element. Let n be the smallest element of S. Note that n cannot be prime, since n divides itself and if n were prime, it would be its own prime divisor. Thus, n is composite: it must have a divisor d with 1<d<n. But then d must have a prime divisor (by the minimality of n). Call it p. Then p|d, but d|n, so p|n. This is a contradiction. Therefore, S is empty. ■

1.2. Every positive integer more than 1 can be written as a product of one or more primes.

Solution. We apply here the same approach as in previous exercise. **Let S be the set of positive integers >1 that cannot be written as a product of primes.** We are going to prove that S is empty. Assume the contrary. Suppose S is nonempty. By WOP, the set S has a smallest element. Let n be its smallest element. Note that n is not prime, since then n=n would be a prime factorization. So, n is composite: n=de with 1<d, e<n. But d and e can be written as a product of primes, by the minimality of n. So, n is a product of products of primes and hence is a product of primes itself. This is a contradiction. Therefore, S is empty. ■

Note 1. Clear that statements on 1.1 and 1.2 are equivalent and they are so-called "existence" statements. We say nothing about how to find "prime" decomposition of a positive integer more than 1.

Note 2. In Lecture Notes 10-11, Example 6 we presented another proof of Exercise 1.2 based on Strong Induction. It was expectable because as we mentioned in Part 1 of this Lecture Notes MI, SI, and WOP are equivalent.

1.3. Use WOP to show that every amount of postage that can be paid exactly using only 10 cent and 15 cent stamps, is divisible by 5.

Solution. Let S(n) be the statement:

S(n)= "n cents postage can be assembled using only 10 and 15 cent stamps".

We must prove that

S(n)→5|n; for all positive integers n for which S(n) is true. **(1.3 - 1)**

Proof. We apply the Template of using WOP from the Lecture Notes.

Fill in the missing portions (indicated by "••• ") of the following proof of (1.3 - 1).

Let C be the set of counterexamples to (1.3 - 1), namely:

$$C=\{n \mid S(n) \text{ and NOT}(5|n)\}=\{ n \mid S(n) \text{ NOT}(5|n)\}$$

We must prove that C is empty. Assume for the purpose of obtaining a contradiction that C is nonempty. Then by the WOP, there is a smallest number $m \in C$. Then S(m-10) or S(m-15) must hold because the m cents postage is made from 10 and 15 cent stamps, so we remove one.

So, suppose S(m-10) holds. Then 5|(m-10), because •••

But if 5|(m-10), then 5|m, because •••

contradicting the fact that m is a counterexample.

Next suppose S(m-15) holds. Then the proof for m-10 carries over directly for m-15 to yield a contradiction in this case as well.

Since we get a contradiction in both cases, we conclude that C must be empty. That is, there are no counterexamples to (1.3 - 1), which proves that (1.3 - 1), holds. ■

1.4. Use WOP to prove that

$$1+2+\ldots+n=n(n+1)/2. \qquad (1.4-1)$$

for all positive integers $n \in \mathbf{N}$

Proof.

By contradiction. Assume that (1.4 – 1) is false. Then, some positive integers serve as counterexamples to it. Let us collect them in a set:

$$C=\{n \in \mathbf{N} \mid 1+2+\ldots+n \neq n(n+1)/2\}.$$

Assuming there are counterexamples, C is a nonempty set of positive integers.

So, by WOP, C has a minimum element, which we'll call **c**. That is, among the positive integers, **c** is the smallest counterexample to equation (1.4 – 1).

Since c is the smallest counterexample, we know that (1.4 – 1) is false for n=c but true for all positive integers n<**c**. But (1.4 – 1) is true for n=1, so **c**>1. This means **c**-1>0, that is, **c**-1 is a positive integer, and since it is less than **c**, equation (1.4 – 1) is true for **c**-1. That is

$1+2+\ldots+(c-1)=(c-1)c/2$

But then, adding **c** to both sides, we get

$1+2+\ldots+(c-1)+c=\frac{(c-1)c}{2}+c=\frac{c^2-c+2c}{2}=\frac{c(c+1)}{2}$

which means that (1.4–1) does hold for **c**. This is a contradiction, and we are done. ■

1.5. Use WOP to prove that for all nonnegative integers we have:

$$\sum_{k=0}^{n} k^2 = \frac{n(n+1)(2n+1)}{6} \qquad (1.5-1)$$

Proof. By contradiction.

Let C be the set of counterexamples to (1.5 – 1). Namely,

$$C=\left\{c\in\mathbf{N}\cup\{\mathbf{0}\} \mid \sum_{k=0}^{c} k^2 \neq \frac{c(c+1)(2c+1)}{6}\right\} \qquad (1.5-2)$$

If C is nonempty then it has a minimal element, say c_0, $c_0\in C$. Such c_0 must be positive because for n=0 both sides of (1.5 - 1) are 0. Since c_0 is minimum of C and $c_0>0$ so $m=c_0-1$ being nonnegative **does** satisfy (1.5 – 1). Hence we have:

$$\sum_{k=0}^{m} k^2 = \frac{m(m+1)(2m+1)}{6} = \frac{(c_0-1)c_0(2(c_0-1)+1)}{6} = \frac{(c_0-1)c_0(2c_0-1)}{6}$$

Then we obtain the following chain:

$$\sum_{k=0}^{c_0} k^2 = \sum_{k=0}^{m} k^2 + c_0^2 = \frac{(c_0-1)c_0(2c_0-1)}{6} + c_0^2 = \frac{(c_0^2-c_0)(2c_0-1)+6c_0^2}{6}$$

$$= \frac{2c_0^3-3c_0^2+c_0+6c_0^2}{6} = \frac{2c_0^3+3c_0^2+c_0}{6} = \frac{c_0(c_0+1)(2c_0+1)}{6}$$

That is c_0 does satisfy (1.5 – 1) after all. WE obtain the contradiction: c_0 does not satisfy (1.5 – 1). Hence, C is empty and (1.5 – 1) holds for every nonnegative integer n.

1.6. Use WOP to prove that **there is no** solution over the positive integers to the equation:

$$4a^3+2b^3=c^3 \qquad (1.6-1)$$

Proof.

Assume the contrary. Assume that (1.6 – 1) has a positive solution (a>0, b>0, c>0). Define S={a>0 | there exist b>0, c>0 and which satisfy (1.6 – 1)}

According to our assumption S≠∅. By WOP let a_0 be a smallest element of S. By the definition of S there exist b_0 and c_0 such that

$$4a_0^3+2b_0^3=c_0^3 \qquad (1.6-2)$$

The left side of this equation is even, so c_0^3 is even, so c_0 is even. Let $c_0=2c_1$. Substitution into (1.6 – 2) yields

$$2a_0^3+b_0^3=4c_1^3 \qquad (1.6-3)$$

or

$$b_0^3=4c_1^3-2a_0^3 \qquad (1.6-4)$$

Hence b_0^3 and therefore b_0 is even. Let $b_0=2b_1$. Substitution into (1.6 – 4) yields

$$4b_1^3=2c_1^3-a_0^3 \qquad (1.6-5)$$

or

$$a_0^3=2c_1^3-4b_1^3 \qquad (1.6-6)$$

Hence a_0^3 and therefore a_0 is even. Let $a_0=2a_1$. It means that

$$a_1>0 \text{ and } a_1<a_0 \qquad (1.6-7)$$

Substitution the value of a_0 into (1.6 – 6) yields

$$4a_1^3=c_1^3-2b_1^3 \qquad (1.6-8)$$

or

$$4a_1^3+2b_1^3=c_1^3 \qquad (1.6-9)$$

So we obtain two solutions of the given equation:

(a_0, b_0, c_0) – equation (1.6-2) and

(a_1, b_1, c_1) – equation (1.6-9)

with condition $a_1<a_0$ (1.6 – 7) which is impossible because a_0 is the smallest element of S. Therefore, our assumption is false and $S=\varnothing$ as claimed. ■

1.7. Use the WOP to prove that for every positive integer n, the sum of the first n odd numbers is n^2, that is,

$$\sum_{k=0}^{n-1}(2k+1) = n^2 \qquad (1.7-1)$$

for all $n>0$

Proof. Assume to the contrary that equation (1.7 – 1) failed for some positive integer n. Let m be the least such number.

(a) Why must there be such an m?

(b) Explain why $m\geq 2$.

(c) Explain why part (b) implies that

$$\sum_{k=1}^{m-1}(2(k-1)+1) = (m-1)^2 \qquad (1.7-2)$$

(d) What term should be added to the left-hand side of (1.7 - 2) so the result equals

$$\sum_{k=1}^{m}(2(k-1)+1)\ ?$$

(e) Conclude that equation (1.7 - 1) holds for all positive integer n. ■

1.8. The Fibonacci numbers F(0); F(1); F(2); … are defined as follows:

$$F(n)=\begin{cases}0, & \text{if } n=0\\ 1, & \text{if } n=1\\ F(n-1)+F(n-2), & \text{if } n>1\end{cases} \qquad (1.8-1)$$

Exactly which sentence(s) in the following bogus proof contain logical errors? Explain.

False Claim. *Every Fibonacci number is even*.

Bogus proof. Let all the variables n, m, k mentioned below be nonnegative integer valued.

1. The proof is by the WOP.
2. Let EF(n) mean that F(n) is even.
3. Let C be the set of counterexamples to the assertion that EF(n) holds for all $n\in\mathbf{N}\cup\{\mathbf{0}\}$, namely,

   $C=\{n\in\mathbf{N}\cup\{\mathbf{0}\} \mid \neg EF(n)\}$
4. We prove by contradiction that C is empty. So, assume that C is not empty.
5. By WOP, there is a least nonnegative integer $m\in C$.
6. Then m>0, since F(0)=0 is an even number.
7. Since m is the minimum counterexample, F(k) is even for all k<m.
8. In particular, F(m-1) and F(m-2) are both even.
9. But by the definition, F(m) equals the sum F(m-1)+F(m-2) of two even numbers, and so it is also even.
10. That is, EF(m)is true.
11. This contradicts the condition in the definition of m that ¬EF(m) holds.
12. This contradiction implies that C must be empty. Hence, F(n) is even for all $n\in\mathbf{N}\cup\{\mathbf{0}\}$.■

Tasks on Recursive Definitions and Structural Induction will be added.

EXERCISES. SET 2. (Supplementary Problems)

Well Ordering Property

2.1. Use WOP to find the smallest element (if any) in the subsets S of **N** defined as: $S=\{n\in N \mid n=x^2-10x+28$ for some $x\in\mathbf{Z}\}$, where **Z** is the set of all integers.

2.3. Use WOP to prove the identity: $2+4+6+\cdots+2n=n(n+1)$

2.4. You are given a series of envelopes, respectively containing 1; 2; 4; …; 2^m dollars. Define

> **Property m**: For any nonnegative integer less than 2^{m+1}, there is a selection of envelopes whose contents add up to exactly that number of dollars.

Use the WOP to prove that **Property m** holds for all nonnegative integers m.

Hint: Consider two cases: first, when the target number of dollars is less than 2^m and second, when the target is at least 2^m.

2.5. Use the WOP to prove that any integer greater than or equal to 8 can be represented as the sum of nonnegative integer multiples of 3 and 5.

2.6. Use the WOP to prove that any integer greater than or equal to 50 can be represented as the sum of nonnegative integer multiples of 7, 11, and 13.

2.7. Use the Well Ordering Principle to prove that

$$n \leq 3^{n/3} \qquad (2.7-1)$$

for every nonnegative integer n.

Hint: Verify (2.7 – 1) for $n \leq 4$ by explicit calculation.

2.8. Use the WOP to prove that

$$2+4+6+\ldots+2n=n(n+1) \qquad (2.8-1)$$

for all n >0

2.9. Prove by the WOP that for all nonnegative integers, n:

$$0^3+1^3+2^3+\ldots+n^3=\left(\frac{n(n+1)}{2}\right)^2 \qquad (2.9-1)$$

Hint: Use the identity $1+2+3+\cdots+n=n(n+1)/2$, if needed

2.10. Use the WOP to prove that

$$1\cdot 2+2\cdot 3+3\cdot 4+\ldots+n\cdot(n+1)=\frac{n(n+1)(2n+1)}{3} \qquad (2.10-1)$$

2.11. Say a number of cents is *makeable* if it is the value of some set of 6 cent and 15 cent stamps. Use the WOP to show that every integer that is a multiple of 3 and greater than or equal to twelve is makeable.

2.12. (a) Prove using the WOP that, using 6¢, 14¢, and 21¢ stamps, it is possible to make any amount of postage over 50¢. To save time, you may specify assume without proof that 50¢, 51¢, . . . 100¢ are all makeable, but you should clearly indicate which of these assumptions your proof depends on.

(b) Show that 49¢ is not makeable

Tasks on Recursive Definitions and Structural Induction will be added.