Recursion and Induction - Supplementary

COMP9020 Tutorial

JIAPENG WANG

24T3 Week8, H18B & F15A

1 Tutorial Outline

1.1 Definition

Ask yourself these questions below to test your understanding.

- What is the basic *Mathematical Induction*?
  - What variations are there?
- How do we *recursively* define a set of objects?
  - What methods do we have to solve a recursive problem?
    - What is *Master Theorem* (in our course)?
    - How is mathematical induction used in recursion?

1.2 Brainstorming

The following questions are open and have no standard answers.

- What are the differences between *deduction* and *induction*?
  - Why do we call it *mathematical induction* instead of *mathematical deduction*?
- What is the relationship between *induction* and *recursion*?
- Why is *logarithm* so important in computer science?
- ......

2 Explanation for Some Exercises

This section is intended to supplement the parts I didn't explain clearly in class.

Exercise 3: Let $\Sigma = \{a, b\}$. Define a set of rules that describe the set of *palindromes* L, words that remain the same after being reversed.

Analysis:

For this type of problem, we can approach it from two perspectives:

- Defining it recursively;
- Using existing set construction mappings to obtain the range.

Approach 1: For a *recursive definition*, we need to find a way to generate new palindromes from existing ones. Given the property of words, we can more concisely describe the recursive process by adding the same letter to both the beginning and end of the old palindromes. Due to this recursive rule, where each addition increases the length by 2, our base case needs to include words of length 0 and 1, i.e.,

(B) $\lambda,\ a,\ b \in L$;

(R) If $w \in L$, then $awa,\ bwb \in L$.

Approach 2: For *defining by mapping*, we need to identify the existing domain. In this problem, we can consider the Kleene star Σ^*. For any given word, we can concatenate it with its reversed version to obtain palindromes of even length. For palindromes of odd length, we simply need to add an a or b in the middle, i.e.,

Define an operation $\mathrm{rev}(w)$, which returns the reverse of the string w.[a] Let $L_{\mathrm{even}}(w)$, $L_{\mathrm{odd}_1}(w)$, and $L_{\mathrm{odd}_2}(w) : \Sigma^* \to \Sigma^*$ be functions defined by

- $L_{\mathrm{even}}(w) := w\,\mathrm{rev}(w)$;
- $L_{\mathrm{odd}_1}(w) := (wa)\,\mathrm{rev}(w)$;
- $L_{\mathrm{odd}_2}(w) := (wb)\,\mathrm{rev}(w)$.

Then we define L as the union of the images of these functions:

$$L := \mathrm{Im}(L_{\mathrm{even}}) \cup \mathrm{Im}(L_{\mathrm{odd}_1}) \cup \mathrm{Im}(L_{\mathrm{odd}_2}).$$

[a] The definition for rev(w) is not unique; for example, it can be given recursively, as in Ex10.

Both definitions are feasible; by comparison, the recursive definition better illustrates the relationships between palindromes, while the mapping-based approach more easily establishes connections with existing sets. Once a particular definition is chosen, the other can be proven as a property.

Exercise 6: Let $n \in \mathbb{N}$. Suppose that we have n lines on a piece of paper where

- The lines go across the entire page
- No lines are parallel
- No point belongs to more than 2 lines

The lines divide the paper into regions. Prove that we can colour these regions with two colours such that adjacent regions (i.e. regions that share an edge) never have the same colour.

Analysis 1:

We note that the page with n lines is generated by adding one new line to the plane with $n - 1$ lines. This makes it straightforward to identify the recursive pattern. Therefore, we can apply mathematical induction on n. For detailed reasoning and solutions, please refer to the solution section.

Analysis 2:

Now, I'd like to discuss whether we can tackle the general case directly and prove this problem using alternative methods:

For this type of coloring problem, if it's possible to cover the regions with two colors, it means that these regions can be divided into two **equivalence classes**. Therefore, if we can devise a classification method with an equivalence relation that yields only two equivalence classes, ensuring that adjacent regions fall into different classes, we can successfully prove this problem.

Therefore, we often consider using assignment operations to transform it into an equivalence class problem involving *numbers*. The most common equivalence relation with only two classes is **parity**—odd and even, i.e.,

- Find an assignment method such that any two adjacent regions have opposite parity in their assigned numbers.

Guided by this idea, we can solve the problem as follows:[1]

[1] The construction methods may vary; any assignment that successfully divides corresponding regions into two classes will work.

Proof:

We begin by establishing two lemmas:

- `Lemma 1`: A single line divides the plane into exactly two regions.
- `Lemma 2`: Any two adjacent regions share exactly one edge; otherwise, since all the lines intersect, lines extending these edges would create an additional region, contradicting the initial setup.

Let n lines $l_1, l_2, \ldots, l_n$ divide the page. We assign an initial value of 0 to all regions, then adjust as follows: for each line l_i, select one side and add 1 to the regions on that side, leaving the other side unchanged.[2]

Now consider two adjacent regions A and B that share only the edge l_1.[3] By construction, A and B lie on opposite sides of l_1 and on the same side of $l_2, l_3, \ldots, l_n$, so their values differ by 1, giving them opposite parity.

Thus, we can color regions with even values one color and those with odd values another, ensuring adjacent regions differ in color. $\square$

[2] Due to `Lemma 1`, this operation is feasible.

[3] Due to `Lemma 2`, all adjacent regions follow this pattern. Since the lines are numbered arbitrarily, we can always assume they intersect at l_1.