Polynomial Restrictor - Project Report

Project Title:

Polynomial Restrictor

Introduction:

The Polynomial Restrictor project is designed to provide an interactive graphical user interface (GUI) for performing polynomial restriction based on partial assignments of variables. The application is built using Python's Tkinter library, providing users with an intuitive interface to input polynomials, specify a field kkk, and apply partial assignments to restrict the polynomial.

Objective:

The primary objective of this project is to create a user-friendly interface that allows users to:

1. Input a polynomial expression.
2. Specify a prime field kkk for modular arithmetic.
3. Input partial variable assignments.
4. Display the resulting restricted polynomial after applying the assignments.

Functionality:

1. Input Handling:

- **Polynomial Input:** Users can enter polynomial expressions in the form of `coefficients` and `variables`. The polynomial is represented as a string (e.g., `1+x1*x3+2*x1^3`).
- **Field kkk Input:** The user can specify the prime field kkk for modular arithmetic. This ensures that all calculations are done modulo kkk.
- **Partial Assignment Input:** Users can input partial assignments for variables in the form of a string (e.g., `x1=1, x2=2`).

2. Polynomial Parsing:

- The polynomial input is parsed to identify the coefficients and variables.
- The `parse_polynomial` function splits the polynomial into terms, separates coefficients and variables, and stores them in a dictionary.

3. Partial Assignment:

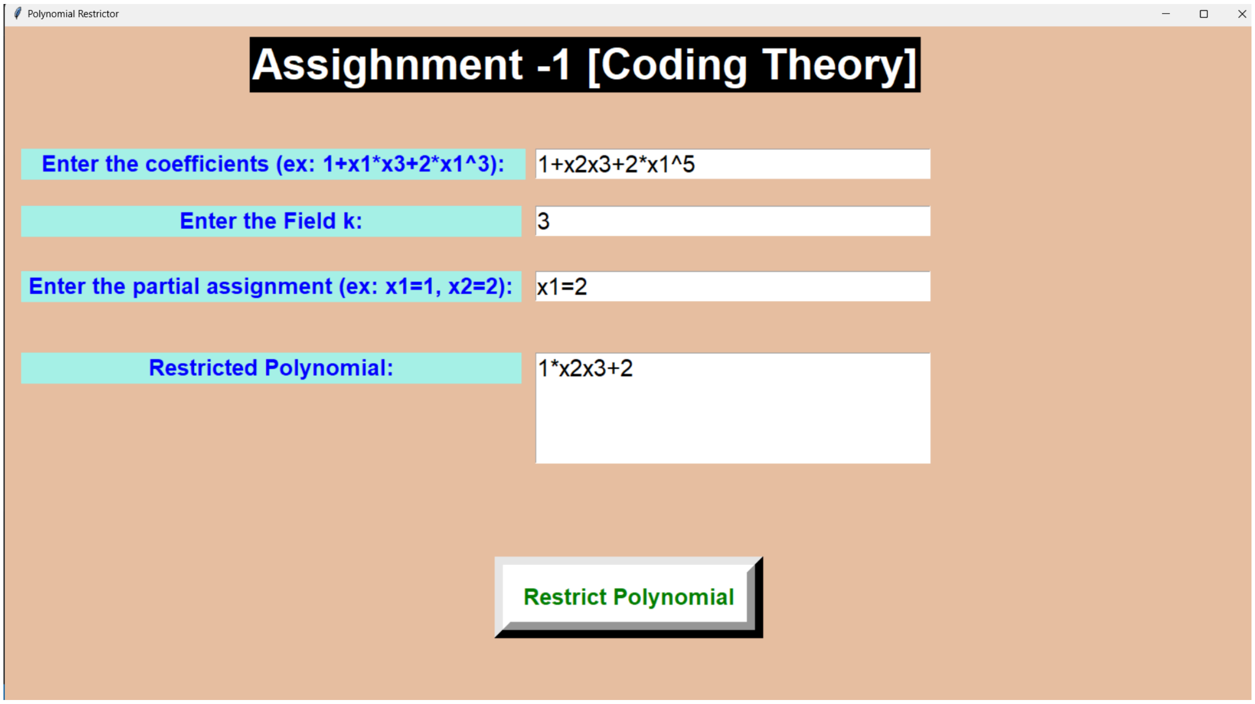
- The partial assignment input is parsed to create a dictionary of variable-value pairs.
- The `partially_polynomial` function applies these assignments to the polynomial, evaluating terms with known variable values and simplifying the polynomial accordingly.

4. Polynomial Restriction:

- The restricted polynomial is computed by substituting the assigned values into the polynomial and simplifying the expression.

5. Output Display:

- The resulting restricted polynomial is displayed in the GUI text box after the `Restrict Polynomial` button is clicked.

- 

Enter the coefficients (ex: 1+x1*x3+2*x1^3): 1+x2x3+2*x1^5

Enter the Field k: 3

Enter the partial assignment (ex: x1=1, x2=2): x1=2

Restricted Polynomial: 1*x2x3+2

Graphical User Interface (GUI) Design:

The GUI is designed using Tkinter, with the following components:

- **Labels:** Used to display instructions and titles.
- **Entry Widgets:** For user input of polynomial, field kkk, and partial assignments.
- **Text Widget:** Displays the restricted polynomial after the computation.
- **Button:** Triggers the computation and displays the result.

Implementation Details:

Classes and Functions:

- **PolynomialRestrictorGUI:** The main class that handles the GUI creation and interaction.
- **print_polynomial(coefficients):** Combines coefficients and variables to form the polynomial string.
- **parse_polynomial(polynomial, k):** Parses the polynomial input into a dictionary of coefficients and variables.
- **parse_variable_input(input_string):** Parses the partial assignment input into a dictionary.
- **partially_polynomial(polynomial, variables_values, k):** Applies partial assignments to the polynomial and computes the restricted polynomial.

User Interaction:

1. **Step 1:** The user enters the polynomial in the specified format.
2. **Step 2:** The user inputs the field kkk.
3. **Step 3:** The user optionally provides partial assignments for variables.
4. **Step 4:** The user clicks the "Restrict Polynomial" button.
5. **Step 5:** The restricted polynomial is displayed in the text box below.

Conclusion:

The Polynomial Restrictor application effectively demonstrates the use of GUI in handling complex polynomial operations. It provides an intuitive interface for users to input data and observe the results in real time. The use of Tkinter ensures that the application is both interactive and easy to use.

This project serves as a practical implementation of polynomial restriction and modular arithmetic concepts, with potential applications in coding theory and algebra.

Future Work:

- **Error Handling:** Implement robust error handling to manage invalid inputs and ensure smooth user experience.
- **Extended Functionality:** Add support for more complex polynomial operations, such as differentiation or integration.

References:

- Python Tkinter Documentation
- Modular Arithmetic and Polynomial Theory