CSE 341 Fall 2024 – Programming Languages Assignment 1

Due on Oct. 27, 2024 at 23:59

In this assignment, you will implement a functional recursive program in Lisp that reads a C file line by line, converts each line into Lisp code, and writes the fully converted code to a new file.

Requirements:

1. **Functional Programming:** No mutable assignments are allowed. Your solution must strictly follow functional programming principles.
2. **Recursive Design:** The primary conversion function must be recursive. Each line of C code should be processed one at a time, and the conversion function should call itself recursively until the end of the file is reached.
3. **Line Types:** Your program must handle a variety of C constructs. The minimum set of C constructs you need to support includes:
    - if statements
    - logical and arithmetical operations
    - for and while loops
    - variable assignments and definitions
    - function and function prototype definitions
    - function calls
    - variable assignment by function return
4. **Functions to Implement:**
    - **line-type**: This function takes a line of C code as input and returns the type of the line as one of the types listed above.
    - **conversion-foo**: This function takes the line type as input and returns the appropriate conversion function for that type.
    - **convert**: This function takes two arguments: the line of C code and the conversion function. It returns the corresponding Lisp code for that line by applying the conversion

function. You can use a counter parameter for line index if it is required in your reading function.

- **Conversion Functions:** You must implement conversion functions for each C line type. These functions will handle the transformation of each line of C code into its Common Lisp equivalent. Example: converting a for loop to a loop construct in Lisp.
- **read_file:** A function that reads the input C file and returns the text content line by line. You can use external libraries for reading the file. The function may be an iterator that returns the next line at each call, or it can return the line by a given index.
- **write_file:** A function that writes the fully converted Lisp code to the output file after all conversions are complete.

5. **Recursive Conversion Process:** You should recursively process each line using the basic recursive structure below:
    - Determine the type of the line using line-type.
    - Select the appropriate conversion function with conversion-foo.
    - Convert the line using the conversion function.
    - If there are more lines, call the recursive function to process the next line and combine the results.

Submission: Submit your Lisp code along with a report containing a brief explanation of how each function works. You can send a message on Teams if you think there is a missing or mistaken part in the homework.