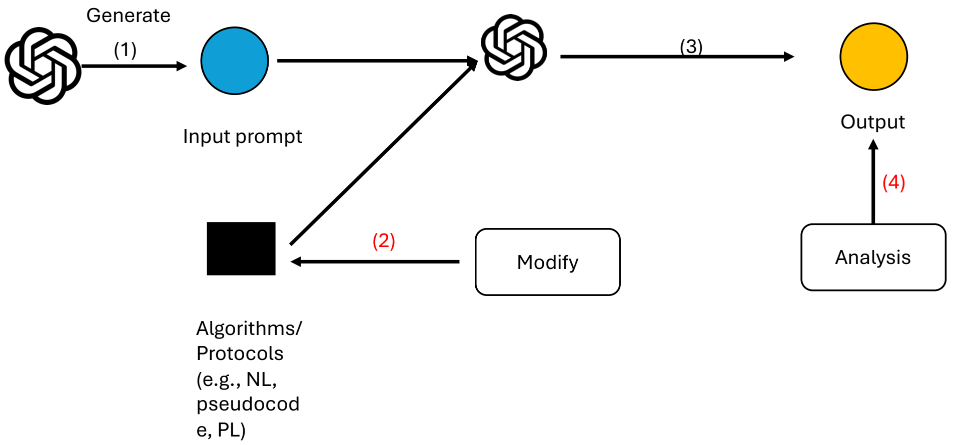
Preparing:

+ LLMs: 3-5 state of the art for code translation LLM, such as GPT4o.
+ Protocols: up to date path finding protocols, some mutual exclusion protocols

Goals: Focus on code translation ability of LLM on several aspects:

+ Different inputs, from natural language/pseudocode/programming to a specification language
+ To answer a question: “will LLM memorize or be able to translate codes”
    + Function names are different to its behaviors
    + Anonymous function names
    + Function behaviors exist some bug (do LLMs fix or only translate based on bugs)
+ To control the validity (the translated code can be run as its behavior inputs), several approaches will be done:
    + Runnable (let compilers or interpreters do)
    + Create test cases by LLMs
    + Human experts checking
    + Run the translated codes in Maude/CafeOBJ to model check
+ For others:
    + Cost estimate
    + Time estimate



Steps:

1) (TODO) Create a prompt based on LLM suggestion: “Suppose you are an expert LLM prompter based on specific LLM, give me a prompt for LLMs that can: Translate natural language/pseudocode/programming language to a specification language. Keep the translation only no need to correct when it is false (optional)
    a. (TODO) Run it to LLM and get the best prompt (survey on several LLMs to keep one or take prompts for each prompt.
    b. (TODO) Select algorithms/protocols as inputs for task 2)
2) (TODO) Use the prompt from 1a) and algorithms/protocols from 1b) as two inputs for LLM, where algorithms/protocols are modified with several scenarios
    a. (TODO) Function names are different to its behaviors
    b. (TODO) Anonymous function names
    c. (TODO) Function behaviors modification, such as adding some errors (to show that “do LLMs fix errors or translate such errors – conform with translation requirements”)
3) Run 2) with several LLMs and get outputs
4) Analyze outputs based on several criteria:
    a. (TODO) Runable (can the output code be run on compiler/interpreter?)
    b. (TODO) Test cases (generated by LLMs)
    c. Human experts analyzing
    d. Conduct model checking