Group Meeting (10/10/2023)

Subgroup 1 (Maximo & Shahir & Fox): For the most part our team started to develop a demo API using .NET core 7 and MySQL as the database engine. The API runs locally and we're testing with user handling, getting records of all users, updating data of a user, and other basic functions for an API.

Subgroup 2 (John & Ahnaf & Pranav & Youssif)

We are implementing the pipeline described in the last group meeting. Our subgroup divided into two parts to address **Step 1** and **Step 2**.

Step 1. Construction of the AST worked by @Chu, John and @Rahman, Shaikh Ibrahim.

Overview

- **Software:** Ubuntu 20.04.6 (WSL)
  - Trying to resemble the 84 Lumber corporate software setting.
  - Working on everything with the command line interface.
- **Pipeline**

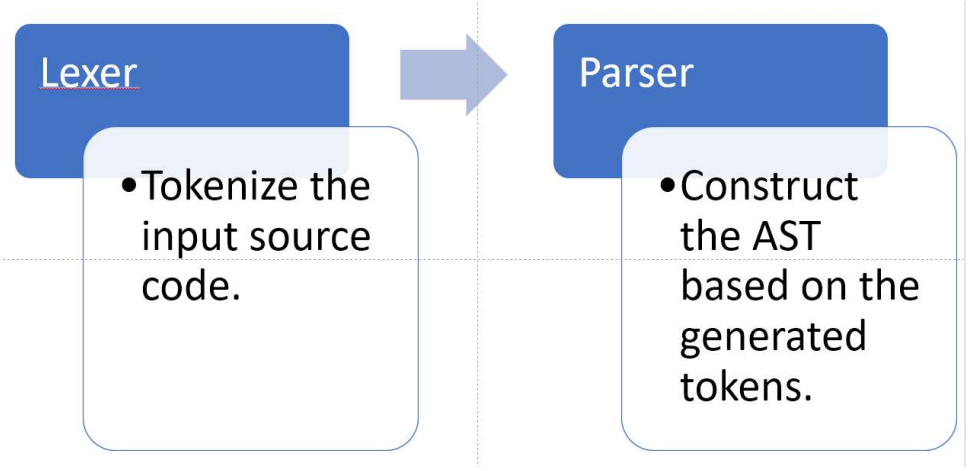


Figure 1. Overview of Step 1.

Progress

- Currently, we are practicing defining a grammar that encompasses the framework of C and BASIC codes. We have been learning the syntax, interpreters, and compilers required to construct the AST. This acquisition of those concepts is for designing blueprints of defining regular expressions for creating grammars that encompass C & BASIC framework. So far, we implemented one very trivial example of constructing the AST that adds positive numbers. Please refer to the figures described on the next page.

Future Plan: Strive to find a regular expression that will be defined in Expr.g that interprets both C & BASIC framework.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/jc981073/.hushlogin file.
jc981073@DESKTOP-R3H5HUN:~$ cd
.cache/   .local/   .ssh/     84Lumber/
jc981073@DESKTOP-R3H5HUN:~$ cd 84Lumber/
jc981073@DESKTOP-R3H5HUN:~/84Lumber$ ls
scratch
jc981073@DESKTOP-R3H5HUN:~/84Lumber$ cd scratch/
jc981073@DESKTOP-R3H5HUN:~/84Lumber/scratch$ ls
__pycache__  ex01
jc981073@DESKTOP-R3H5HUN:~/84Lumber/scratch$ cd ex01
jc981073@DESKTOP-R3H5HUN:~/84Lumber/scratch/ex01$ ls
Expr.g       Expr.tokens       ExprLexer.py      ExprListener.py  __pycache__
Expr.interp  ExprLexer.interp  ExprLexer.tokens  ExprParser.py    test.py
jc981073@DESKTOP-R3H5HUN:~/84Lumber/scratch/ex01$ python3 test.py
? 3+3
(root (expr (expr 3) + (expr 3)) <EOF>)
```

Figure 2. Trivial AST construction example

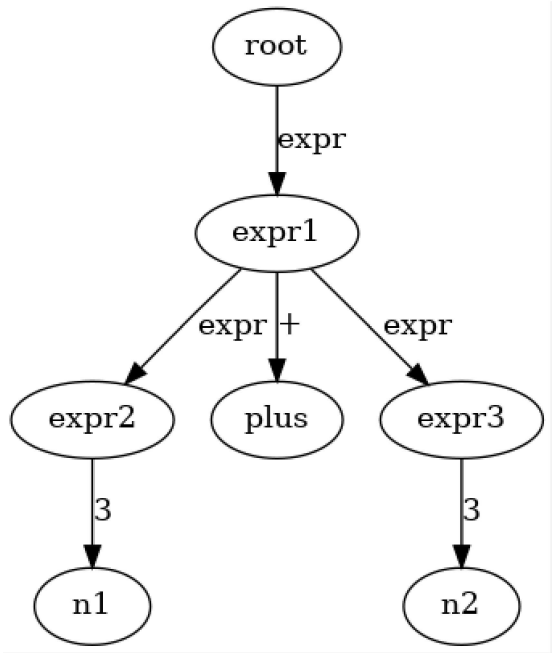


Figure 3. Visualized AST using graphviz.

Step 2. Generate comments on each function and variable based on the AST above by @Mishra, Pranav @Goda, Youssif.

For this task, we are adopting a “code-and-test" approach. The first function (GraphifyI) is under implementation right now. At the same time, we have created a sample code file as well as sample abstract syntax tree to test the function upon its creation:

```
def GraphifyI(file):
    '''Function that takes in an AST File and converts it to a Graph'''
    lines = file.readLines()
    for line in lines:
        numLeadingWhitespaces = len(line) - len(line.lstrip())
        level = numLeadingWhitespaces/4 + 1
        tempList = line.split(',')
        lineNum = tempList[-1]
        content = line[numLeadingWhitespaces+1:len(line)-2]
        #Create an edge object
        #add edge object to graph
```

```java
package lumberTry;

public class SchoolClass {

	int studentNumber;
	int classRoom;
	String teacherName;

	public SchoolClass(int studentNumber, int classRoom, String teacherName) {
		this.classRoom = classRoom;
		this.studentNumber = studentNumber;
		this.teacherName = teacherName;
	}

	public void addStudent() {
		studentNumber ++;
	}

	public void classRoomNum(int classRoomNumber) {
		classRoom = classRoomNumber;
	}

	public static void main(String args[]) {
		SchoolClass CS = new SchoolClass(20,118,"Goble William");
		CS.addStudent();

	}

}
```

```
PackageName ,1
ClassDeclaration ,3
        StudentNumberDeclaration ,5
        ClassRoomDeclaration ,6
        TeacherNameDeclaration ,7
        Constructor ,9
                classRoomInitial ,10
                studentNumberInitial ,11
                teacherNameInitial ,12
        methodAddStudent ,15
                AddStatemwnt ,16
        methodClassRoomNum ,19
                AddStatemwnt ,20
        methodMain ,23
                NewSchoolClass ,24
                addStudentCall ,25
```

Future work: Complete this method and define the specific graph data structure used to implement it and future methods.

Subgroup 3 (Boosung & William)

Group Goals

- Propose a GitHub Copilot pipeline for the company to use to modernize code directly
- Propose a System Architecture redesign to partition code and code modernization

Motivation

A GitHub copilot would be a robust method to modernize the codebase without looking into very specific solutions. Copilot is the standard used in the industry, being used by over 1M developers and over 20,000 organizations.

Code translation will differ according to different sections of the system: frontend may translate to Javascript/PHP, backend may translate C to Python, and data migration softwares may be used for transferring over to data warehouses. By coming up with a proposed system architecture, we will be able to come up with specific code translation pipelines per section.

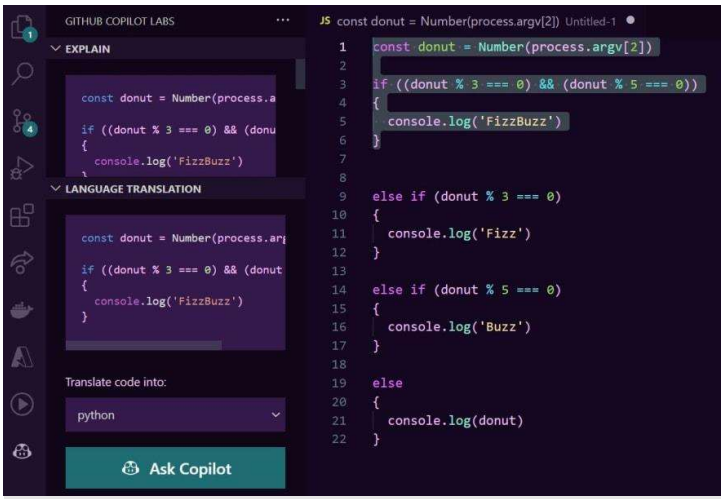
Task 1:Translation Pipeline

Option 1: Copilot Pipeline

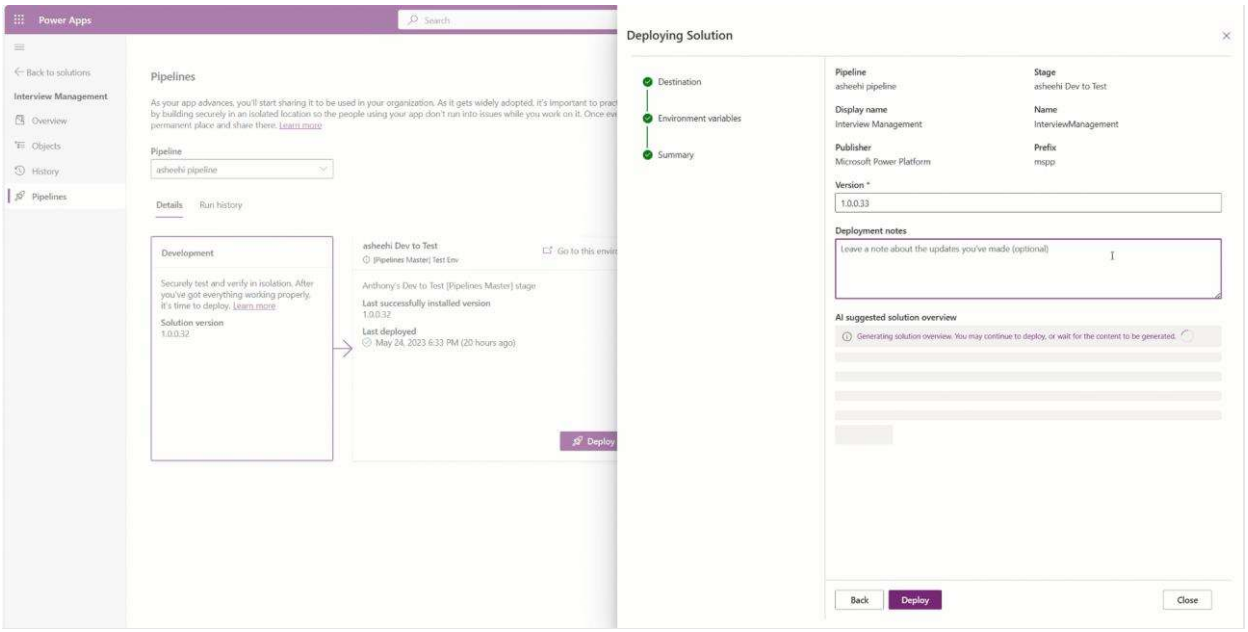
Used by other enterprises, like 3M, Prada Group, Kraft Heinz, and Microsoft.

Automatic translation

Use Copilot Labs



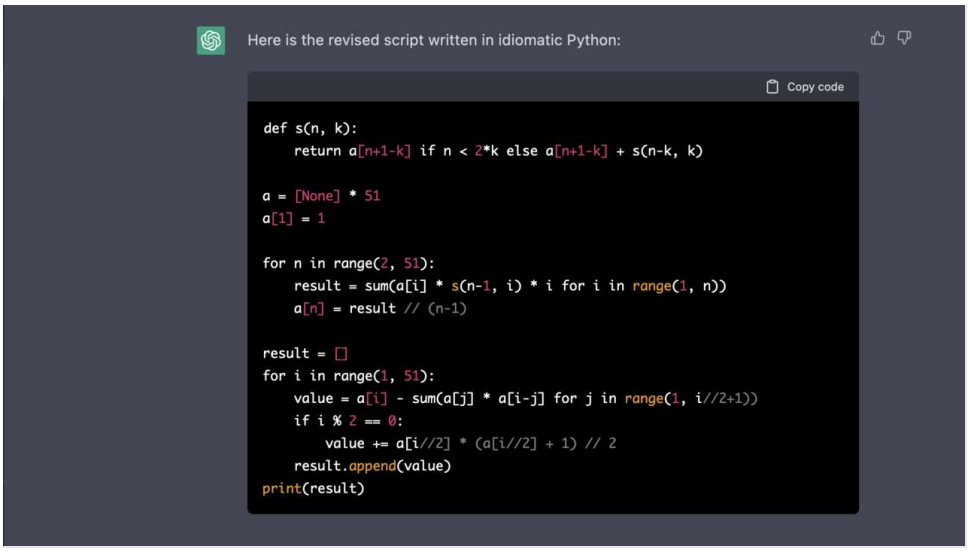
Integration with Power Platform



Code Security

Built in code security functionality for enterprises.

Option 2: ChatGPT



Can perform translation but needs human guidance

Potential Pipeline as to what to add with the code to help increase the correctness of GPT's translation.

Outperforms copilot given specific guidance, but copilot is more robust

Option 3: Transpillers

https://github.com/DAN-329/C_to_Python_translator

c-program.txt - Notepad

File Edit Format View Help

```
#include<stdio.h>
#include<string.h>
#include<math.h>
void work(int i)
{
        while(i<10)
        {
                printf("The value of i is %d\n",i);
                i++;
        }
}
void work2(int num)
{
        int sum;
        sum=0;
        for(int k=0;k<num;k++)
        {
                sum+=k;
        }
        printf("The sum is %d\n",sum);
}
int main()
{
        int i,a,b,num;
        char name[10],ch;
        int arr1[10];
        int arr2[]={1,2,3,4,5,6,7,8,9};
        printf("Enter your name\n");
        scanf("%s",&name);
        printf("Enter a or b or c\n");
        scanf("%c",&ch);
        switch(ch)
        {
                case 'a':
                        printf("The letter is %c\n",ch);
                        break;
                case 'b':
                        printf("The letter is %c\n",ch);
                        break;
                case 'c':
                        printf("The letter is %c\n",ch);
                        break;
                default:
                        printf("Invalid input\n");
                        break;
        }
        printf("Enter the value of i\n");
        work(i);
        printf("Enter values of a and b\n");
        scanf("%d",&a);
        scanf("%d",&b);
        if(a>b)
        {
                printf("a is bigger\n");
        }
        else
        {
                printf("b is bigger\n");
        }
        printf("Enter the value of j\n");
        scanf("%d",&j);
        do
        {
                printf("The value of j is %d\n",j);
                j++;
        }while(j<10);
        printf("Enter a number\n");
        scanf("%d",&num);
        work2(num);
        return 0;
}
```

Input file in C

py-program.txt - Notepad

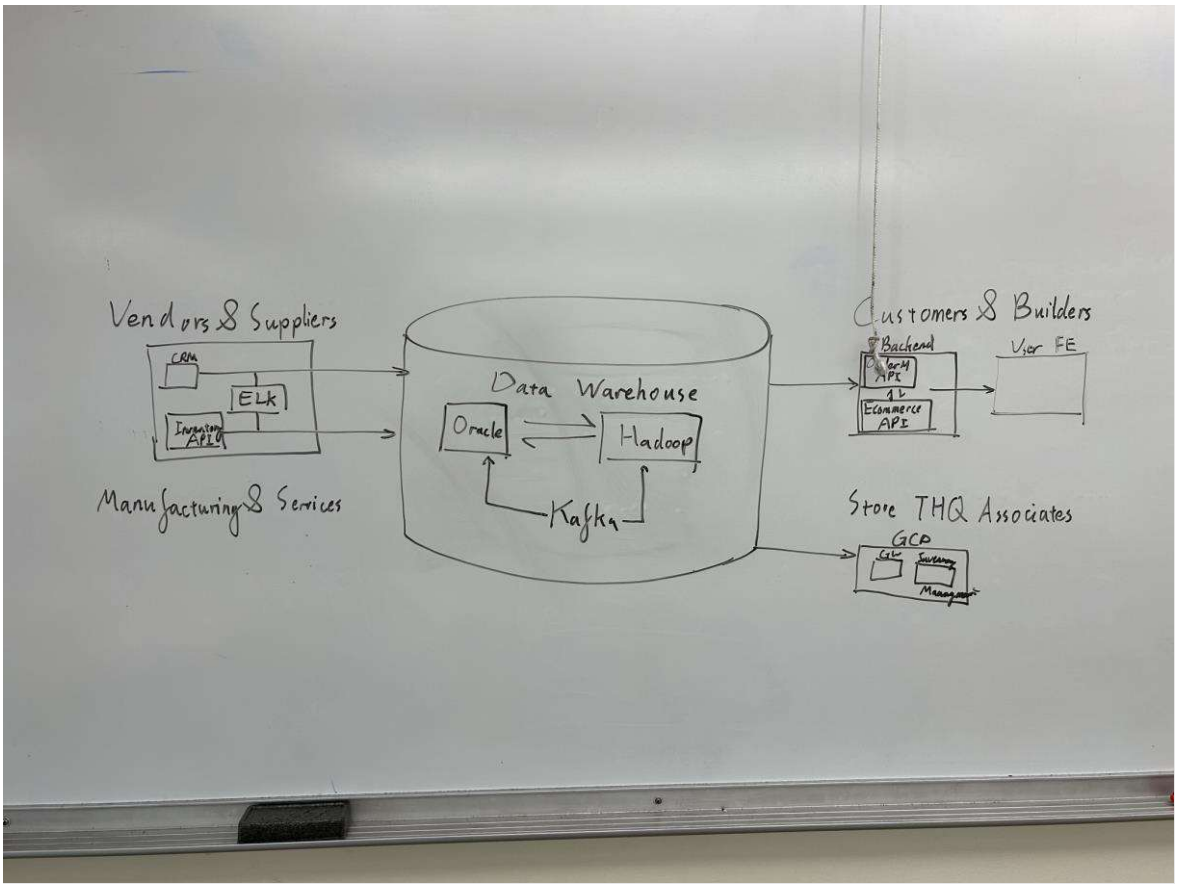
File Edit Format View Help

```
import string
import math
def work(i):
        while(i<10)
                print("The value of i is {}\n".format(i))
                i+=1
def work2(num):
        sum=0
        for i in range(0,num):
                sum+=k
        print("The sum is {}\n".format(sum))
arr2=[1,2,3,4,5,6,7,8,9]
print("Enter your name\n")
name=input()
print("Enter a or b or c\n")
ch=input()
if ch== 'a':
        print("The letter is {}\n".format(ch))
elif ch== 'b':
        print("The letter is {}\n".format(ch))
elif ch== 'c':
        print("The letter is {}\n".format(ch))
else:
        print("Invalid input\n")
print("Enter the value of i\n")
work(i)
print("Enter values of a and b\n")
a=int(input())
b=int(input())
if(a>b):
        print("a is bigger\n")
else:
        print("b is bigger\n")
print("Enter the value of j\n")
j=int(input())
while True:
        print("The value of j is {}\n".format(j))
        j+=1
        if j>=10 : break
print("Enter a number\n")
num=int(input())
work2(num)
```

Output file in python

	Copilot	ChatGPT	Transpillers
Userbase	Enterprises & Developers	Developers	Students
Pricing	$19 (enterprise)	Free (v3.0)	Free
Licensing	Enterprise	GPL	GNU/GPL
Notes	- Biggest database of publicly available	- State of the art LLM - Can perform	- Free

Task 2: System Architecture Design



Timeline

1. Research existing GitHub Copilot pipelines/services from industry
2. Propose a similar Copilot pipeline for 84 Lumber codebase (or pipelines using other tools)
3. Consider pricing (enterprise), data privacy, speed, robustness
4. Come up with system architecture

Questions for 84 Lumber (@Kim, Boosung**,** @Cheng, William **)**

- What does the budget look like for the translation pipeline?
- What pipeline most appeals to you?
- Can we get a better idea of what the overarching architecture currently looks like/will look like?