Compilers [Fall 2016]

Programming Assignment I

Objectives

1. To understand the definitions of DJ and DISM, which will serve as source and target languages for a compiler built in future assignments.
2. To implement small DJ and DISM programs.
3. To become familiar with and able to use a DISM simulator.

Due Date: Sunday, September 4, 2016 (at 11:59pm).

Machine Details

Complete this assignment by yourself on the following CSEE network computers: c4lab01, c4lab02, ..., c4lab20. These machines are physically located in ENB 220. You can connect to the C4 machines from home using SSH. (Example: Host name: *c4lab01.csee.usf.edu* Login ID and Password: <your NetID username and password>) You are responsible for ensuring that your programs compile and execute properly on these machines.

Assignment Description

For this assignment, you will acquaint yourself with the DJ and DISM languages by implementing one small program in each language. You will write a DJ program in a file called *sub.dj* and a DISM program in a file called *sub.dism*.

These programs input a sequence of *words*. A word is a string of characters, where each character is encoded as a number and each word is terminated by the special character 0. The programs immediately halt after an empty word is entered (empty words just contain the character 0). If not halted, the programs output, for every word *w* entered after the first word *f*, two numbers: the first being a 1 or 0 to indicate whether *w* is a *subsequence* of *f*, and the second being a 1 or 0 to indicate whether *w* is a *substring* of *f*.

An example of this desired functionality appears below. The example is a trace of running the program in *sim-dism*, so it ends with a statement indicating that the simulation has completed. In this example, *f* (the first word) is "1234".

```
Enter a natural number: 1
Enter a natural number: 2
Enter a natural number: 3
Enter a natural number: 4
Enter a natural number: 0
Enter a natural number: 1
Enter a natural number: 4
Enter a natural number: 0
1
0
Enter a natural number: 2
Enter a natural number: 3
Enter a natural number: 0
1
1
```

```
Enter a natural number: 3
Enter a natural number: 2
Enter a natural number: 0
0
0
Enter a natural number: 0
Simulation completed with code 0 at PC=52.
```

Hints

Whenever a DJ or DISM program attempts to read a natural number, the prompts of "Enter a natural number: " get printed automatically. Hence, you don't need to worry about outputting those prompts. DJ and DISM programs can only input and output natural numbers (using the *readNat* and *printNat* calls in DJ, and the *rdn* and *ptn* instructions in DISM).

To give you a rough idea of the effort involved: It took me about 2 hours to do this assignment (but I started already familiar with DJ and DISM). My DJ implementation is 46 lines of code, and my DISM implementation is 53 lines of code (excluding whitespace/comments). If you find yourself spending significantly more time (like over 12 hours), please consider asking the TA for help with whatever is slowing you down.

Testing Your DISM Program

Please use the DISM simulator, sim-dism, to test your DISM program. When your DISM program halts, it may halt with any code.

Testing Your DJ Program

Because you're writing a program in a new language for which no compiler yet exists, you can't test your program by executing it! This situation is unpleasant but realistic. You'll have to ensure by hand that your DJ program is valid and would behave correctly if executed. You could, however, modify your DJ program into a valid Java program (using Java's *Scanner* class to mimic *readNat*), and then test that Java program.

Formatting, Grading, and Submission Notes

- To make it easier for our TA to read and evaluate your code, use spaces rather than tabs in your code and avoid long lines of code (I try to limit lines to 80 characters).
- Your programs will be graded on both correctness and style, so include good comments, well-chosen variable names, etc. For full credit, your code must not be significantly more complicated than necessary.
- The TA will test submissions on inputs not shown in the example above.
- Type the following pledge as an initial comment in every file you submit for this course: "I pledge my Honor that I have not cheated, and will not cheat, on this assignment." Type your name after the pledge. Not including this pledge in a submitted file will lower your assignment grade 50%.
- Upload and submit both of your files for Assignment 1 in Canvas.
- You may submit your assignment in Canvas as many times as you like; we will grade your latest submission.
- For every day that your assignment is late (up to 3 days), your grade reduces 10%.