CPSC-402 Report
Compiler Construction

Aidan Wall
Chapman University

February 28, 2022

Abstract

Short summary of purpose and content.

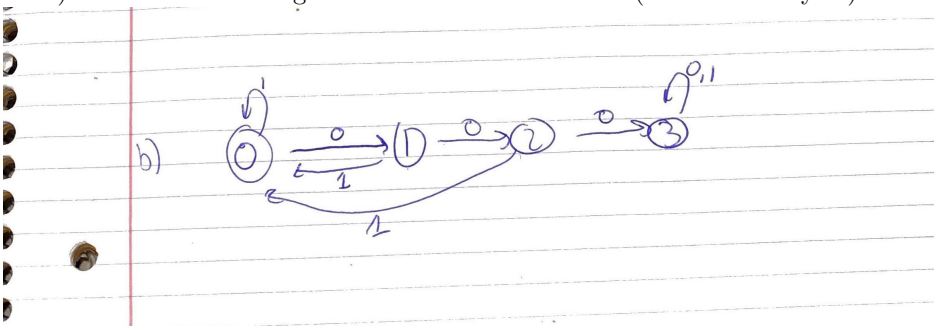
Contents

1 Homework

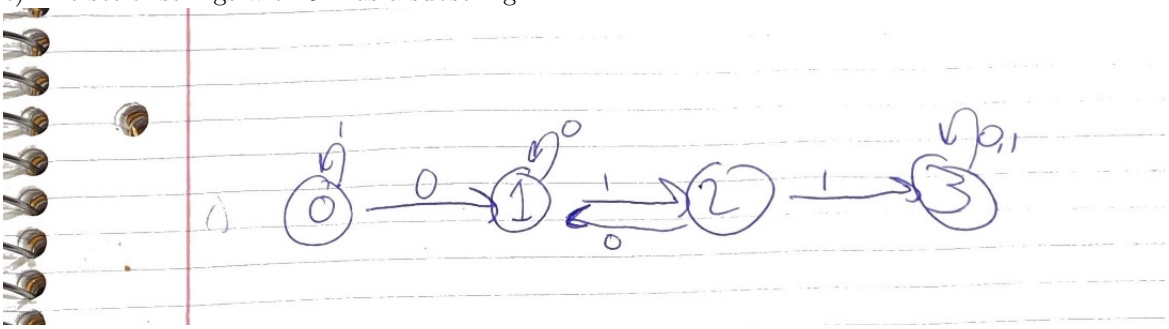
For most weeks, you will have a subsection that contains your answers.

1.1 Week 1

Exercise 2.2.4 Give DFA's accepting the following language over the alphabet 0,1:
b) The set of all strings with three consecutive 0's (not necessarily at)



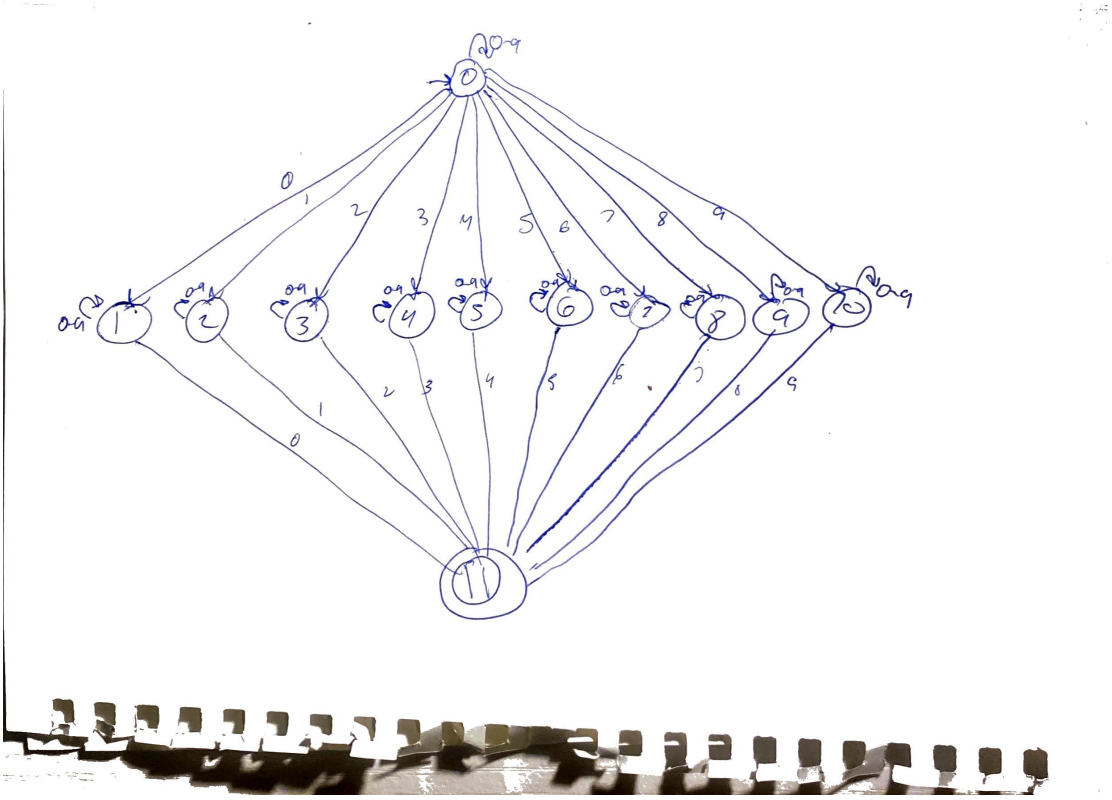
c) The set of strings with 011 as a substring.



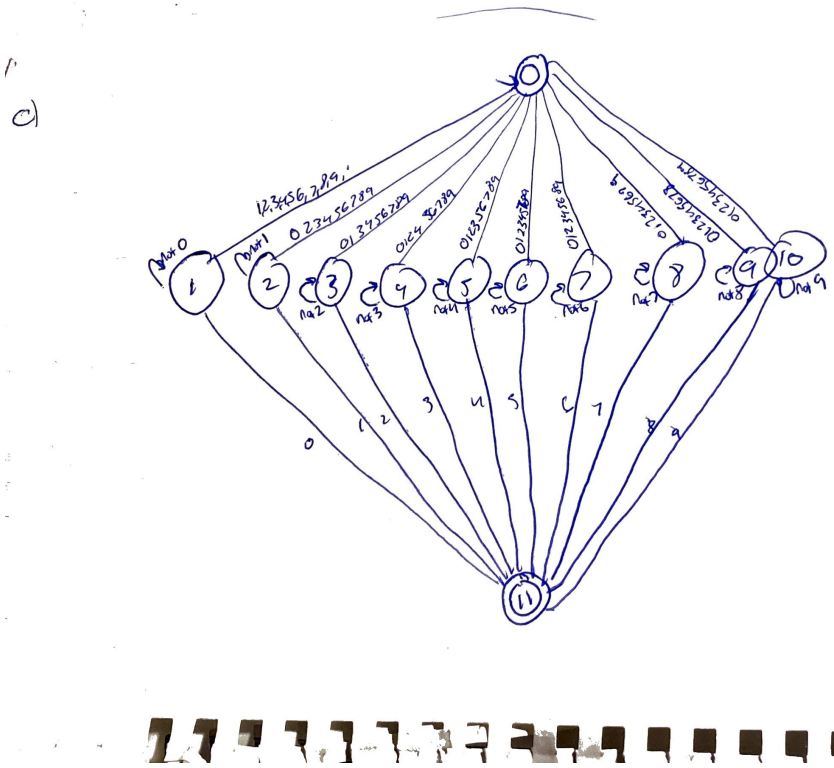
1.2 Week 2

Exercise 2.3.4 Find NFA's that recognize:

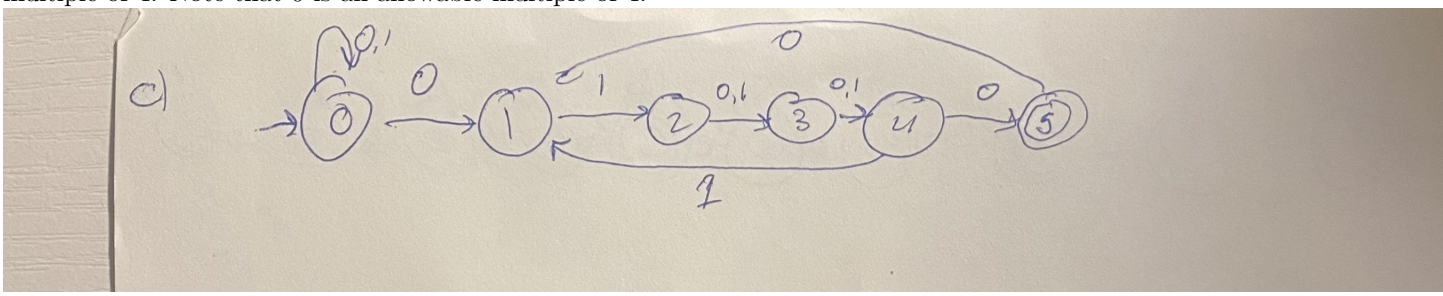
a) The set of strings over alphabet 0,1,...,9 such that the final digit has appeared:



b) The set of strings over alphabet 0,1,...,9 such that the final digit has not appeared before.

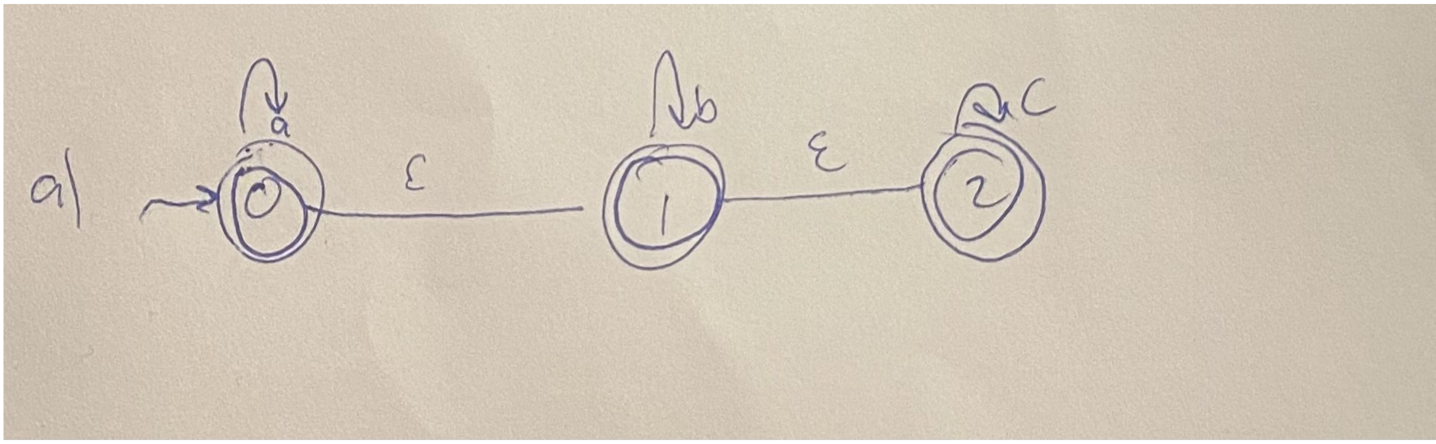


c) The set of strings of 0's and 1's such that there are two 0's separated by a number of positions that is a multiple of 4. Note that 0 is an allowable multiple of 4.

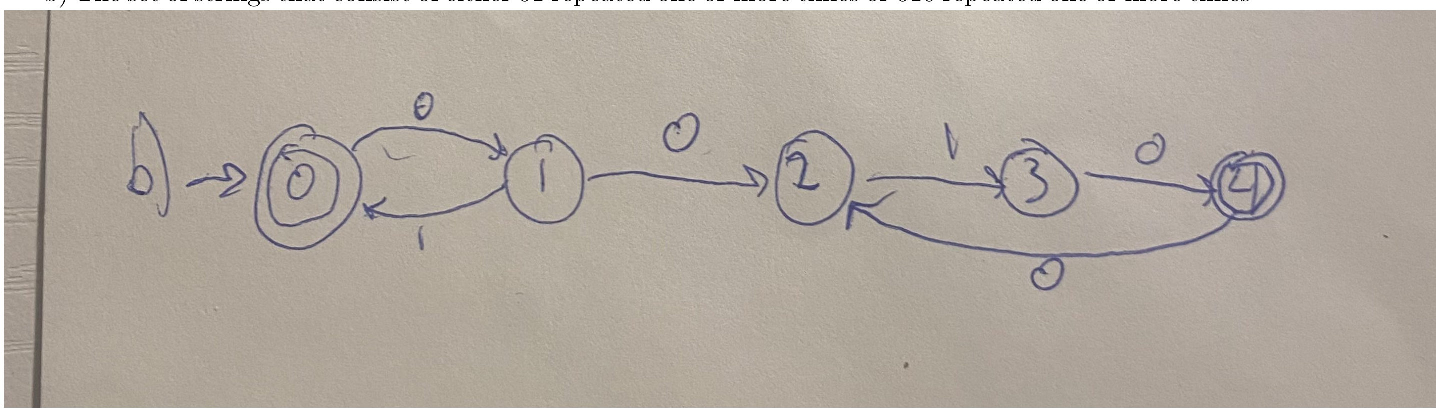


Exercise 2.5.3 Find NFA's that recognize

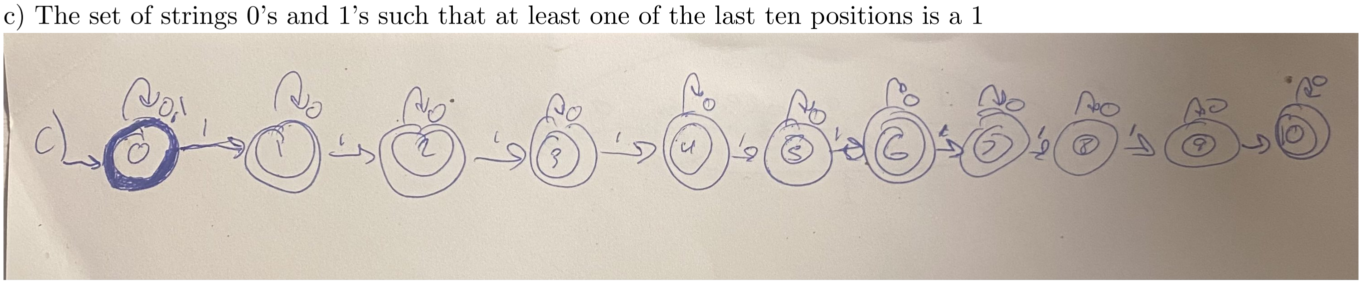
a) The set of strings consisting of zero or more a's followed by zero or more b's, followed by zero or more c's



b) The set of strings that consist of either 01 repeated one or more times or 010 repeated one or more times



c) The set of strings 0's and 1's such that at least one of the last ten positions is a 1

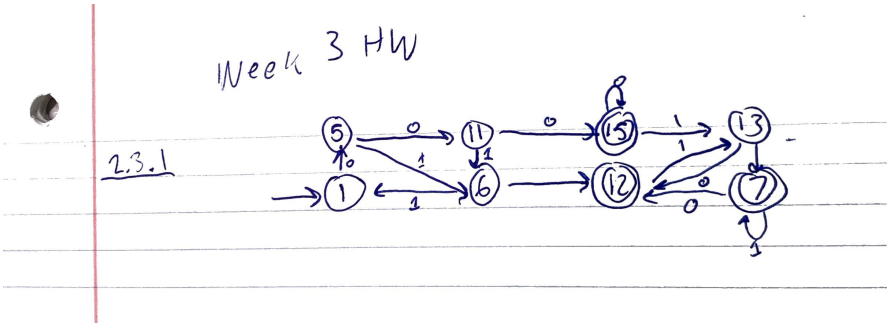


1.3 Week 3

Exercise 2.3.1 Convert to a DFA the following NFA:

	0	1
$\to p$	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	$\emptyset$
$*s$	$\{s\}$	$\{s\}$

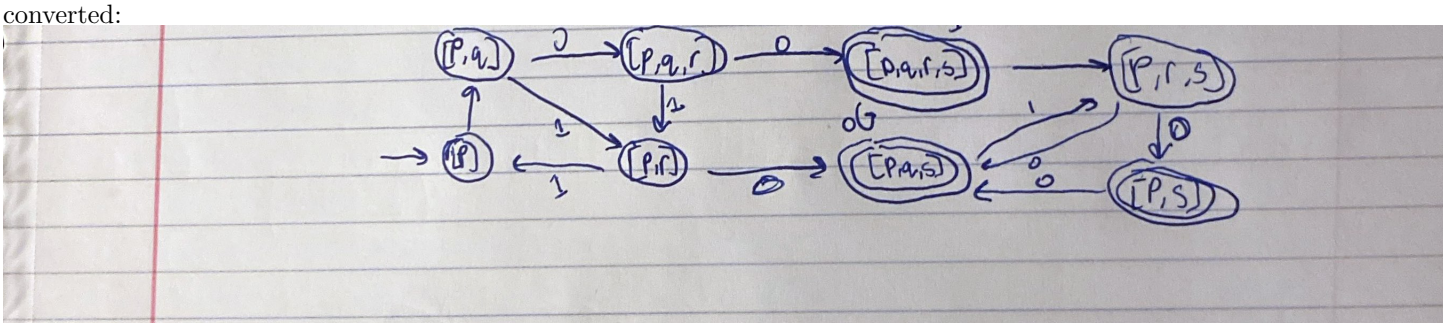
converted:



Exercise 2.3.2 Convert to a DFA the following NFA:

	0	1
$\rightarrow p$	$\{q, s\}$	$\{q\}$
$*q$	$\{r\}$	$\{q, r\}$
r	$\{s\}$	$\{p\}$
$*s$	$\emptyset$	$\{p\}$

converted:



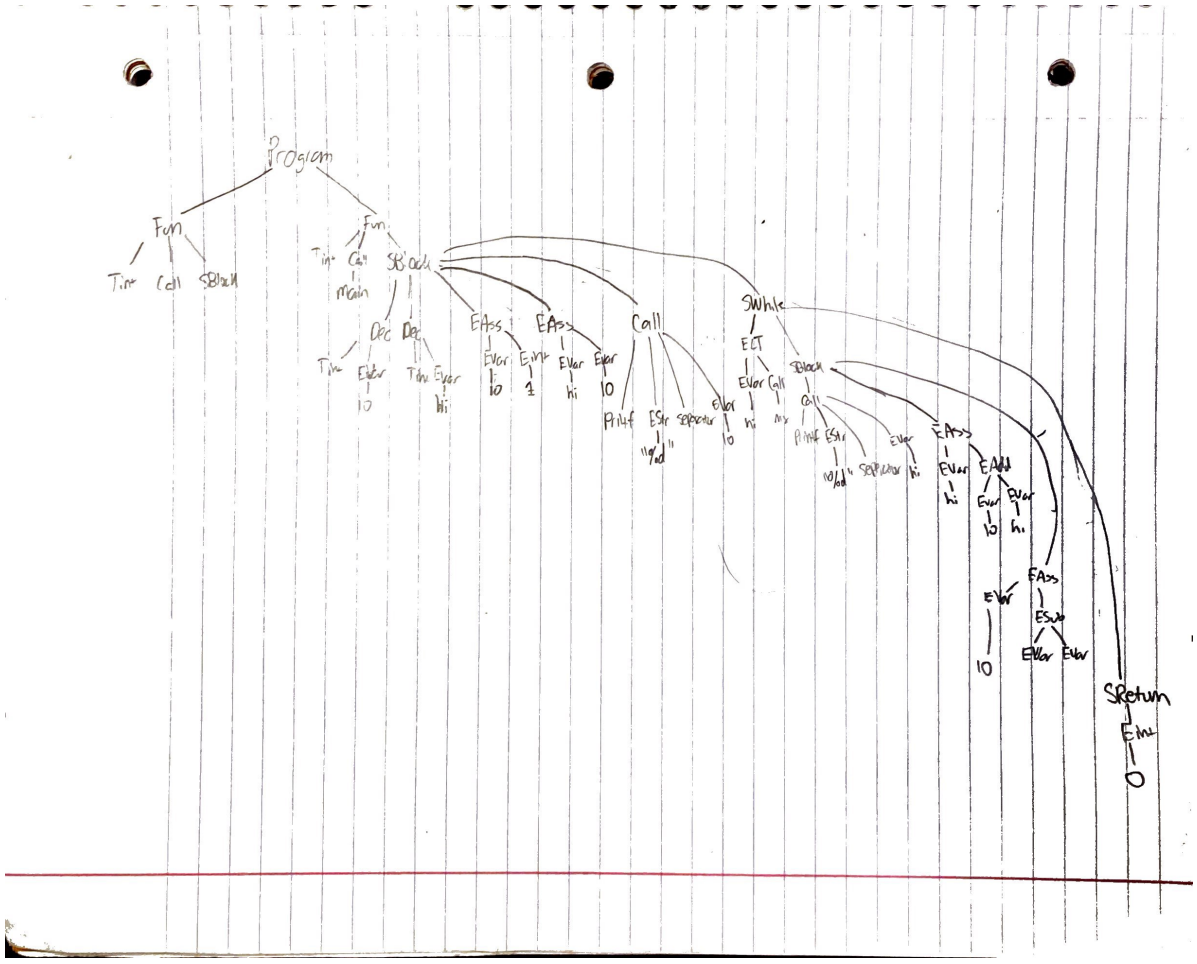
Converting NFAs to DFAs In Haskell Using the List Monad:

```
-- convert an NFA to a DFA
nfa2dfa :: NFA s -> DFA [s]
nfa2dfa nfa = DFA {
  -- exercise: correct the next three definitions
  dfa_initial = [nfa_initial nfa],
dfa_final = let final qs = disjunction(map(nfa_final nfa) qs) in final,
dfa_delta = let
    delta [] c  =[]
    delta (q:qs) c = concat [nfa_delta nfa q c,delta qs c]
    in delta}
```

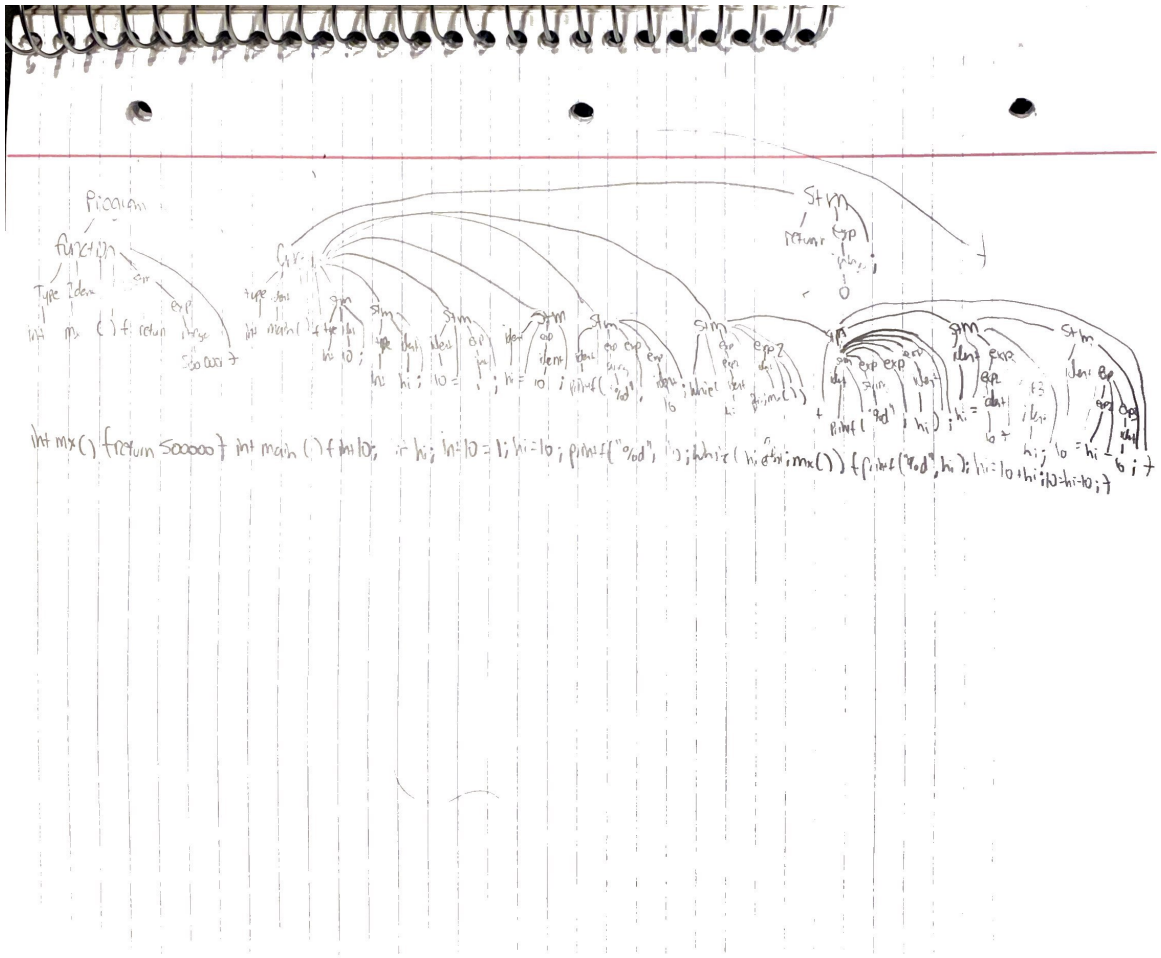
Explanation: The goal is to write an algorithm that will convert an NFA to a DFA. Because the states of the DFA are sets of states of the NFA, we will need to do this many times, which we will do iteratively in dfa_final. dfa_initial has the initial state of the sets, which will contain the initial state of the NFA as list as the only element. dfa_final will have will only be final when it is a final state of the NFA, which we can get from the code already provided. dfa_delta uses the Haskell concat function to concatenate all of the sets together.

1.4 Week 4

Write out the abstract syntax tree for the complete fibonacci program



Write out the parse tree (=concrete syntax tree) for the complete fibonacci program:



2 Project

The project will be to compile code from a programming language of your choice to assembly and to explain the assembly code and compilation process using the knowledge your learned in this course.

Pro tips:

- Choose a compiled (not interpreted) programming language.
- Choose a language that you find interesting anyway.
- Start early.
- Come to office hours. I have not run this part of the course before and I am really interested what you will find.

3 Conclusions

(approx 400 words)

In the conclusion, I want a critical reflection on the content of the course. Step back from the technical details. How does the course fit into the wider world of programming languages and software engineering?

References

[HMU] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: Introduction to automata theory, languages, and computation, 3rd Edition. Pearson international edition, Addison-Wesley 2007