Assignment 3

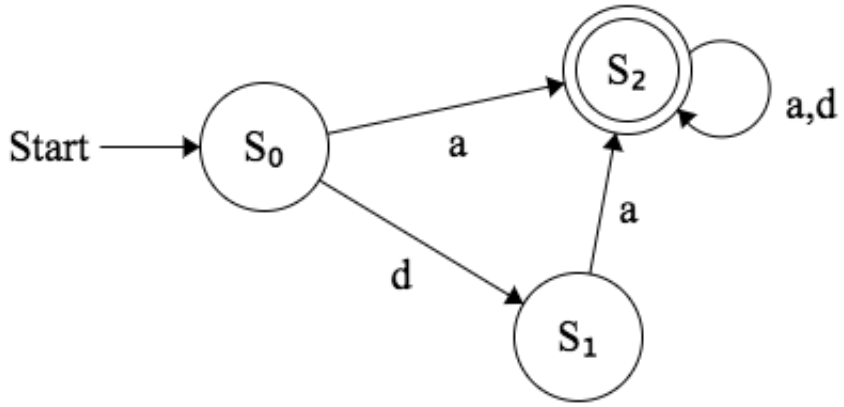
3.1 Rewrite the productions for each of the following nonterminals as right regular grammars: Identifier, Float.

```
Identifier -> Letter | Letter Integer | Letter Identifier
Float      -> . Integer | Integer . Integer
```

3.2 Rewrite the productions for each of the following nonterminals as left regular grammars: Indentifier, Float.

```
Identifier -> Letter Identifier | Letter Integer | Letter
Float      -> Integer . Integer | . Integer
```

3.3 Draw a DFSA for identifiers that contain only letters and digits, where the identifier must have at least one letter, but it need not be the first character. Hint: everything to the left of the leftmost letter must be a digit.



Σ = {a, d} where a = any letter and d = any digit

3.4 Try to define the language {a^n b^n } using a regular grammar. Discuss why this might

not be possible.

From Assignment 2:

If L is a regular language then the pumping lemma must hold true:

```
Pumping Lemma:

There exists an integer p ≥ 1 depending only on L such that every string w in L
of length at least p (p is called the "pumping length") can be written as w =
xyz (i.e., w can be divided into three substrings), satisfying the following
conditions:

|y| ≥ 1,
|xy| ≤ p,
for all i ≥ 0, x(y^i)z ∈ L
```

- Claim: $a^n b^n$ is not regular
- Let p be the number from the pumping lemma
- Let w = $a^p b^p$
- Since w ∈ L and |w| >= p the pumping lemma applies:
    - w = xyz where y != {} and |xy| <= p
- Note: the xy part must occure in the first p characters:
    - Therefore y must occur in the firt p characters
    - y = a^k where 0 < k <= p
    - x = a^q wher 0 <= q < p
    - z = $a^{p-k-q} b^p$ (what's leftover)
- Per the pumping lemma: xyyz ∈ L:
    - xyyz = $a^q a^k a^k a^{p-k-q} b^p$
    - xyyz = $a^{q+k+k+p-k-q} b^p$
    - xyyz = $a^{k+p} b^p$
- $a^{k+p} b^p$ is not an element of L (contradiction)
- L is not regular

3.10 For floating point numbers in scientific notation, give:

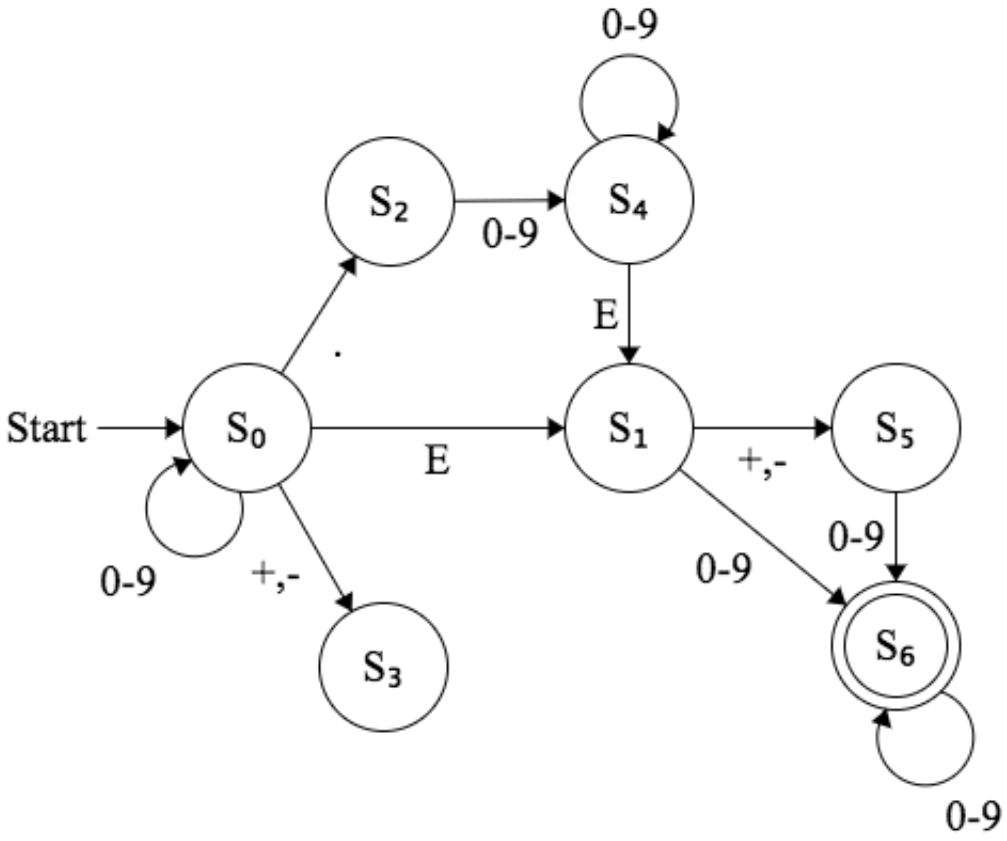
(a) a right regular grammar

```
Float -> Integer Float | . Integer | Exp
Exp   -> E + Power | E - Power | E Power
Power -> Integer Power | Integer
```

(b) a regular expression

```
(0-9)* (.(0-9))* E+ (+|-)* (0-9)+
```

(c) a DFSA. Give examples of numbers that are legal and illegal.



Legal: 3.14E1, .3E3, 2E-2

Illegal: 3.14, +3.14E1, 2E

3.16 Consider the following grammar:

```
S -> | a |(T)
T -> T,S | S
```

After augmenting the grammar:

(a) Draw the left dependency graph.

(b) Compute First for each nontenninal