ECE 553: Compilers Spring 2018
Midterm

Name: NetID:

There are 8 questions, with the point values as shown below. You have 75 minutes with a total of 75 points. Pace yourself accordingly.

This exam must be individual work. You may not collaborate with your fellow students. However, this exam is open notes, so you may use your class notes.

I certify that the work shown on this exam is my own work, and that I have neither given nor received improper assistance of any form in the completion of this work.

Signature:

#	**Question**	**Points Earned**	**Points Possible**
1	SML		8
2	Regular Languages		10
3	NFA to DFA		8
4	NFA to Regexp		10
5	Context Free Grammars		10
6	LR Parsing		10
7	Types		12
8	Type Inference		7
	Total		75
	Percent		100

Question 1 SML [8 pts]

1. Write `foldl` using `foldr` and `rev`.

2. Write tail recursive factorial in SML (`fact: int` → `int`). You may write any helper function you want, but they must be placed in a `let` inside of the `fact` function.

3. Suppose I had the following:

```
functor X(K: ORD_KEY) =
struct
 structure S = SplaySetFn(K)
 structure M = SplayMapFn(K)


 fun remapSet m s =












end
```

Write the code for the `remapSet` function, which takes has type `K.ord_key M.map -> S.set -> S.set` . This function takes all of the items in the Set (s), looks them up in the Map (m), and builds a new set with the corresponding values. If any item from the input set is not found in m, this function should raise `NotFound`.

For example, if `m` is the map $1 \mapsto 2, 3 \mapsto 4, 5 \mapsto 6$ and `s` is the set $1, 5$, this function should return the set $2, 6$.

You may find it useful to recall that `ORD_MAP` has:

```
val find : ('a map * Key.ord_key) -> 'a option
```

and that `ORD_SET` has:

```
val empty : set
val add : (set * item) -> set
val foldl : (item * 'a -> 'a) -> 'a -> set -> 'a
```

Question 2 Regular Languages [10 pts]

1. Which of the following strings are in the language of the regexp $a(b|c)^*de$? (Circle **all** that are in the language)

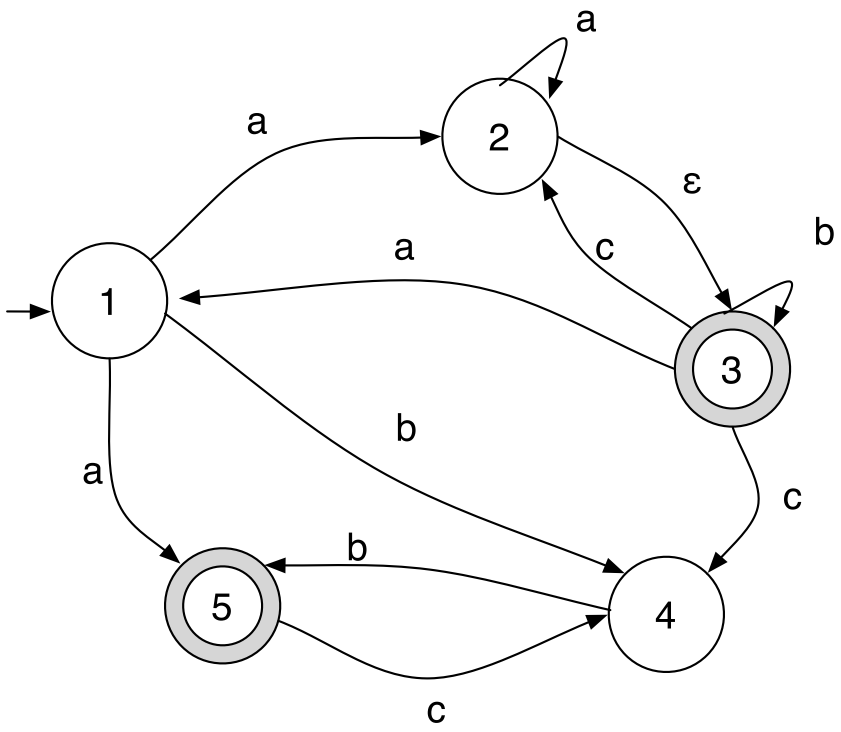
ϵ	e	ade
abde	abccbbde	abdbdcde

2. Write a regular expression for all strings of `0`s and `1`s where any occurrence of the sequence `110` is followed by `10`.

3. Write a regular expression for all strings of `x`s and `y`s where `xyy` may only appear at the end of the string (that is, if the string contain `xyy`, the no other character appear after that subsequence).

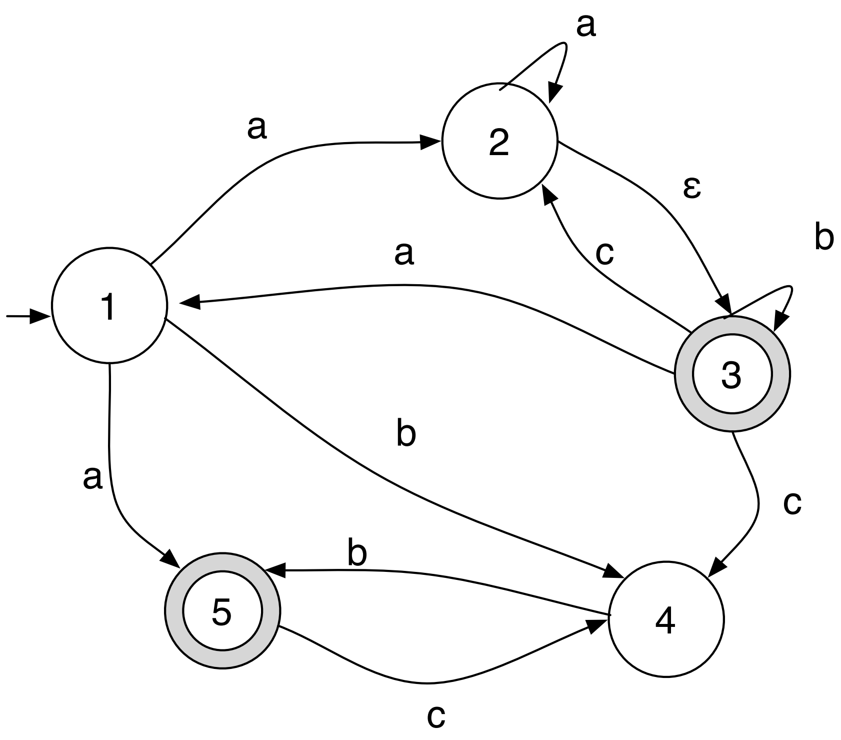
Question 3 NFA to DFA [8 pts]

Convert the following NFA to a DFA.



Question 4 NFA to Regexp [10 pts]

Convert the following NFA to a Regular Expression. Showing your work is encouraged, as it makes it easier to give partial credit if you go wrong. You have the next page for this problem too:



(This page is for the NFA to Regexp Problem)

Question 5 Context Free Grammars [10 pts]

Consider SML-lite, which is like SML, but only has the following features (and no type system):

- fn without pattern matching, and only a single variable as an argument
- variables
- function application
- parenthesis around expressions
- lists of expressions

For example, the following are legal SML-lite expressions:

```
x
fn x => x
(fn y => fn x => x y) z
[y, z, fn x => x]
```

Write a CFG for the syntax of SML-lite:

Question 6 LR Parsing [10 pts]

Draw the FSM for an LR parser for the following grammar:

```
S -> X $
X -> X Y
   | 1
Y -> 0 Y 2
   | 3
```

Question 7 Types [12 pts]

1. Suppose $\Gamma = \{f \mapsto \text{string} \to \text{int}, x \mapsto \text{int}, z \mapsto \text{Record(a:string,b:int)}\}$. Draw the typing derivation for

   ```
   z.b := f(z.a) + x;
   ```

2. Suppose $\bot \sqsubseteq$ `Lion` $\sqsubseteq$ `Cat` $\sqsubseteq$ `Animal` $\sqsubseteq \top$. For each of the following, write True if the statement is type theoretically sound, and False if it is not.

   (a) Lion ref $\sqsubseteq$ Cat ref

   (b) Lion `[]` $\sqsubseteq$ Animal ref

   (c) Lion $\to$ Cat $\sqsubseteq$ Animal $\to$ Cat

   (d) Lion $\to$ Animal $\sqsubseteq$ Cat $\to$ Cat

   (e) Cat $\to \bot \sqsubseteq \top \to$ Cat

3. Extra Credit: What property must any $\top \to \bot$ function have?

Question 8 Type Inference [7 pts]

1. Perform type inference on the following function. You MUST show your work for full credit.

```
fun f g x = case x of
              [] => []
            | (SOME a)::l => (g a)::(f g l)
            | NONE::l => (f g l)
```