ECE 553: Compilers
Midterm

Name: NetID:

There are 8 questions, with the point values as shown below. You have 75 minutes with a total of 100 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		10
2	Regular Languages		10
3	NFA to DFA		10
4	NFA to Regexp		10
5	Context Free Grammars		15
6	LR Parsing		15
7	Types		15
8	Type Inference		15
	Total		100
	Percent		100

Question 1 SML [10 pts]

1. Write `map` (curried or taking a tuple—either is fine) using `foldr`.

2. If you had written your `map` function exactly the same, but used `foldl` instead, what would have been wrong with the result?

3. Suppose I had the following:

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


 fun invert m =









end
```

Write the code for the `invert` function, which takes has type `K.ord_key M.map -> S.set M.map`. This function should create the inverse map of the map passed in. If the input map contains $1 \mapsto 2, 3 \mapsto 4, 5 \mapsto 2$, the output map should contain $2 \mapsto 1, 5, 4 \mapsto 3$—that is, if you look up a key in the output map, you should get the set of keys in the input map which map (in the input map) back to that value. You may find it useful to recall that `ORD_MAP` has:

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

and that `ORD_SET` has:

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

Question 2 Regular Languages [10 pts]

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

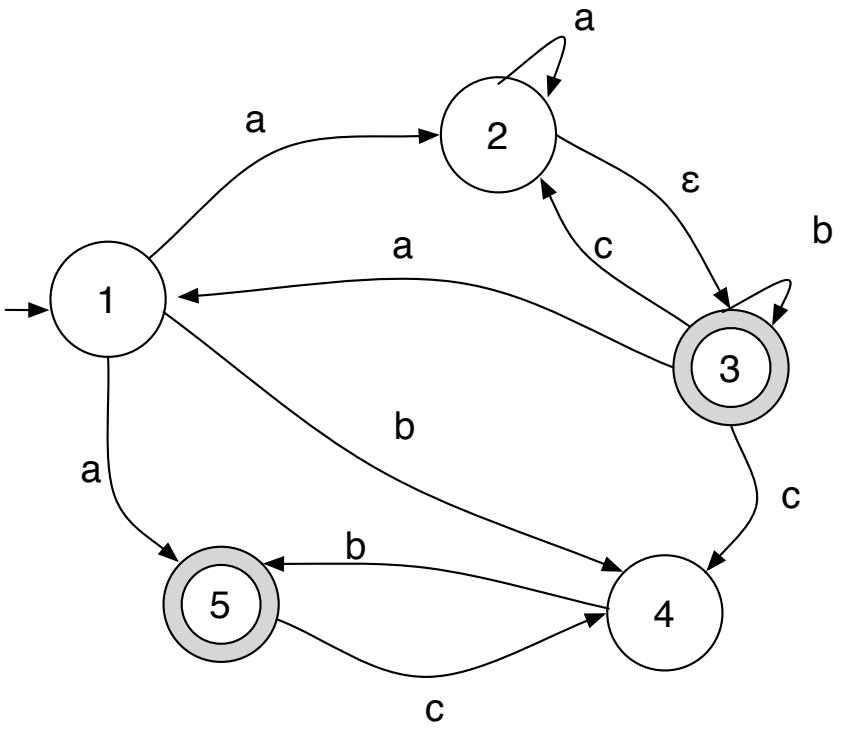
ϵ	e	ae
ade	abdaacde	abdbdcde

2. Write a regular expression for all strings of `0`s and `1`s where any occurrence of the sequence `111` is followed by at least two `0`s.

3. Write a regular expression for all strings of `x`s and `y`s where an odd number of `x`s must be followed by an even number of `y`s (an even number of `x`s may be followed by any number of `y`s.

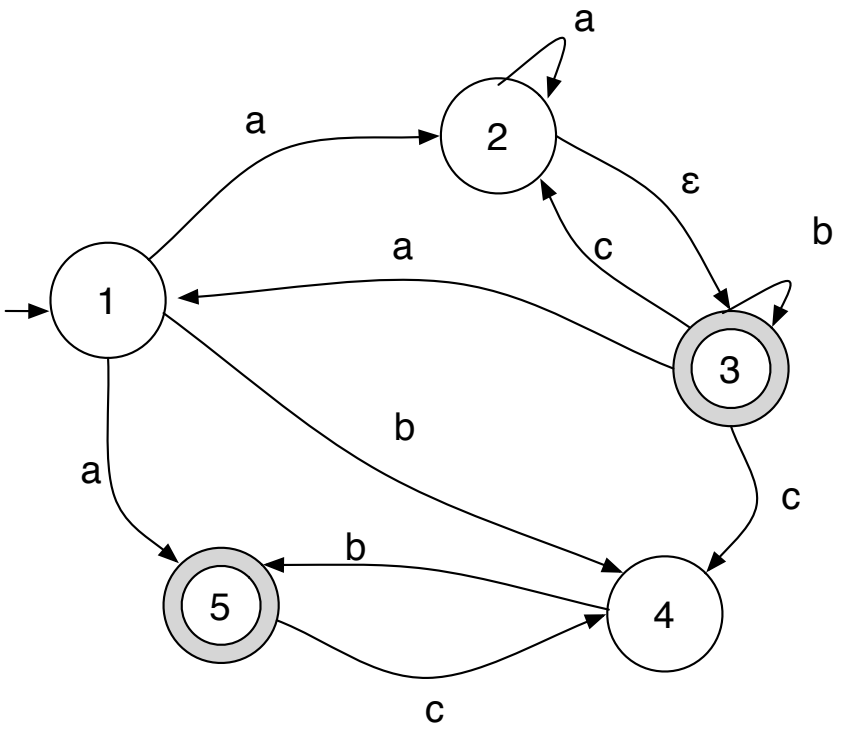
Question 3 NFA to DFA [10 pts]

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 [15 pts]

1. Which of the following strings are in the language of the following CFG? (Circle **all** that are in the language)

```
S -> S b A
   | e
A  -> a A x
   | B
B -> b B
   |    /* epsilon */
```

ϵ	e	ebaxabbbx
eb	ebbbbbbbx	eaxebexe

2. Suppose you have terminals of `id num [ ] , = {` and `}` and a language for describing values (similar to, but not the same as JSON). In this language, a value is either a number, an object, or an array. An object is represented by curly braces around a comma separated list of `id = value`s. An array is a comma separated list of values. Examples (one per line) of valid strings in this language include:

```
num
{id=num, id={id=num}, id=[num, {id=num, id=num}]}
[num, num, {id=num}]
{}
[{}, num, {id={id={id=num}, id=num}, id=[num,num]}]
```

Write the CFG for this language:

Question 6 LR Parsing [15 pts]

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

```
S -> X $
X -> a X b
   | c Y
Y -> y Y
   | z
```

Question 7 Types [15 pts]

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

   ```
   if f(y) < 4 then x + 2 else z.a
   ```

2. Suppose `Lion` $\sqsubseteq$ `Cat` $\sqsubseteq$ `Animal`. Now, suppose I have a CatInputStream (that is, an input stream that reads Cats). Fill in the blanks below with either $\sqsubseteq$ or $\sqsupseteq$ to correctly describe the subtype relationship between CatInputStream, AnimalInputStream, and LionInputStream:

   AnimalInputStream _____ CatInputStream _____ LionInputStream

3. What would the relationship be for OutputStreams (which write Animals/Cats/Lions)?

   AnimalOutputStream _____ CatOutputStream _____ LionOutputStream

Question 8 Type Inference [15 pts]

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

```
fun f x = case x of
            (g, [a]) => g a
          | (g, a::l) => g(f(g,l))
```

(More on next page)

Compiling the previous code gives "Warning: match nonexhaustive". The compiler implicitly inserts `raise Match` for the missing cases, resulting in the following being the code that was actually type checked:

```
fun f x = case x of
            (g, [a]) => g a
          | (g, a::l) => g(f(g,l))
          | (g, [] ) => raise Match
```

2. What type does the SML compiler (which does NOT have subtyping) give the expression `raise Match`?

3. If your Tiger compiler (which does have subtyping, but does NOT have parametric polymorphism) supported exceptions, what type would `raise Match` have?