Using the Lock-down Browser

You will have to answer each question using just typing Here are examples.

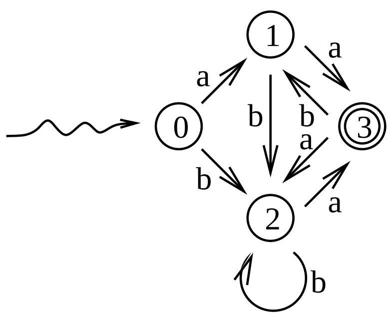
I. Design an NFA which accepts the language of all strings over a, b which have three consecutive a's.

Your answer:

States are 0, 1, 2, 3 where 3 is the only final state.

```
delta(0,a) = 0,1
delta(0,b) = 0
delta(1,a) = 2
delta(1,b) = emptyset
delta(2,a) = 3
delta(2,b) = emptyset
delta(3,a) = 3
delta(3,b) = 3
```

II. Give a minimal DFA equivalent to the following DFA:



Your answer: States 1 and 2 are equivalent. the minimal DFA has states 0, 1/2, 3, and emptyset (dead state). The transition function of the minimal DFA is:

```
delta(0,a) = 1/2
delta(0,b) = empty set (dead state)
delta(1/2,a) = 3
delta(1/2,b) = 1/2
delta(3,a) = 1/2
delta(3,b) = 1/2
```

III. Give a regular expression for the language of all strings over $\{a, b\}$ with three consecutive a's.

Your answer:

```
(a+b)*aaa(a+b)*
```

IV. Let $L = \{a^n b^n \mid n \geq 0\}$. Design a PDA which accepts L.

Your answer:

```
Three states, 0, 1, and 2  2 is the only final state.  The bottom of stack
```

```
symbol is z.
There is a self-loop at state 0.  The labels on that loop are:
z/a/az, a/a/aa.
There is an arc from 0 to 1.  The labels on that arc are:
z/lambda/z,a/lambda/a.
There is a self-loop at state 1.  The label on that loop is:
a/b/lambda.
There is an arc from 1 to 2.  The label on that arc is
z/lambda/lambda.
```

V. Consider the following CNF grammar:
$S \to AB$
$A \to a$
$B \to b$
Use the CYK algorithm to prove that ab is in the language generated by the grammar.

Your answer:

`V[1,1] = A, V[2,2] = B, V[1,2] = S`

Therefore ab is generated by the grammar.

VI. Write exponents using the carat symbol. That is, a^2 should be written `a^2`

Practice Exam

1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time. In the questions below, $\mathcal{P}$ and $\mathcal{NP}$ denote $\mathcal{P}$-TIME and $\mathcal{NP}$-TIME, respectively.

   (i) ______ Let L be the language over $\{a, b, c\}$ consisting of all strings which have more a's than b's and more b's than c's. There is some PDA that accepts L.

   (ii) ______ The language $\{a^n b^n \mid n \geq 0\}$ is context-free.

   (iii) ______ The language $\{a^n b^n c^n \mid n \geq 0\}$ is context-free.

   (iv) ______ The language $\left\{a^i b^j c^k \mid j = i + k\right\}$ is context-free.

   (v) ______ The intersection of any three regular languages is regular.

   (vi) ______ The intersection of any two context-free languages is context-free.

   (vii) ______ If L is a language and L^* is regular, then L must be regular.

   (viii) ______ If L is a context-free language over an alphabet with just one symbol, then L is regular.

   (ix) ______ The set of strings that your high school algebra teacher would accept as legitimate expressions is a context-free language.

(x) ______ Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

(xi) ______ Every context-free language is generated by some unambigous context-free grammar.

(xii) ______ The problem of whether a given string is generated by a given context-free grammar is decidable.

(xiii) ______ If G is a context-free grammar, the question of whether $L(G) = \emptyset$ is decidable.

(xiv) ______ Every language generated by an unambiguous context-free grammar is accepted by some DPDA.

(xv) ______ The language $\{a^n b^n c^n d^n \mid n \geq 0\}$ is decidable.

(xvi) ______ Every problem that can be mathematically defined has an algorithmic solution.

(xvii) ______ $\mathcal{P} = \mathcal{NP}$.

(xviii) ______ There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a binary numeral.

(xix) ______ The language consisting of all strings over $\{a, b\}$ which have more a's than b's is context-free.

(xx) ______ Every context-free language is in $\mathcal{P}$.

(xxi) ______ Every context-sensitive language is in $\mathcal{P}$.

(xxii) ______ Every language generated by a general grammar is decidable.

(xxiii) ______ The problem of whether two given context-free grammars generate the same language is decidable.

(xxiv) ______ Every bounded function is recursive (that is, computable).

(xxv) ______ Recall that if $\mathcal{L}$ is a class of languages, co-$\mathcal{L}$ is defined to be the class of all languages that are not in $\mathcal{L}$. Let $\mathcal{RE}$ be the class of all recursively enumerable languages. If L is in $\mathcal{RE}$ and also L is in co-$\mathcal{RE}$, then L must be decidable.

(xxvi) ______ If a language L is both $\mathcal{NP}$ and co-$\mathcal{NP}$, L must be $\mathcal{P}$.

(xxvii) ______ Every language is enumerable. That means, either L is finite or there is a one-to-one function from the positive integers to L.

(xxviii) ______ There is a non-recursive function which grows faster than any recursive function. (Recursive function means computable function.)

(xxix) ______ There exists a machine[1] that runs forever and outputs the string of decimal digits of π (the well-known ratio of the circumference of a circle to its diameter).

[1] As always in automata theory, "machine" means abstract machine, a mathematical object whose memory and running time are **not** constrained by the size and lifetime of the known (or unknown) universe, or any other physical laws. A computer program, which can have any length, can be considered to be a machine; by assuming it runs on an abstract machine. If we want to discuss the kind of machine that exists (or could exist) physically, we call it a "physical machine."

(xxx) ______ For every real number x, there exists a machine that runs forever and outputs the string of decimal digits of x.

(xxxi) ______ There is a polynomial time algorithm which determines whether any two regular expressions are equivalent.

(xxxii) ______ If two regular expressions are equivalent, there is a polynomial time proof that they are equivalent.

(xxxiii) ______ The halting problem is undecidable.

2. Let L be the language consisting of all strings over the binary alphabet whose last three symbols are '`010`.' Design an NFA with four states which accepts L.

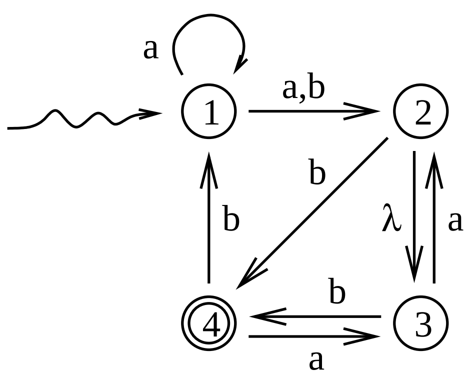
3. Let L be the language consisting of all strings over $\{a, b\}$ which do not contain the substring aab. Write a regular expression for L and draw a minimal DFA which accepts L. (Hint: 3 states.)

4. Consider the NFA shown below.

   (i) Draw a state diagram for a minimal DFA equivalent to that NFA.

   (ii) Write a regular expression which describes the language accepted by that NFA.

   (iii) Give a regular grammar which generates the language accepted by that NFA.



5. Consider the language L generated by the CF grammar given below.

   $S- > wS$
   $S- > iS$
   $S- > iSeS$
   $S- > a$

   (i) Give a Chomsky Normal Form grammar for L.

   (ii) Use the CYK algorithm to decide whether $iiwaea \in L$. Show all values of $V_{i,j}$.

6. Let L be the set of all algebraic expressions, where the only operations permitted are addition, subtraction, and multiplication, where there are only two variables, x and y and no constants, and where multiplication is indicated by concatenation, as in the expression $x + y(xy + x)$. Parentheses can be used. Write a context-free grammar for L. Your grammar may be ambiguous.

7. Design a PDA which accepts the Dyck language. To make it easier to grade, use a and b for left and right parentheses.