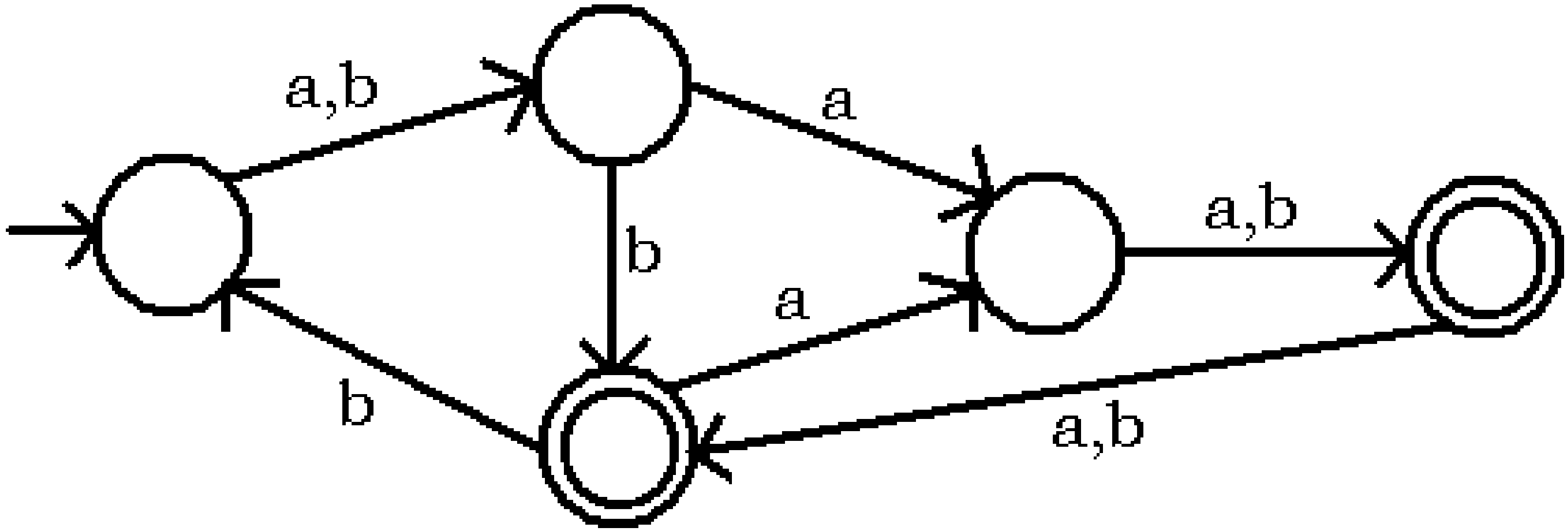
Given the DFA below which accepts L, prove that $L^R = \{ u^R : u \in L\}$ is regular by designing a NDFA which accepts L^R.

For example, abaa is in L so $(abaa)^R$ = aaba is in L^R.



Announcements

The midterm is in class on Wed. June 21.
There is a tutorial on Tuesday June 13.
No tutorial on Tuesday June 20.
Midterm tutorial: Monday June 19, 6:30pm, ECS 123.
Bring any questions you have about assignments 1-3, old midterms or any other class material.

Assignment 3 is due on Friday at the beginning of class.

The preliminary final exam schedule has CSC 320 at 2pm on Monday August 14.

Let $L = \{ a^n b^p : n+4 \le p \le 5n+7\}$

1. What string w would you choose if you want to prove that L is not regular?

2. What are the cases for y?

3. How many times would you pump for each case?

4. If you chose instead $w = a^r b^{2r}$ where all you assume is that $k \le 3r$, how does this change the proof?

The Pumping Lemma for Regular Languages:

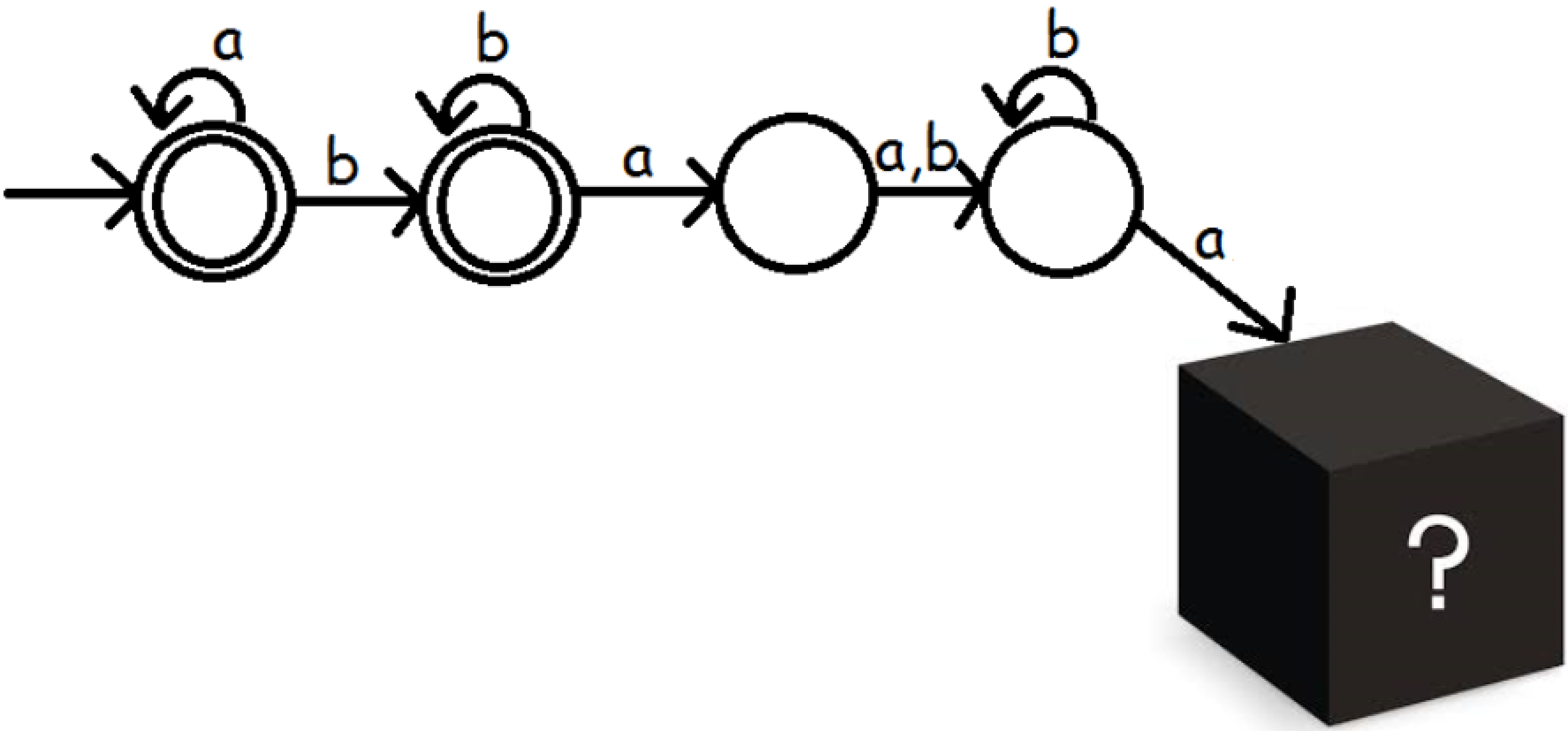
If L is a language accepted by a DFA with k states, and $w \in L$, $|w| \geq k$, then there exists x, y, z such that

1. $w = x\,y\,z$,
2. $y \neq \varepsilon$,
3. $|\,x\,y\,| \leq k$, and
4. $x\,y^n z$ is in L for all $n \geq 0$.

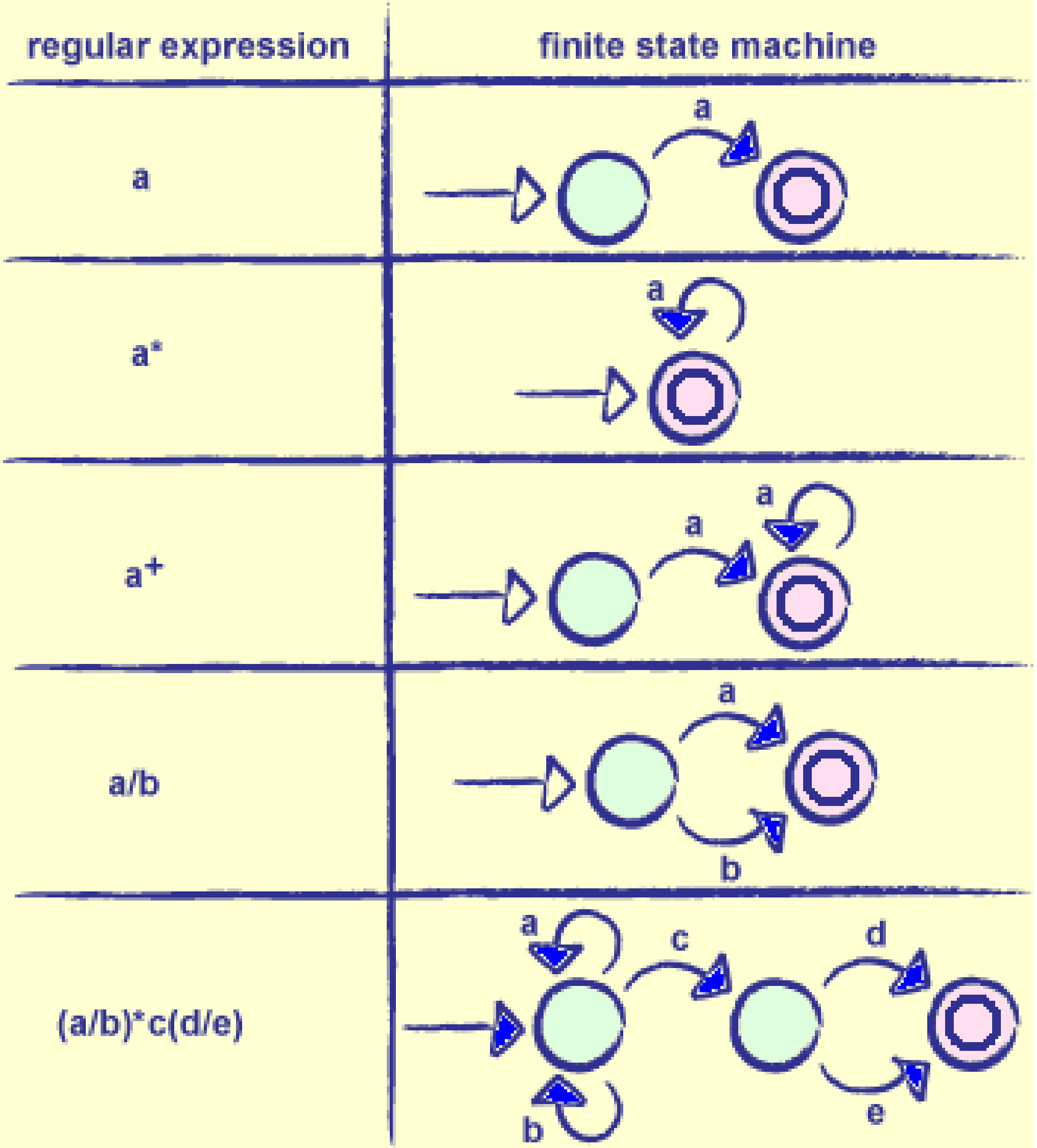
Recipe for using the pumping lemma (proof by contradiction):

1. Assume L is regular and is accepted by a DFA M that has k states.
2. Choose w so that $w \in L$ and $|w| \geq k$.
3. Find all factorizations of w as w=xyz such that $|xy| \leq k$ and y is not the empty string.
4. For each factorization of w, find n so that when you pump n times to get $x y^n z$ the resulting string is not in L. Make sure you look at the definition of L here and not just the pattern for w.

What machine M_2 would your construction from assignment 2 create as input for isEmpty given this machine M_1 and u=aaba:



Black box image from:
http://socialcapitalmarkets.net/2013/04/03/whats-inside-the-socap13-black-box/



http://www.cgl.ucsf.edu/Outreach/bmi280/slides/swc/lec/re.html

Algorithms to Answer Questions about Regular Languages

if (if23==23)
  x= -23.2e23-6;

The first step of a compiler is to break your program into tokens.
Tokens:

```
if (if23==23)
   x= -23.2e23-6;
```

Keywords: if

Brackets: ( )

Variables: if23 x

Assignment: =

Math Operator: -

Logical: ==

Delimiter: ;

Double: -23.2e23

Integers: 23 6

Keywords:

if U while U int U double U switch U ....

Variables: Not a Keyword but of the form:

(a-z U A-Z)(a-z U A-Z U 0-9 U _)*

Non-negative Integers: N= (0 U (1-9)(0-9)*)

Numeric values:

(Φ* U -) U N (Φ* U . (0-9)*)

(Φ* U (eUE)(Φ* U + U -) N)

The pumping lemma or closure properties can be used to prove languages are not regular.

Regular languages are closed under:

- union
- concatenation
- Kleene star
- complement
- intersection
- exclusive or
- difference
- reversal

There are algorithms for the following questions about regular languages:

1. Given a DFA M and a string w, is $w \in L(M)$?

2. Given a DFA M, is $L(M) = \Phi$?

3. Given a DFA M, is $L(M) = \Sigma^*$?

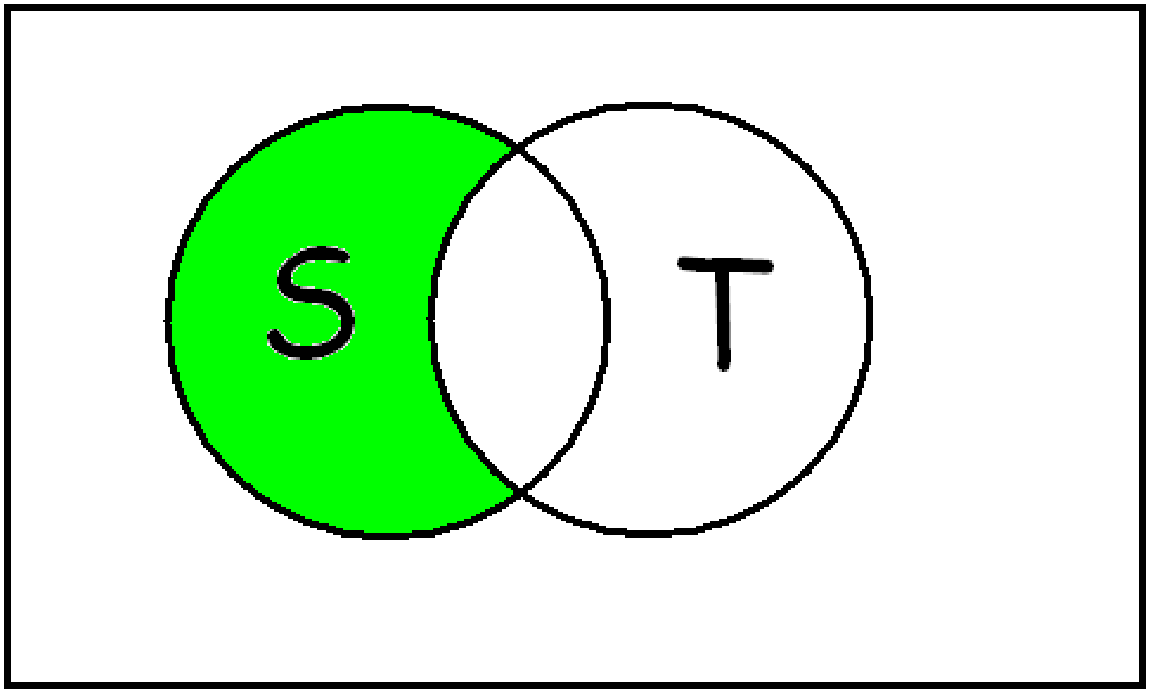
4. Given DFA's M_1 and M_2, is $L(M_1) \subseteq L(M_2)$?

5. Given DFA's M_1 and M_2, is $L(M_1) = L(M_2)$?

How can we use these to check correctness of student answers for the java tutorial?

Java Regular expression tutorial:

S= student answer, T= teacher answer

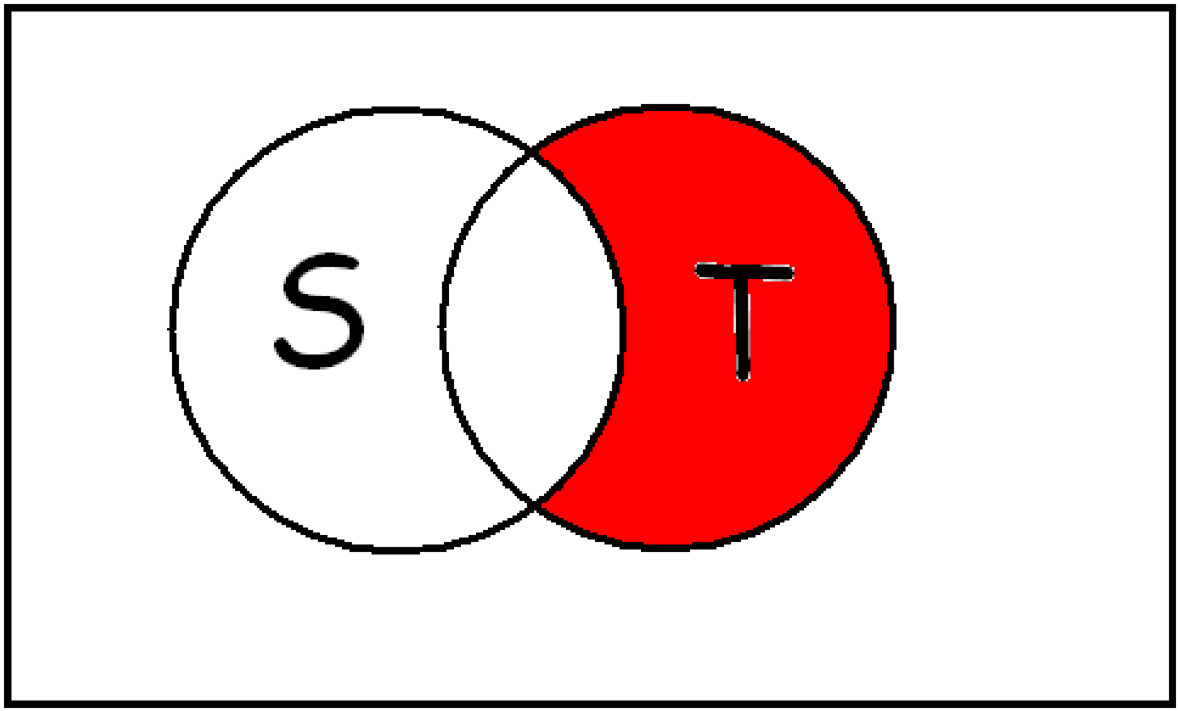


Strings student generates but should not.

Is S intersect the complement of T = Φ?

Java Regular expression tutorial:

S= student answer, T= teacher answer



Strings student should generate but does not.

Is T intersect the complement of S = Φ?

The Pumping Lemma for Regular Languages:

If L is a language accepted by a DFA with k states, and $w \in L$, $|w| \geq k$, then $\exists$ x, y, z such that

1. $w = x\,y\,z$,
2. $y \neq \varepsilon$,
3. $|\,x\,y\,| \leq k$, and
4. $x\,y^n\,z$ is in L for all $n \geq 0$.

The pumping lemma is NOT strong enough to work directly to prove that certain languages are not regular.

Let L be a language which has a constant k such that for all $w \in L$, $|w| \geq k$, $\exists$ x, y, z such that

1. $w = x\, y\, z$,
2. $y \neq \varepsilon$,
3. $|\, x\, y\,| \leq k$, and
4. $x\, y^n z$ is in L for all $n \geq 0$.

This is necessary but not sufficient for a language to be regular.

Then you CANNOT conclude that L is regular.

Counterexample: See assignment 3.

$L_1 = \{\, u\, u^R\, v : u, v \text{ in } \{0, 1\}^+ \}$