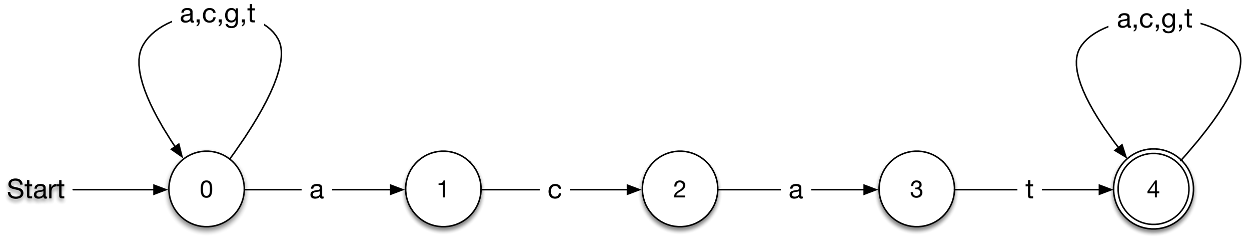
Homework 3 — Algorithms

Due February 10, 2020

1. Converting an NFA to a DFA:

Use the construction in the slides to convert the following NFA to a DFA. The NFA recognized all strings containing “acat” in the alphabet Σ={a,c,g,t} of DNA strings. It basically decides sometimes on seeing an “a” in the initial state that this might be the beginning of the pattern. This is actually the only feature of non-determinism.



2. Use mathematical induction to show that when *n* is an exact power of 2, the solution of the following recurrence is *T*(*n*) = n $\log_2$(n).

$$T(n) = \begin{cases} 2, & \text{for } n = 2 \\ 2T(n/2) + n, & \text{for } n = 2^k \text{ with } k > 1 \end{cases}$$

3. Given the following C-program, show that the loop invariant $y = 2^i - 1$ is true. Deduce the value of y after the function has run.

```
extern int i;

y=0;
for(i=0; i<=n; i++)    {
   y += pow(2,i);
}
```

Hint: Find an expression of i that is equal to y before the loop starts, that if before an iteration is true, is also true after the iteration.

4. Given the following Python program, prove the loop invariant acc $= \frac{i(i+1)}{2}$.

```
def litgau(n):
   i = 0
   acc = 0
   while i <= n:
      acc += i
      i += 1
   return acc
```

5. Use induction to show that for the Fibonacci numbers $f_0 = 0, f_1 = 1, f_{i+2} = f_{i+1} + f_i$ and with $\Phi = \dfrac{1+\sqrt{5}}{2}$, we have $f_n > \Phi^{n-2}$. You might want to show first that $1 + \Phi = \Phi^2$.