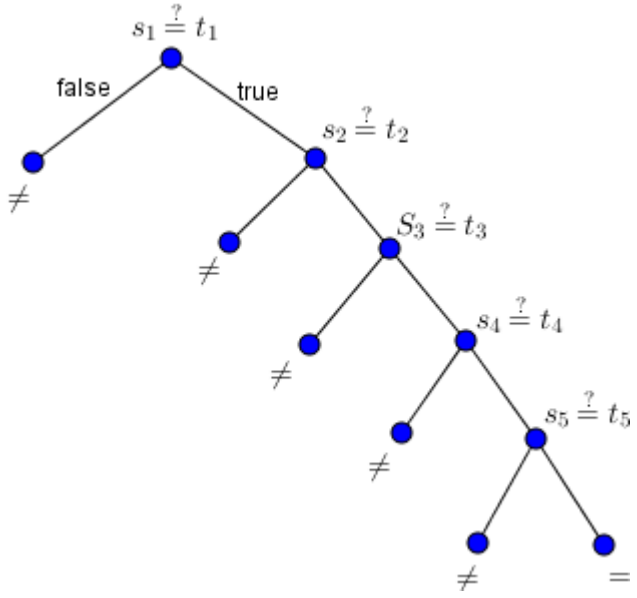
Math 15 - Spring 2017 - Homework 5.4 Solutions

1. (5.4 # 18) Suppose s and t are two strings of length 5 in the alphabet $a \ldots z$. An algorithm checks to see if $s = t$ by comparing corresponding symbols in each string, from left to right. Draw the decision tree that models this algorithm.

   ANS:



2. (5.4 # 20) Suppose U is a set of n integers. Approximate the worst-case complexity of an algorithm that runs through all possible subsets of U and adds up the elements in each subset to see if there is a subset whose elements add up to 0.
   The power set of U contains 2^n subsets. In the worst case, you'd need to look at all these, taking $\Theta(n)$ operations to add the elements of each subset. So the complexity is $\Theta(n \cdot 2^n)$.
   In class we addressed the problem of devising such an algorithm and came up with this:

```
import random
n=15
m=1000
L = [random.randrange(-m,m) for i in range(n)]
#L

#L = [-15,-12,-9,-4,2,5,6,14,23]
for i in range(1,2**n):
    S = 0
    k=i
    x = []
    for j in range(n):
        if i%2 == 1:
            S += L[j]
            x += [L[j]]
        i //= 2
    if S==0:
        print(k)
        print(x)
```

3. (5.4 # 22) Suppose P is a formula in propositional logic containing n variables. Approximate the worst-case complexity of an algorithm that runs through all possible true/false values of the n variables

to see if there is an assignment that makes P true.
ANS: This is much like the previous problem. There are 2^n different ways the n variables can be assigned true/false values. Assuming that the number of operations in the formula is $\Theta(n)$, the complexity is $\Theta(n \cdot 2^n)$.