# Regular Expression and Finite State Automation

## Regular Expression

The below regular expression matches literals “a”, “an” or “the” without case sensitivity.

[aA][nN]?|[Tt][Hh][Ee]

## Finite State Automation

Below diagram represents the state transitions for the above regular expression.

**S1** is the starting state.

**S2**, S4, S5, S6 represents the transit states during traversal.

**S3** is the success state when input string matches the regular expression at the end of traversal.

**E1** is the state reached when the input string fails to match the regular expression during the traversal.

H|h

E|e

T|t

A|a

[]

[]

N|n

[]

## Python function used

* Python function findall (pattern, string) included in the “re” python library is used to search every occurrence of the above regular expression in the string passed to censor (string) custom function.

# Linear Algebra

## Solution

Linear equations are formed for the given problem and solved using matrix multiplication rules.

Below are the linear equations formed.

an \* a + bn \* b = n

ap \* a + bp \* b = p,

where,

* an is the amount of nitrogen in 1kg of type A fertiliser
* ap is the amount of phosphate in 1kg of type A fertiliser
* bn is the amount of nitrogen in 1kg of type B fertiliser
* bp is the amount of phosphate in 1kg of type B fertiliser Page 3
* n is the amount of nitrogen required by the crop
* p is the amount of phosphate required by the crop
* a is the amount of fertilizer of type A required &
* b is the amount of fertilizer of type B required.

Converting the linear equations to matrix form,

Applying inverse rule,

## 2.2 Python library/ functions used

* numpy library is imported
* numpy.array function is used to form the matrices
* numpy.inv function is used to find the inverse of the matrix