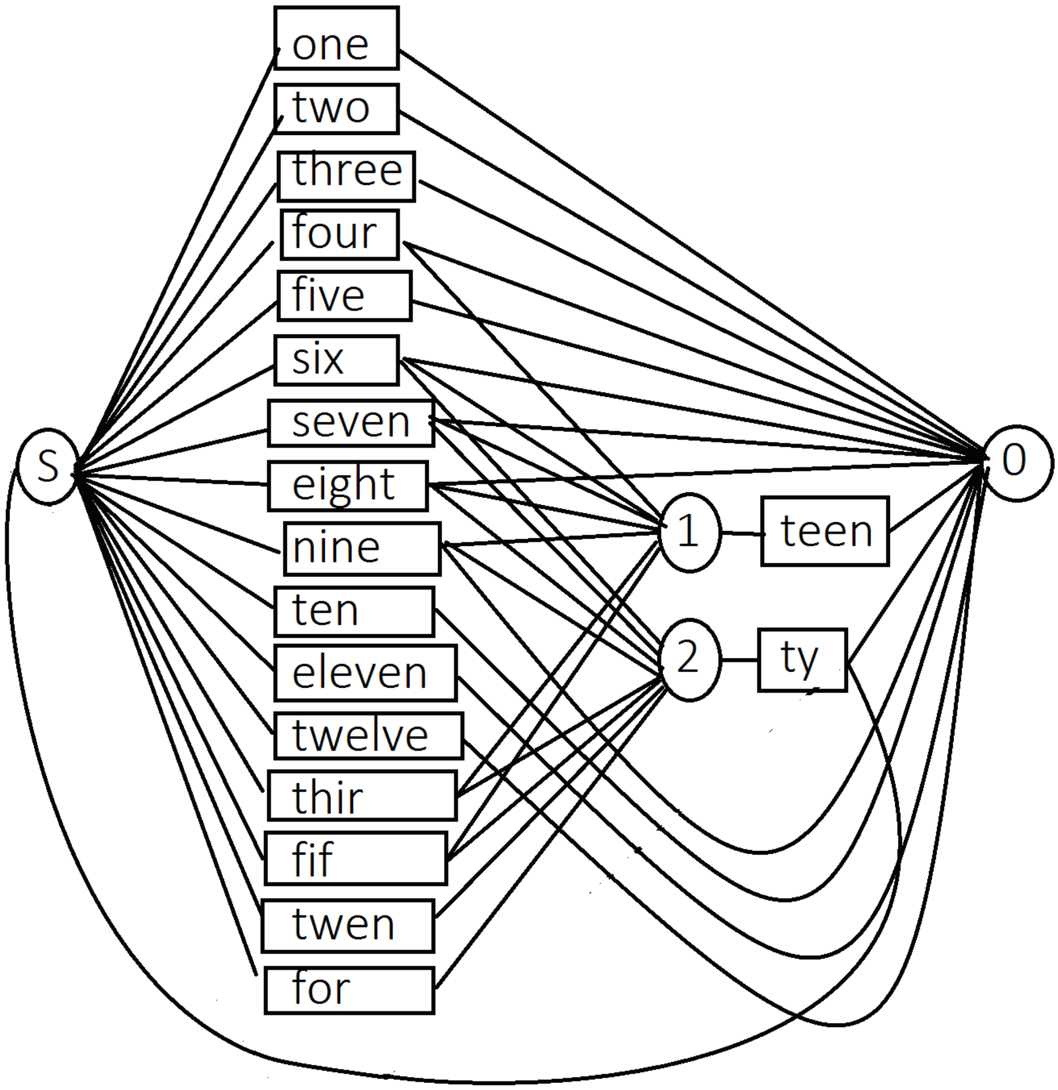
<u>Question 5:</u> Transitions

Author: Harold Somers

The diagram on the next page represents a “transition network”. The circles represent “states” while the boxes represent sequences of letters that can be “generated” from any given state, as indicated by the lines (the “transitions”). The aim is to start at “S” and get to the end state “0”. For some boxes there is a choice of transition. The lines are directional (it would have been even more messy to add the arrowheads), so note that you can only enter a state from the front (left). So you can go from “ty” to “S”, but not the other direction. As you can see, by following the paths you can generate the names of some numbers in English.



The above diagram is already quite messy, and it can be represented more neatly by a set of rules as below. Each rule is identified (in square brackets) but this is ONLY for ease of reference in answering the questions. Apart from that each rule consists of a state (the symbol before the “:”), a sequence of letters, and then, after the arrow (“→”) a list of states to which you can then move. Starting at position "S", you generate the text indicated, and then continue to any ONE of the rules whose start state is listed after the arrow. State "0" is a special case meaning "finish".

[a]	S: one → 0	[j]	S: ten → 0
[b]	S: two → 0	[k]	S: eleven → 0
[c]	S: three → 0	[l]	S: twelve → 0
[d]	S: four → 0,1	[m]	S: thir → 1,2
[e]	S: five → 0	[n]	S: fif → 1,2
[f]	S: six → 0,1,2	[o]	S: twen → 2
[g]	S: seven → 0,1,2	[p]	S: for → 2
[h]	S: eight → 0,1,2	[q]	1: teen → 0
[i]	S: nine → 0,1,2	[r]	2: ty → S,0

So for example, starting at S we can generate "fourteen" by taking rule [d] to state 1, then rule [q] to finish. We cannot generate "twelveteen" because rule [l] only allows one way to progress, namely to finish.

C1. Write out the sequence of rules and states followed to generate the following words: For example, for "fourteen" write "d 1 q 0". [12 points]

(a) sixteen
(b) ninetythree
(c) twentyeight
(d) fifteen

C2. The network above "overgenerates", that is, it allows us to create sequences which are not valid number names. Indicate whether each of the following words can be generated by the network or not. [4 points]

(a) oneten
(b) fiftytwelve
(c) sixteensix
(d) twentyseventeen
(e) fortythirty
(f) eleventythree
(g) fivety

C3. The above network wrongly generates a misspelling in the case of "eighteen" and "eighty...". Suggest a simple fix for this (i.e. a change to one of the existing rules and an additional rule).

[4 points]