Hunter Johnston

GRAMMAR:

e = epsilon
(++),(—),(-),(+) are all one terminal, while (E) is not.
n = a number (neg or pos)

E -> TEP
EP -> +TEP | -TEP | e
T -> FTP
TP -> *FTP | /FTP | %FTP | e
F -> (+)FP | (-)FP | (++)FP | (--)FP | nFPP | (E)FPP
FP -> (++)FP | (--)FP | nFPP | (E)FPP
FPP -> (++)FPP | (--)FPP | e

EPS,FIRST,FOLLOW,PREDICT sets:

EPS(E) = F
EPS(EP) = T
EPS(T) = F
EPS(TP) = T
EPS(F) = F
EPS(FP) = F
EPS(FPP) = T

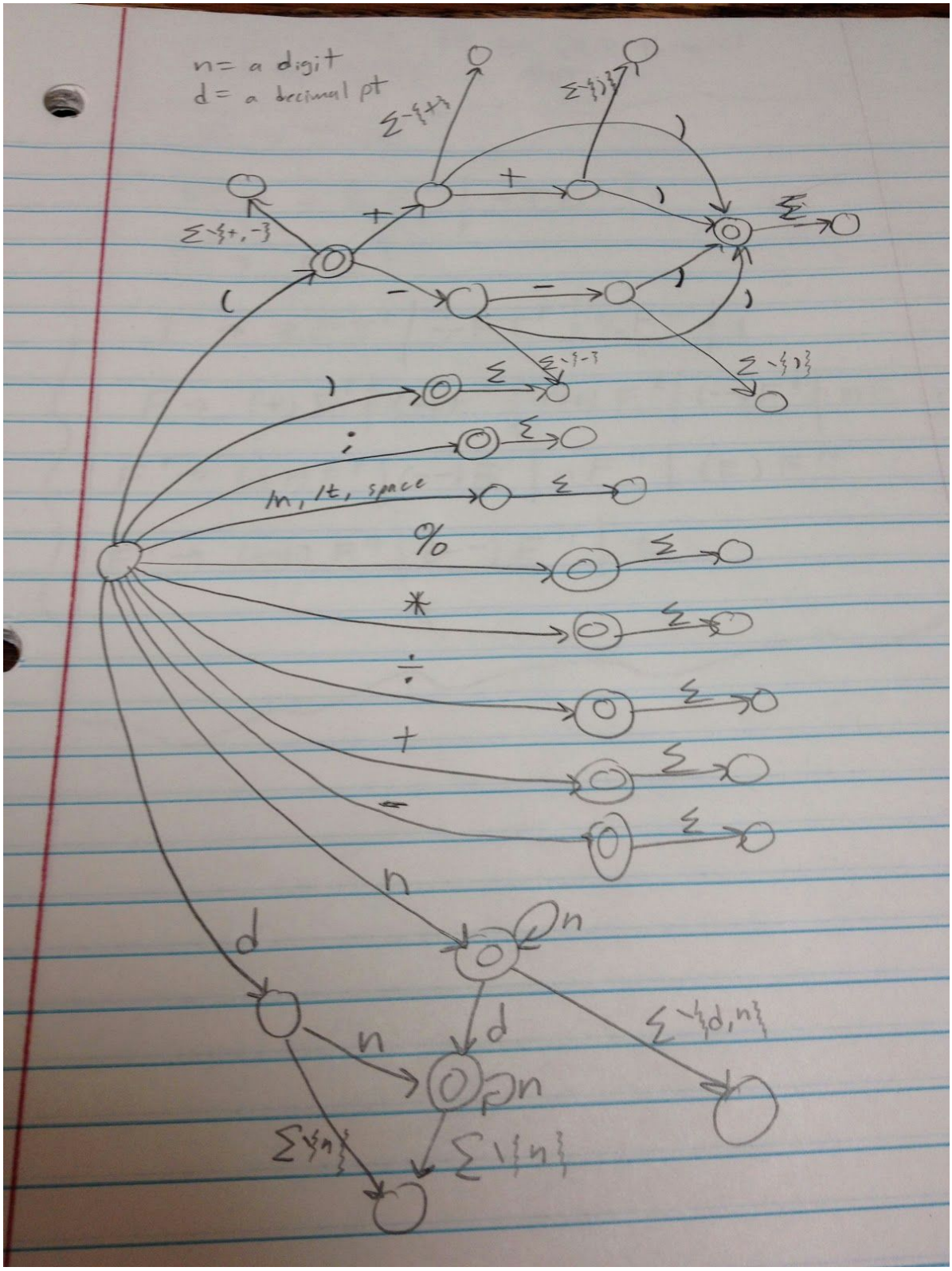
FIRST(E) = {(+),(-),(++),(--),n,(}
FIRST(EP) = {+,-}
FIRST(T) = {(+),(-),(++),(--),n,(}
FIRST(TP) = {*,/,%}
FIRST(F) = {(+),(-),(++),(--)n,(}
FIRST(FP) = {(++),(--),n,(}
FIRST(FPP) = {(++),(--)}

FOLLOW(E) = {)}
FOLLOW (EP) = {)}
FOLLOW(T) = {+,-,)}
FOLLOW(TP) = {+,-,)}
FOLLOW(F) = {*,/,%,+,-,)}

FOLLOW(FP) = {*,/,%,+,-,)}
FOLLOW(FPP) = {*,/,%,+,-,)}

PREDICT(E -> TEP) = {(+),(-),(++),(--),n,(}
PREDICT(EP -> +TEP) = {+}
PREDICT(EP -> -TEP) = {-}
PREDICT(EP -> e) = {)}
PREDICT(T -> FTP) = {(+),(-),(++),(--),n,(}
PREDICT(TP -> *FTP) = {*}
PREDICT(TP -> /FTP) = {/}
PREDICT(TP -> %FTP) = {%}
PREDICT(TP -> e) = {+,-,)}
PREDICT(F -> (+)FP) = {(+)}
PREDICT(F -> (-)FP) = {(-)}
PREDICT(F -> (++)FP) = {(++)}
PREDICT(F -> (—)FP) = {(--)}
PREDICT(F -> nFPP) = {n}
PREDICT(F -> (E)FPP) = {(}
PREDICT(FP -> (++)FP) = {(++)}
PREDICT(FP -> (—)FP) = {(--)}
PREDICT(FP -> nFPP) = {n}
PREDICT(FP -> (E)FPP) = {(}
PREDICT(FPP -> (++)FPP) = {(++)}
PREDICT(FPP -> (—)FPP) = {(--)}
PREDICT(FPP -> e) = {*,/,%,+,-,)}

SCANNER DFA:



NOTE: insert arrows from each “trash” state back to itself with the whole alphabet as a transition