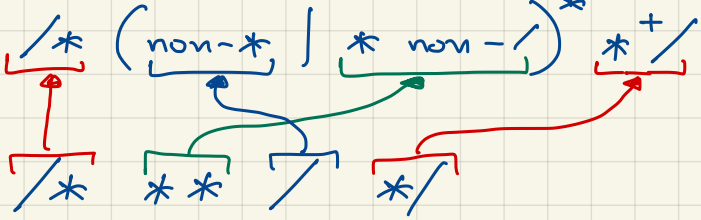
A1 sol

A1-sol

Q1 (a) /***/*/ is not a valid comment but it is accepted by the regular expression as shown below:



(b) Correct expression:

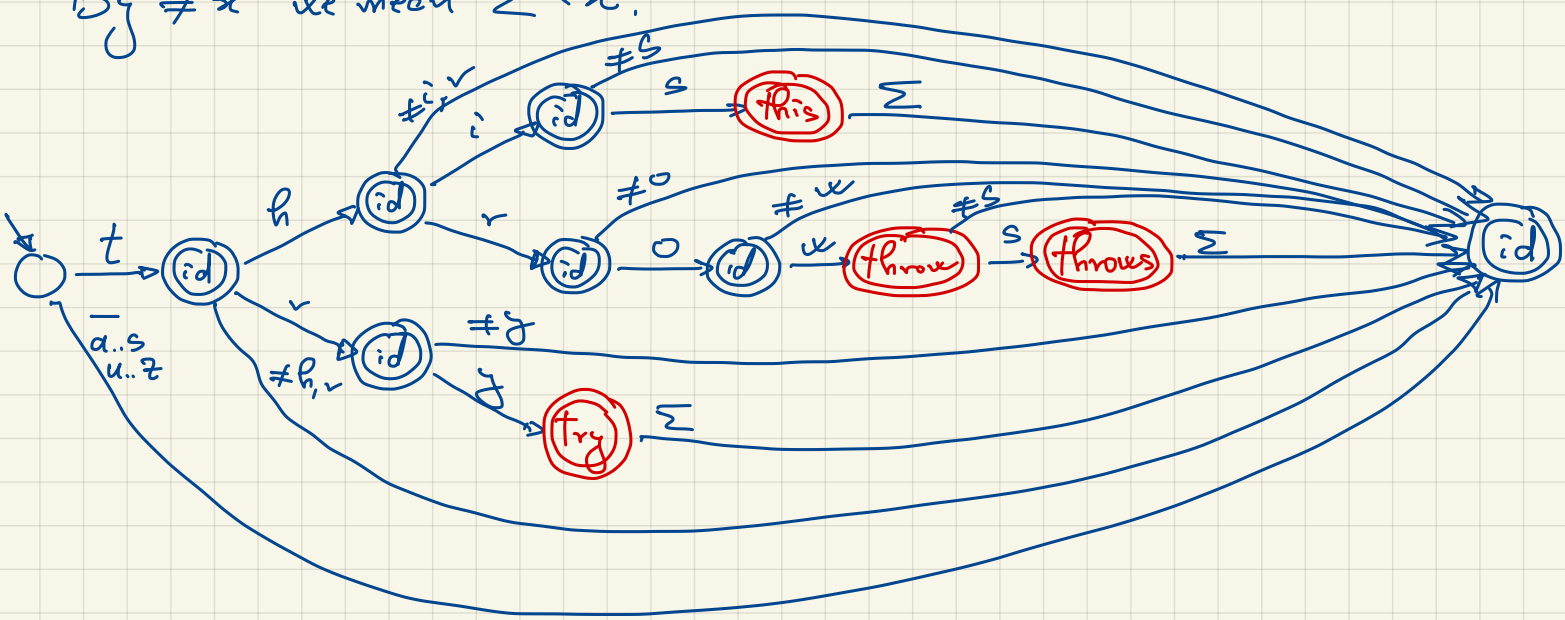
/* (non-* | *+ (non-*/))* *+/

Call *'s within the comment, except for those at the very end (*...*/), are matched here, followed by a non-*/ character.

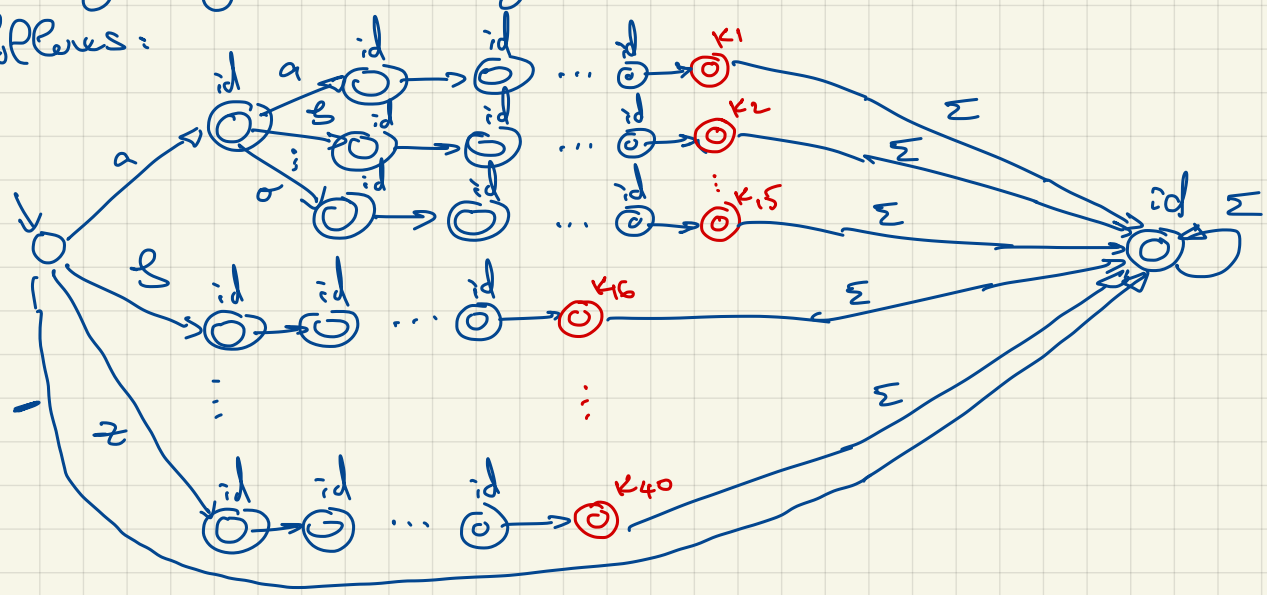
Q2 (a)

Denote $\Sigma = \{-, a, b, \ldots, z, 0, 1, \ldots, 9\}$.

By $\neq x$ we mean $\Sigma \setminus x$.



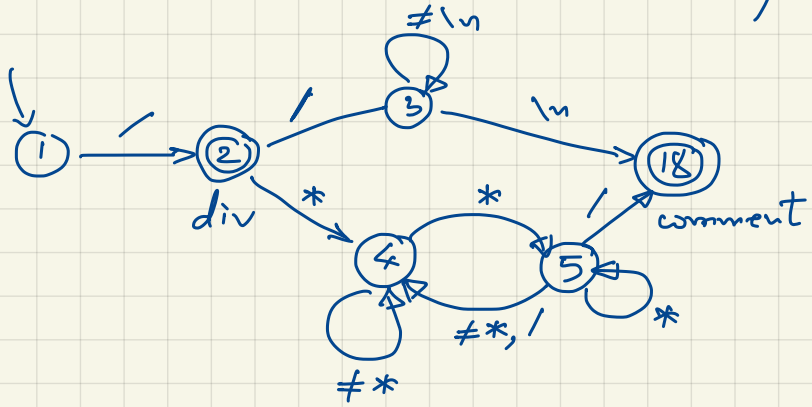
(b) We need to maximize "branching". However, the number of letters, 26, imposes a limit over how much we can branch at the beginning. The largest number of states is achieved as follows:



For keywords K_i, $1 \le i \le 40$, denoting $|K_i|$ the number of characters in K_i, the number of states is $\sum_{i=1}^{40} |K_i| - 14 + 2 = \sum_{i=1}^{40} |K_i| - 12$

Q3 (a)

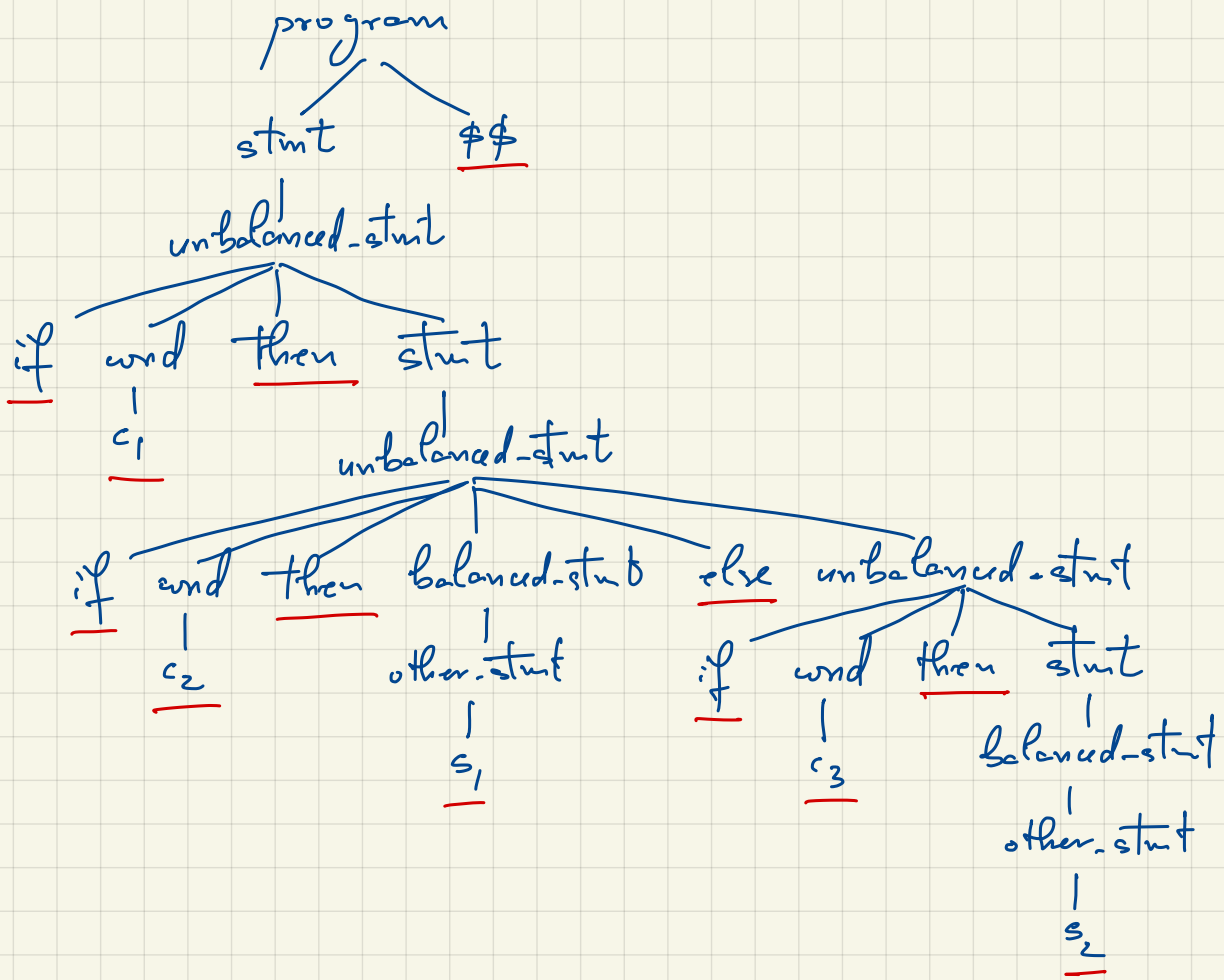
The DFA is slightly different from the one in Fig. 2.6. The "comment" is a token with that part of DFA looking like this:



Therefore, we can choose t_1 = / - div
w = /* - not a token
t_2 = /**/ - comment

(b) Example: /*** (unread characters)

Q4 (a)



(b)

X	FIRST(X)	FOLLOW(X)
program	if(1,1) si(1,1)	$$(2,1)
stmt	if(1,2) si(1,2)	else(2,4) $$(3,2)
balanced_stmt	if(1,4) si(1,5)	$$(3,3)
unbalanced_stmt	if(1,6)	else(3,5), $$(3,5)
other_stmt	si(1,9)	then(2,4)
cond	ci(1,8)	

(c)

PREDICT

1. if, si
2. if, si
3. if
4. if
5. si
6. if
7. if
8. ci
9. si

(d)

stmt → balanced_stmt
stmt → unbalanced_stmt

} same LHS, the PREDICT of both contains if

(e) - there is no left-recursion
- there are common prefixes for productions 6 and 7

- new grammar has the following productions in place of former 6-7:

unbalanced_stmt → if cond then unbal-stmt-tail

unbal-stmt-tail → stmt

unbal-stmt-tail → balanced-stmt else unbalanced_stmt

(f) It is not LL(1). It has the same conflict from before:

stmt → balanced-stmt
stmt → unbalanced-stmt

} same LHS, the PREDICT of both contains if