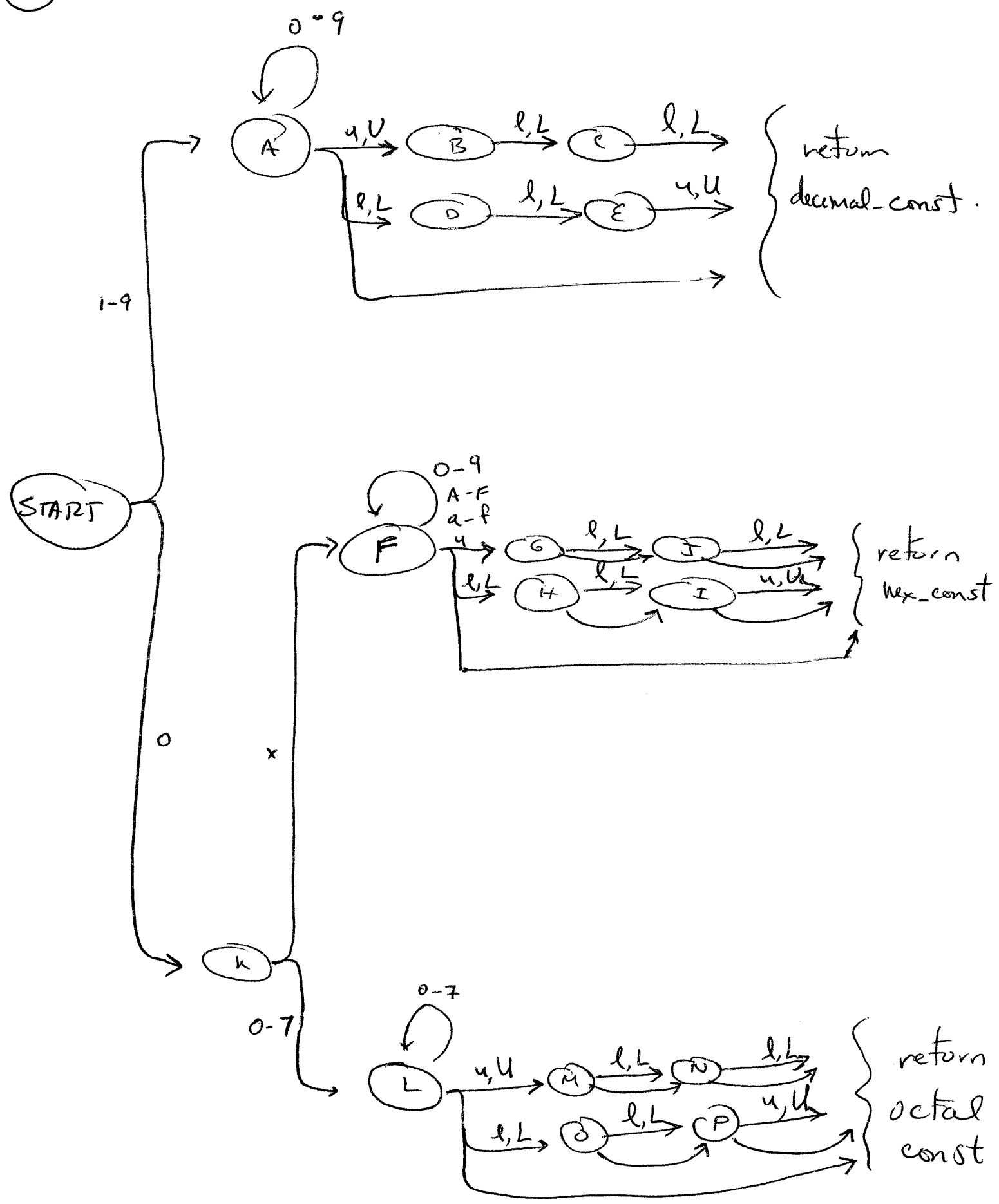
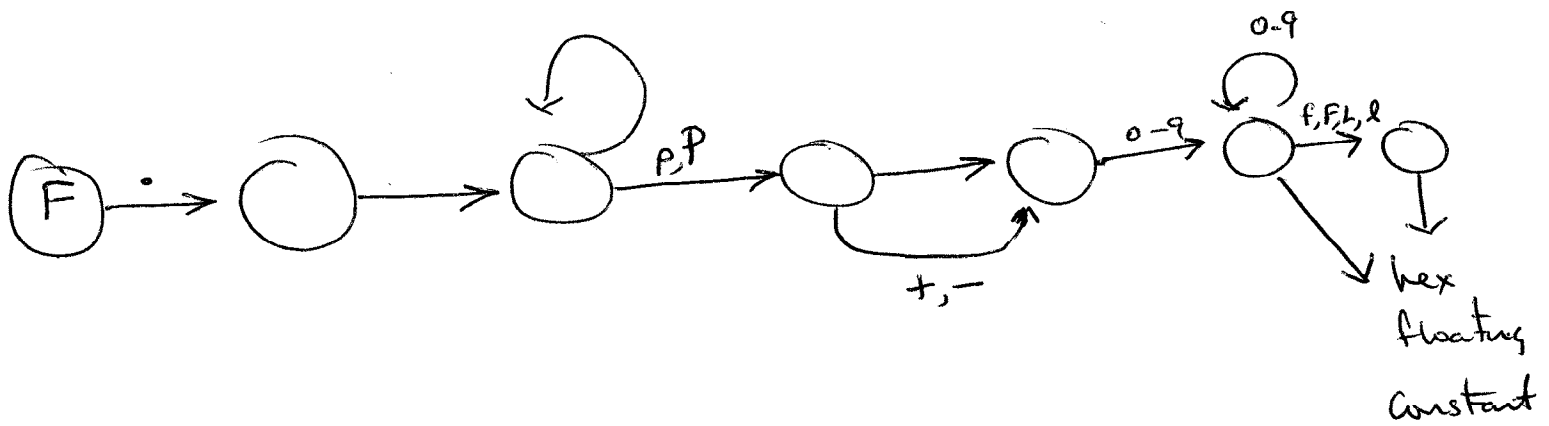
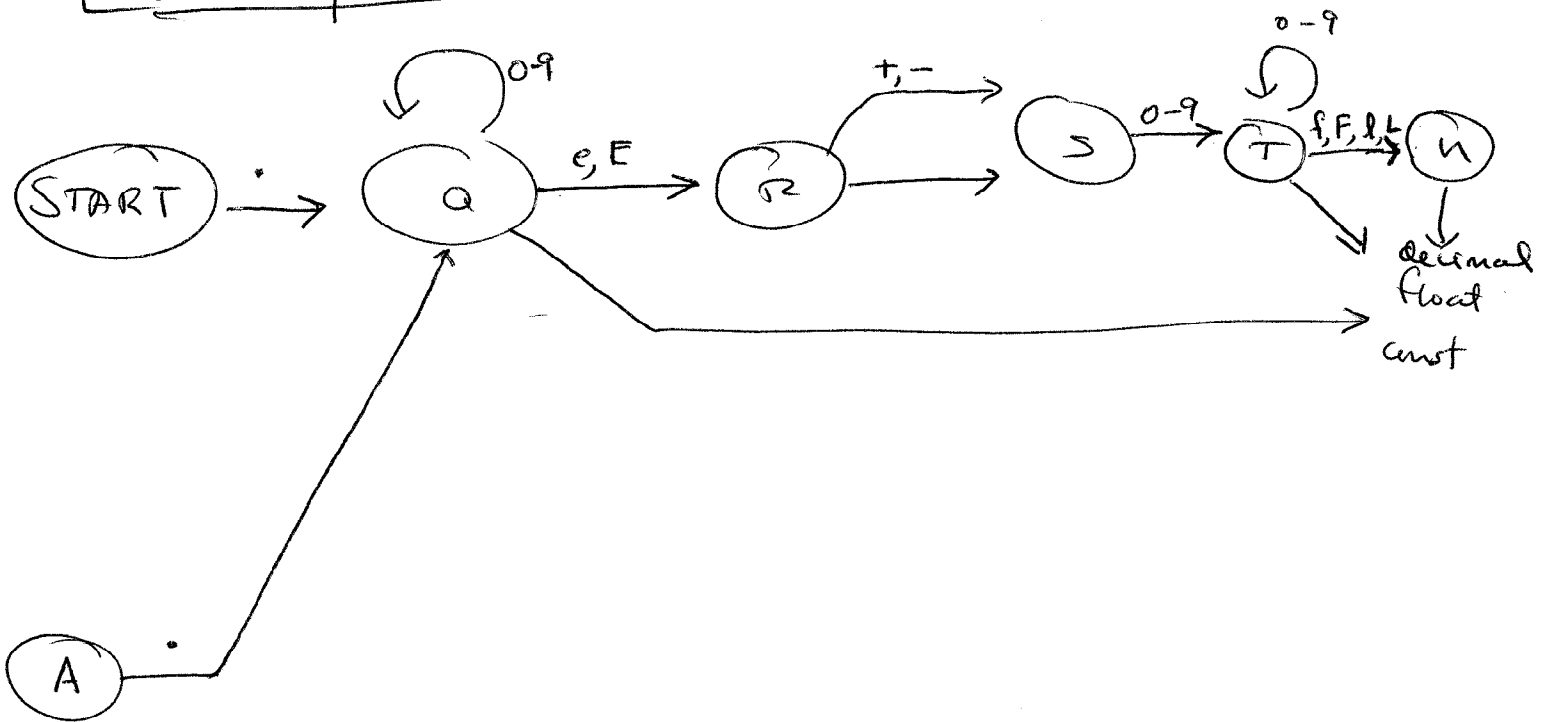


# ANSI C Integer Literals

①



To accommodate floating constants add the following:



② (a) The grammar's ambiguity can be proven by finding an example of a sentence generated by the grammar in two or more distinct derivations (parse trees)

Consider the statement

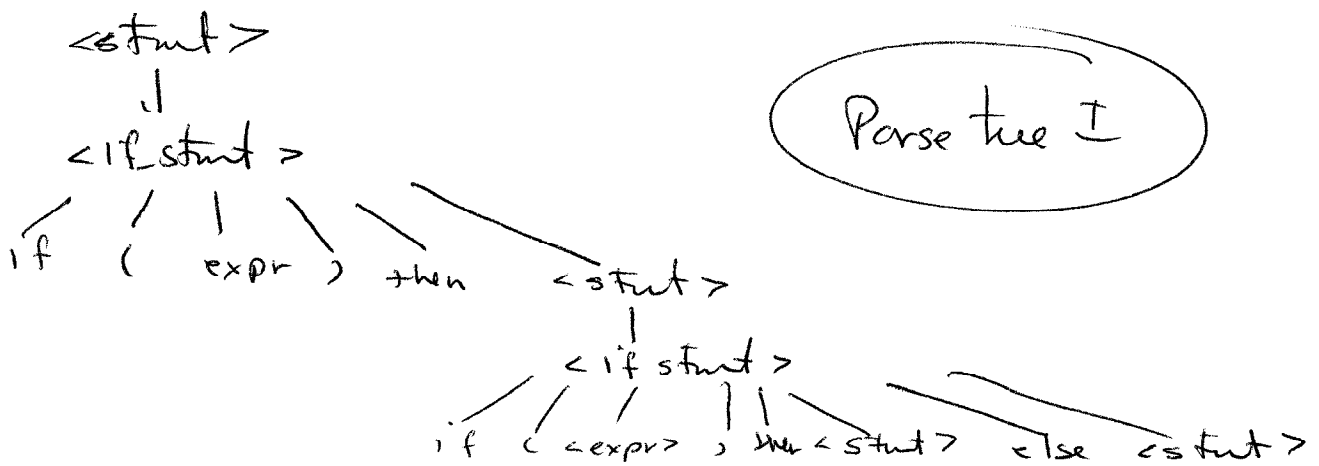
if (<expr>) then

if (<expr>) then <stmt>

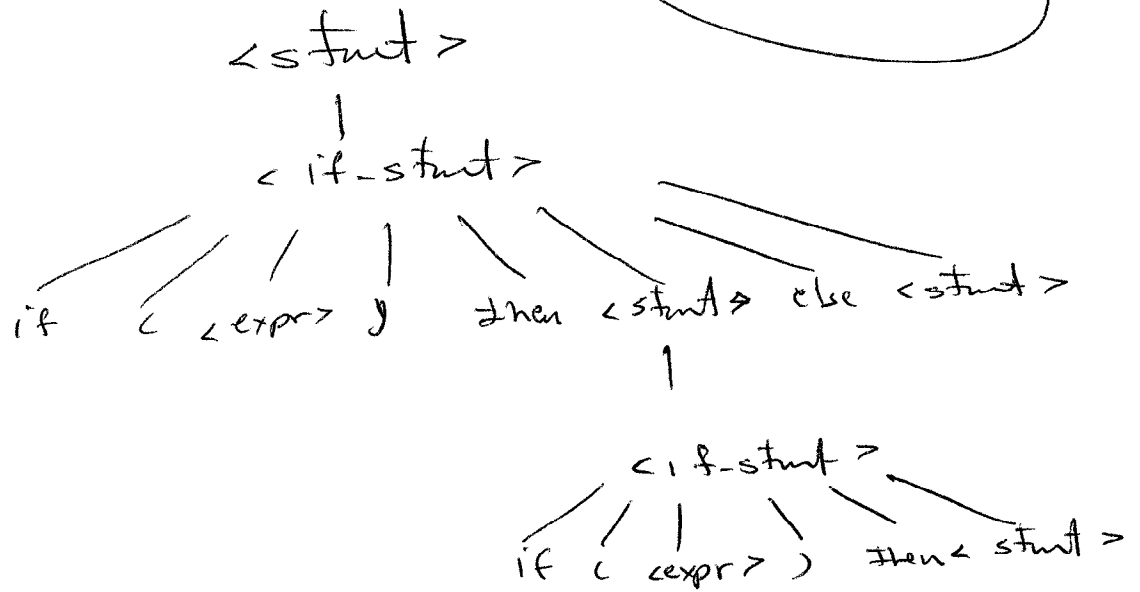
else

<stmt>

The following distinct parse trees are generated by the grammar



## Parse tree II



This well known ambiguity is commonly called the dangling else problem. PASCAL <sup>(and virtually all PL's)</sup> resolves the ambiguity by associating the "else" with the closest unmatched "then". Therefore, these languages prefer parse tree I.

We now provide a new grammar that only allows parse tree I.

stmt  $\rightarrow$  balanced\_stmt  
| unbalanced\_stmt

balanced\_stmt  $\rightarrow$  if (<expr>) then <balanced\_stmt> else <balanced\_stmt>  
| block  
| ...

unbalanced\_stmt  $\rightarrow$  if (<expr>) then <stmt>

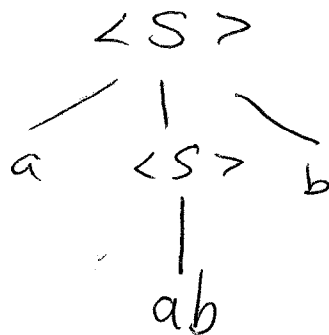
| if (<expr>) then <balanced\_stmt>  
else <unbalanced\_stmt>

③ Sebesta 3.13

~~(a)~~  $\langle S \rangle \rightarrow a \langle S \rangle b \mid ab$

④ Sebesta 3.14

(a) Parse tree for aabb



(b) Parse tree for aaaaabbbb

