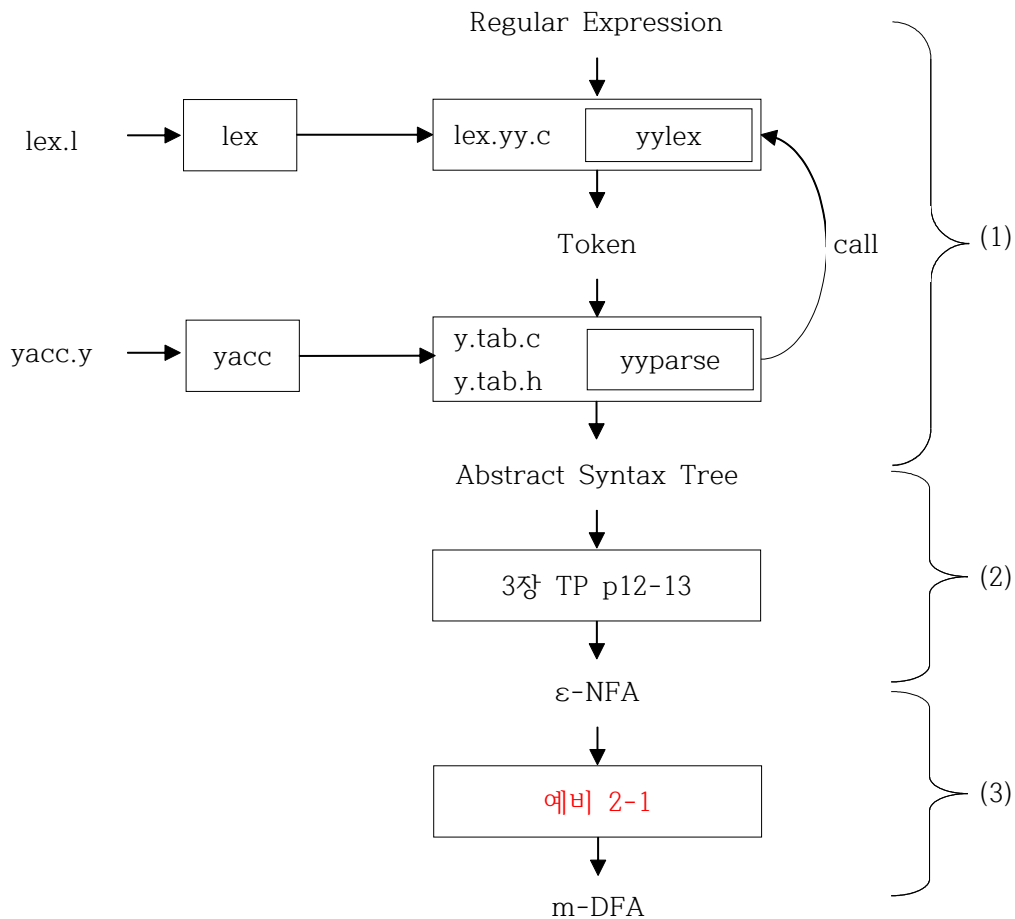


## 본 프로젝트 2 정규식 to m-DFA 변환기

- (1) lex와 yacc<sup>1)</sup>을 이용하여 정규식(regular expression)을 **파싱**(parsing)하며 Abstract Syntax Tree(AST)<sup>2)</sup>를 만든다.
- (2) AST를 지나며(traverse)  $\epsilon$ -NFA를 만든다<sup>3)</sup>.
- (3) **예비프로젝트 2-1의  $\epsilon$ -NFA to m-DFA 변환기를 이용하여**  $\epsilon$ -NFA를 m-DFA로 변환한다.




---

1) lex와 yacc은 UNIX에서 제공된다. 비슷한 일을 하는 flex, bison등을 다른 것을 사용하여도 좋다.  
 2) AST는 parse tree의 부분집합으로 Leaf는  $\epsilon$ ,  $\emptyset$ ,  $a \in T$ 가 있고, Union과 concatenation은 두 개의 subtree를 star는 한 개의 subtree를 가진다.  
 3) 교과서 TP 3장 Regular Expressions and Languages의 p. 12-13에 있는 방식으로 만든다.