

Assignment #1 (이자 마지막...)

아래 문제들을 해결한 후 풀이(해설)과 함께 이클래스를 통해 제출하세요.

- 즉, 소스파일 + 레포트가 제출되면 됩니다, 압축해서 올려주세요.
- 레포트 양식은 자유. 단, 너무 불필요하게 길지는 않게!

- 레포트의 파일 형식은 *pdf*로 제출하셔야 합니다.

제발제발제발제발

- 레포트의 파일 형식은 *pdf*로 제출하셔야 합니다.

제발제발제발제발

- 레포트의 파일 형식은 *pdf*로 제출하셔야 합니다.

pdf아니면 제출 안한거로 간주하겠음!!!!

(세번이나 반복해서 씌는데, 설마 또 hwp일까...)

- 코딩은 큰 주제와 맥락을 벗어나지 않는 선에서, 자유롭게. 창의롭게.
- 첨부해드린 예제 코드에 아직 배우지 않은 vector 클래스가 있습니다. 일단 애는 그냥 조금 편리한 배열 이라고 생각하시면 되겠습니다!

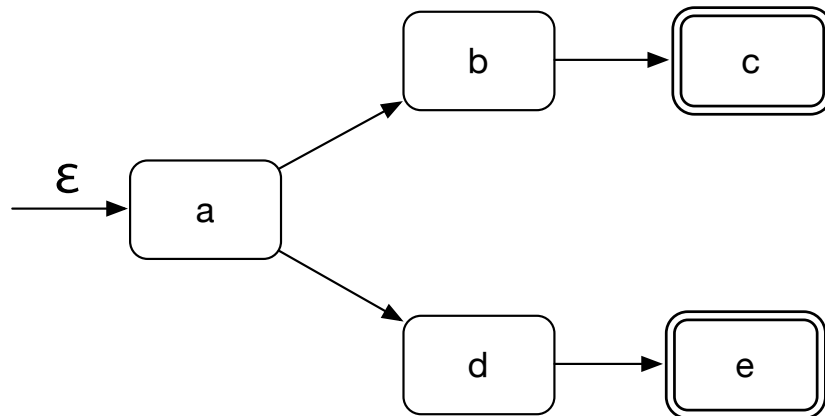
기한: 2023년 12월 22일 금요일 11:59AM 까지

#1. Regular expression and state machine.

- 정규 표현식은 문자열을 패턴으로 검색하기 위해 사용되는 아주 유용한 도구입니다. 한번도 본 적 없으시다구요? 아마 여러분들이 파일을 검색할때 "*.txt"와 같은 검색을 해보신적이 있을겁니다. "*.txt"는 파일 명이 무엇이 되든간에 .txt인 파일을 찾아달라는 의미이죠. 이게 정규표현식입니다!

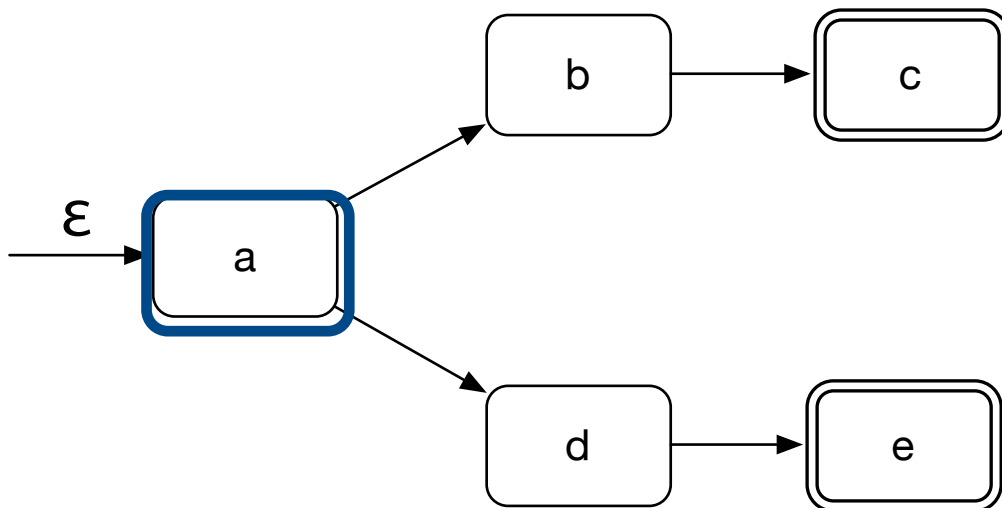
이러한 정규표현식의 매칭여부를 판단하기 위해서는 State machine (상태기계)이라는 것을 이용해 매칭하게됩니다. 상태 기계는 간단합니다. 어떤 노드가 현재 활성화가 되어있을 때, 조건이 만족되면 연결된 다음 노드에 상태를 전이(transition)하는 그런 것을 말합니다.

예를 들어, 아래 그림은 첨부해드린 regex.cpp에 구현된 상태머신을 그림으로 나타낸 것입니다.

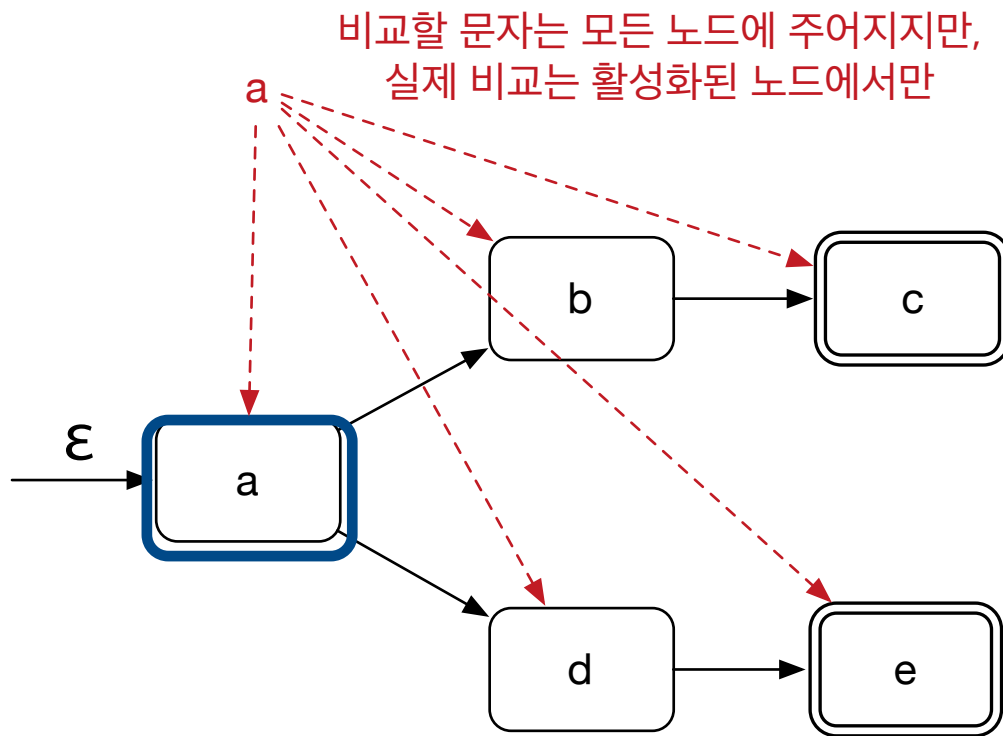


(ϵ (Epsilon signal)은 최초의 노드에 대해 항상 활성화를 시켜주는 그런 애이며, 동그라미가 두겹으로 된 c와 e는 최종 매칭 판단 여부를 나타내는 terminal state 입니다.)

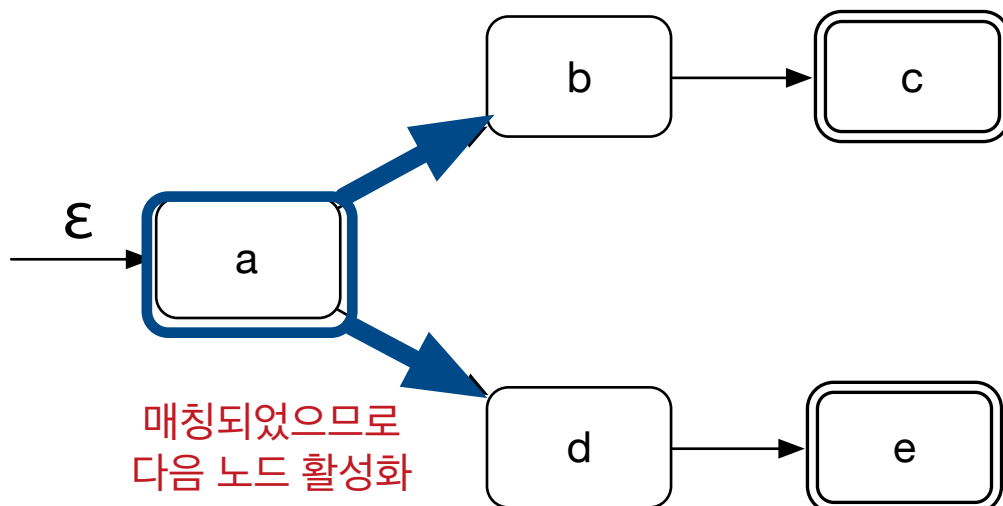
요 상태머신은 abc 또는 ade라는 문자열을 탐지하는 그런 상태머신입니다. 코드의 test1 (즉, abc)을 매칭을 진행한다고 할 때, 우선 아무런 입력이 주어지지 않더라도 a state는 입실론 신호에 의해 아래처럼 항상 활성화가 되어 있게됩니다. (코드에서는 강제로 항상 활성화 신호를 발생시키는 것으로 대신했습니다)



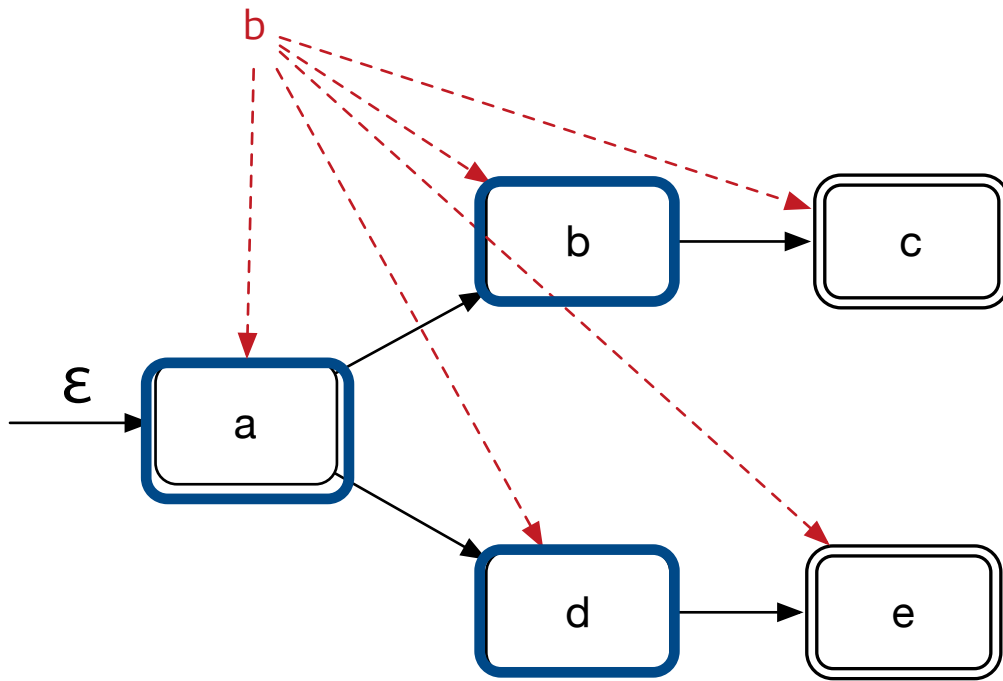
상태가 활성화 된 노드는 주어진 입력값에 대해 비교를 수행할 준비가 된 상태입니다. 이때, abc 중 첫번째 글자 a가 입력으로 주어집니다. 이 입력은 모든 노드로 주어지지만, 비교는 활성화된 a 노드에서만 이뤄지게 됩니다



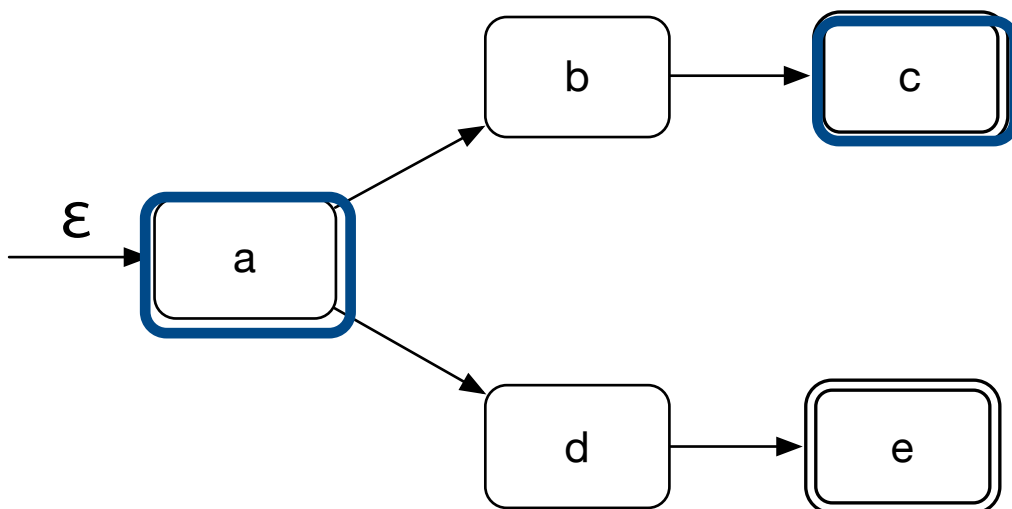
현재 활성화된 노드 a는 입력 문자열 a와 일치하므로, 연결된 다음 노드에 신호 전이를 발생시키며 활성화 시킵니다. 그리고 자신 역시 비활성화가 되어야 하지만, a는 입실론 신호에 의해 계속 활성화 상태가 유지됩니다.



이제, 다음 알파벳인 b가 입력될 차례입니다. 현재 활성화된 노드는 a, b, d입니다. 역시 b가 모든 노드로 전달되지만, 매칭은 a, b, d 세개의 노드에서만 일어납니다



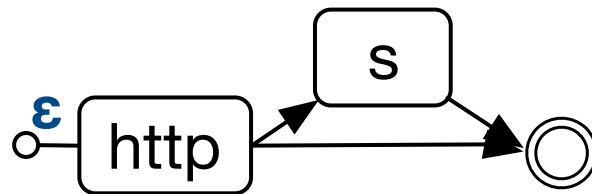
b노드가 입력된 b와 일치하네요. 따라서 다음 c로 상태 전이를 발생시키고 자신은 비활성화 됩니다. 매칭에 실패한 a, d 역시 비활성화가 됩니다. (사실 a도 비활성화가 되긴 하지만, 입실론에 의해 계속 활성화 상태가 유지되니 상태가 되는겁니다). 정리하면 다음과 같은 상태가 되겠네요.



마지막으로, 입력 c가 들어옵니다. c 노드가 활성화 된 상태이니 c에서 비교가 이뤄질꺼고, 이에 대한 일치가 됩니다. c는 terminal 노드로서 여기서 일치가 되면 최종적으로 매칭이 된 것을 의미합니다 (이를 Accept 되었다 라고 합니다). 따라서, 입력 abc에 대해 탐지를 할 수 있게 되는거죠.

요로한 매칭이 반복되어 아무리 긴 문자가 입력(test3 와 같이) 이 되더라도, 중간에 필요한 패턴을 탐지해서 추출할 수 있게 됩니다! 한번 다양한 입력을 주며 테스트 해보셔도 좋습니다. (이상의 예제는 상태기계중에서 NFA(Non-deterministic Finite Automata)라는 녀석입니다. 친구로는 DFA라는 녀석도 있는데, 요고에 대해서는 다음에 다른 수업에서 배우기로 하죠.)

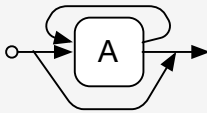
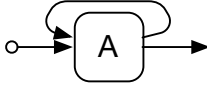
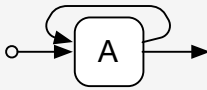
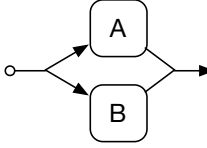
어쨌든 ,정규표현식은 이러한 상태머신을 연결/조합하여 패턴의 일치 여부를 판단하게 됩니다 예를 들어, 정규표현식 "HTTPS{0,1}"의 의미는 HTTP 매칭 이후, S가 0~1회 존재 라는 의미로, 즉 HTTP 또는 HTTPS 의 매칭을 판단하는 겁니다. 이를 State machine을 통해 나타낸다면, 아래와 같이 표현이 됩니다. (동그라미 두개는 accept되었다라는 신호)



그래서, 본 과제의 목적은 '정규표현식 처리기'를 만들어보는겁니다. 대략적인 상태머신을 위한 node는 만들어드렸으니, 정규표현식을 파싱하여 이들의 연결을 만들어내고, 검사까지 하는 코드를 만들어 보도록 하죠!

대신 간략화하기 위해, 검사해야할 패턴속에 특수문자나 숫자는 없이 오로지 단일 알파벳과 정규표현식 syntax들로만 구성되어있다고 가정하겠습니다. 괄호를 통한 우선순위의 상승도 없구요! 그리고 정규표현식 문법(Syntax)은 다음과 같이 . (any character), * (0회 이상), + (1회 이상), {m, n} (m회 이상, n회 이하), A|B (A 또는 B) 만 지원한다고 가정합니다. 아래 각 정규표현식 syntax들이 오토마타 상으로 어떻게 연결되어야 하는지 예제 그림도 있습니다.

Syntax	의미	매칭되는 패턴	NFA 표현식	비고
.	아무 문자	A B C D+의 의미는 아무 문자가 1회 이상

Syntax	의미	매칭되는 패턴	NFA 표현식	비고
A^*	A가 0회 이상 반복	A AA AAA AAAA...		
A^+	A가 1회 이상 반복	A AA AAA AAAA...		
$A\{m, n\}$	A가 m회 이상, n회 이하 반복	만약 {3, 5}라면, AAA AAAA AAAAA	$m \leq A \leq n$ 	내부 카운터가 필요
$A B$	A 또는 B	A B		

참고로 정규표현식을 파싱을 하기 위해서는 보통 후위표기법(postfix)로 변환하는 과정을 포함하는데.... 본 과제에서 정한 정규식의 조건이라면, 굳이 필요는 없을 듯 합니다. 딱히 괄호로 인한 우선순위의 변동이 없으니 그냥 죽 처리하면 되겠네요.

< 요약 >

- **Step 1)** 정규표현식을 처리기를 만들어주세요.
- **Step 2)** 정규표현식 처리기란, **임의의 정규식**을 입력받아 그에 대한 상태머신을 생성하는 것을 말합니다.
- **Step 3)** 생성된 상태머신에서 입력값에 대한 테스트가 가능하여야 하며, 패턴의 일치 여부 역시 판단할 수 있어야 합니다.
- **Step 4)** 아마 주어진 샘플 코드에 일부 확장이 필요할 수 있습니다
- **기대 결과물)** 내용이 반영된 regex.cpp, 이에 대한 설명 또는 레포트
- **사족)** 정규표현식에 대한 내용은 구글에 정말 많습니다! 잘 확인해보세요 :)