어휘분석기

String → tokens

Token

1. 어휘 분석 결과는 어떻게 출력되는가?
2. 토큰이 어떤 유형이 있는가?

1. ex) if (a ≥ 10)

Token number	32	7	4	25	5	8
Token value	0	0	'a'	0	10	0

2.

special form

keyword if while for else

operator symbol + - × < >

Delimiter , ; ( ) [ ]

정규표현 필요 ×, 변환할 필요 없음

General form - Programmer

Identifier 변수명

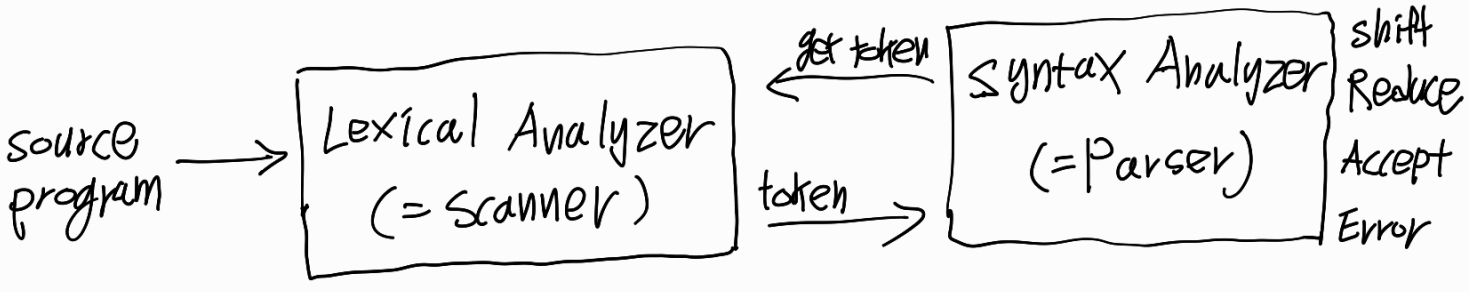
Constant 상수

정규표현으로 기술

ex) id = (L+_)(L+d+_)*

그래서 어휘 분석은 왜 하는가?

구문 분석기 (Syntax Analyzer, parser) 에 token을 넘겨주기 위해서이다



어휘분석과 파싱단계는 더욱 쉽고 효율적이기 위해 서로 구분한다

general token의 인식

1. Token description → R.E , F.A

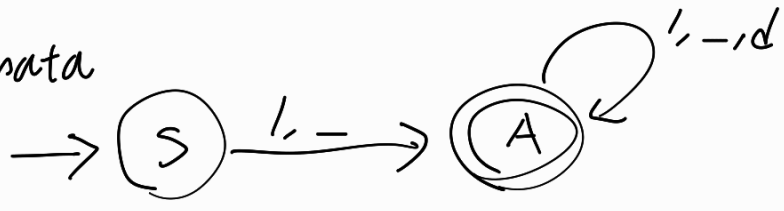
2. develop a scanner

3. verify throug regular Language theory

목표는 정규표현을 얻는 것이다

명칭

Automata



Regular grammer

$S \rightarrow lA \mid \_A$

$A \rightarrow lA \mid \_A \mid dA \mid \varepsilon$

Regular expression

$S = lA + \_A = (l + \_)A$

$A = lA + \_A + dA + \varepsilon = (l + \_ + d)A + \varepsilon$

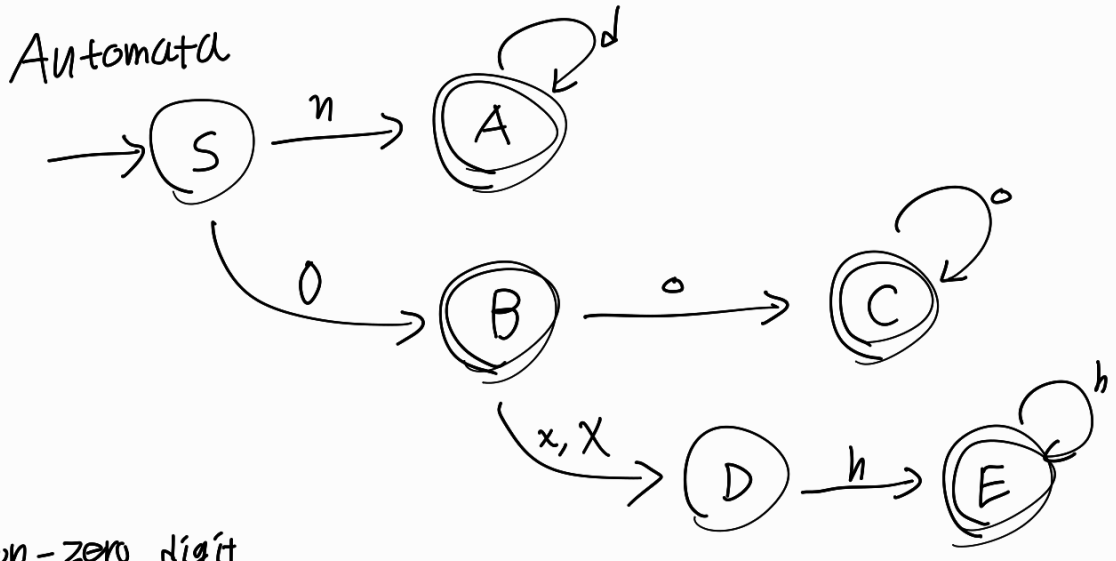
$= (l + \_ + d)^*$

$\therefore S = (l + \_)(l + \_ + d)^*$

정수

10진수 8진수 16진수를 모두 표현해야 한다

Automata



n : non-zero digit

o : 0 ~ 7

h : 0 ~ F

Regular grammer

$S \rightarrow nA \mid 0B$ $\quad A \rightarrow dA \mid \varepsilon$

$B \rightarrow oC \mid xD \mid \varepsilon$ $\quad C \rightarrow oC \mid \varepsilon$

$D \rightarrow hE$ $\quad E \rightarrow hE \mid \varepsilon$

Regular expression

$$S = nd^* + 0 + 0o^+ + 0(x + X)h^*$$

실수

Automata

regular grammer

regular expression

String 상수

Automata

regular grammer

regular expression

Comment

Automata

regular grammer

regular expression

이제 얻은 정규표현을 통해서 어휘분석기를 개발할수 있다

1. 직접개발

2. 도구사용 (lex)

도구사용

Lex 입력 파일 구조

정의부

%%

*규칙부

%%

사용자 서브루틴

정의부

format

name translation

ex)

D [0-9]

L [a-zA-Z]

%%

또다른 기능

%{ ... %}

C언어의 선언문들 #include
전역변수 등...

규칙부

format

regular expression + actions

ex) integer
color C언어 코드
[0-9]+

사용자 서브루틴

input file에 사용되는 subprogram 정의