내용복습. (3장)

Syntax Analysis. : $\boxed{\text{BNF}}$: <meta language>.

↳ Rule들의 집합 = Grammer

Terminal, (lexeme)

Non-terminal 구성 (start symbol → derivation된다.

← Context Free하다.

↕

Context Sensitive.

→ 단어 문장이 문장에 다르게 해석할 수 있다.

↳ 문맥을 이해해야 한다.

② precedence : 우선순위.

Associativity : 결합법칙. → 하나의 식에 같은 등급의 같은 precedence 연산 순서 결정.

원/오 부터 계산할 지 계산

대부분 왼쪽 (left Associativity)

** 같은건 (right Associativity)

③ Brace를 한다.

if { then } else

↳ 가장 가까운 then 과 매칭을 한다.

바로 앞 then과 처리되고 싶으면 중괄호 (brace) 를 한다.

④ EBNF : [ ] : Zero or Once.

{ } : Zero or more

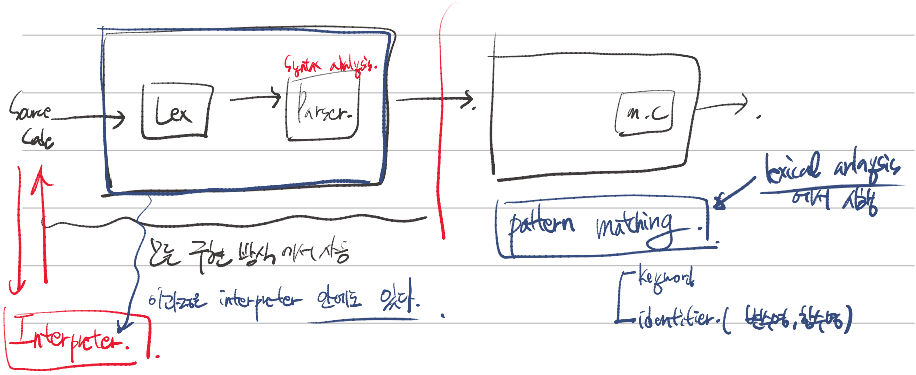
{ }+ : One or More.

재귀적인 표현이 있을 때 유용. (recursive)

↳ '축약' 하기 위해 사용

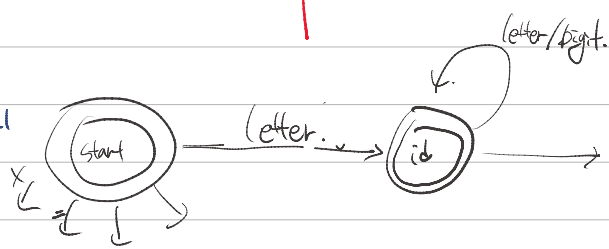
4.1 lexical analysis 와 Syntax analysis 는 모든 언어에 들어간다.

└ 부석 후처리



- identifier는 symbol table에 정리해둔다.
- Lex 에서 result = ← 하나의 lexeme.
  - → lexeme 를 token으로 분리.
  - → identifier 에 = 가 없기 때문에 분리.

그림 4.1



4.4. <expr> → <term> {(+|-)<term>}.

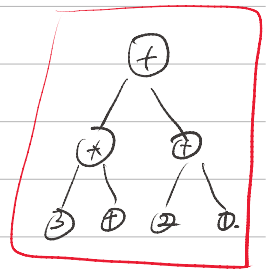
처음에는 Start Symbol 함수로 호출한다.

즉, non-terminal 은 함수를 호출한다. 생각하라.

→ 대문자 B (non-terminal)

소문자 a (terminal)

Top Down - Preorder.



트리방문순서 : preorder inorder postorder

연산자 : prefix. infix. postfix.

+34 3+4 3++.