A(Lisp-Like) TOY Language [Version 1]

toy언어를 배우는 목적 4~5가지

1. 언어가 어떻게 만들어지고 어떻게 확장될 수 있나.

2. 언어의 문법을 어떻게 표현하느나 ex) BNF

3. parsing은 어떻게 되느냐

4. 간단한 중간코드를 생성해서 인터프리터가 해결하는 방법

FUN = {MINUS, IF} 함수 2개 \* version 1에서는 함수가 2가지 밖에 없음

Z x Z to Z 🡪 정수 값 2개 받고 정수를 리턴함

term의 종류 3가지

1. constant ex) -3, -2, -1, 0 , 1, 2…

2. variable ex) v1, v2, identifiers

3. 함수를 이용한 term ex) ( MINUS ( MINUS 1 2 ) ( IF 3 4 ) )도 가능

TOY언어의 문법

<TOY> 🡪 <term>

<term> 🡪 <var>

| <const>

| (MINUS <term> <term>)

| (IF <term> <term>)

<var> 🡪 a | b | c | ….. | x | y | …

<const> 🡪 …| -3 | -2 | -1 | 0 | 1 | 2 | …

interpreter가 toy 프로그램 value 값을 계산

VALUE : TOY 🡪 Z

% 🡪 undefined(모른다)

% 3 🡪 3

% -4 🡪 -4

% (MINUS 3 4) 🡪 -1

% (IF 1 2) 🡪 2 // 양수이면 값은 true(2번쨰 term)

% (IF -2 3) 🡪 0 // 0이거나 음수이면 값은 false(0)

% (MINUS (MINUS 1 2) (IF 3 4)) 🡪 -5

//MINUS -> M

(M (IF 1 2) (M (IF -3 4) 5) ) 🡪 7

(M (IF 1 2) (M (IF -3 4) 5) ) 🡪 7

prefix 방식 – root left right

infix 방식 – left root right

postfix 방식 – left right root

(M (IF 1 2) (M (IF -3 4) 5) ) 🡪 7

중간code 생성 postifx방식

1 2 IF -3 4 IF 5 M M

push 1 -> push 2 -> IF -> push -3 -> push 4 -> IF -> push 5 -> M -> M

책 5.1

사용하는 프로그래밍 언어는 폰 노이만 아키텍처에 기초한다.

5.3

valiable 6가지 특성

name, address, value, type, lifetime(메모리에 할당 받는 시간), scope

5.3.2

x = y + 2

왼쪽은 x의 주소가 필요한 l-value 오른쪽은 값이 필요한 r-value