

## 프로그래밍 언어 개론 hw item3

학번/이름: 201404376 정성욱

201502019 김기범

201701996 김주성

분반: 00반 조은선 교수님 분반

### Define 함수부분 구현 방법

디파인식 함수정의는 일반 매개변수 정의와는 상당히 다른 부분이 있으므로, DF\_table외에 DF\_Func\_table을 정의해 주었다. 설명 문자열로 정의하고 저장 한다 하더라도 저장 시그 문자열에 해당 되게 매개변수와 매개변수를 나타내는 문자열을 스위칭 시켜줘야한다. 그리고 처음 입력을 받을 때, 함수의 바디 부분을 받는다. 가장 중요한 연산 부분이 바디 부분이다. 선언이후 선언된 것을 사용하는 구문이 있을 경우 runlist를 건들여서 아예 리스트 파싱을 할 때, 현재 입력되는 함수의 이름이 테이블 안에 존재한다면, 함수를 실행하는 시퀀스로 진입하여,

들어온 값의 매개변수에 맞게 처리하도록 하였다. 여기서는 최대 2개의 매개변수를 가지고 1개의 매개변수를 가질 경우에, 선언 부분에 저장해둔 테이블에 있는 연산을 위한 상수 값을 불러와서 처리한다. 이것이 앞에 위치할수도 뒤에 위치할 수도 있기 때문에 노드를 3개를 둔 것이다.

### Lambda 구현 방법

람다식에 실 매개변수가 주어지지 않는다면 입력된 람다식을 그대로 출력해줘야 한다.

case LAMBDA: 부분에서 처리가 되어야 하며 oprator 부분과 oprand를 합쳐서 그대로 들어온 람다식이 출력되도록 하였다.

만약 실 매개변수가 주어진 경우라면 람다식과 실 매개변수를 포함하는 리스트가 들어온다

즉 첫번째 노드가 function 노드가 아니라 list 노드가 되는 것이다.

이 경우에 대해서 처리하는 부분이 runList() 함수에는 구현이 되어 있지 않기 때문에 조건문을 통해 ListNode에 대해서 처리를 해주도록 한다.

실 매개변수를 형식 매개변수에 바인딩해주어야 하는데, 바인딩 하는 작업을 varBind() 함수를 만들어서

처리하도록 구현하였다. varBind() 함수는 빈 리스트가 나올 때까지 반복하면서 형식 매개변수의 이름을 키 값으로해서

실 매개변수를 변수 테이블에 저장해주는 함수이다.

varBind를 통해서 변수의 바인딩이 끝났으면 연산 식이 오는 부분을 다시 runExpr() 함수를 통해서 실행시켜주면

자연스럽게 바인딩 된 변수가 불러오면서 연산이 되게된다.

변수 테이블에 할당된 값들은 연산이 끝나면 제거가 되어야 하는데 이것은 매 반복마다 테이블을 초기화 하는 것으로 처리하였다.

구현결과 :

```
$ ( define plus1 ( lambda ( x ) ( + x 1 ) ) )  
...> ( PLUS x 1 )  
$ ( plus1 3 )  
...> 4  
$
```