

PA3_OCaml F1VAE Interpreter REPORT

2020311067 이동국

F1VAE는 First-order Function and VAE로서 프로그램을 실행할 때, 함수를 선언하는 `decl`과 계산 또는 `letIn`, 변수, 함수 호출 등을 표현하는 `expr`로 구성할 수 있다.

OCaml은 Functional programming language로서 다른 Imperative와 다르게 F1VAE를 언어로 표현할 때 강력한 수단(이해하기 쉽고 간단하게 표현)이 될 수 있다.

*concrete syntax

<i>prog</i>	<i>::=</i>	<i>decl expr</i>
<i>decl</i>	<i>::=</i>	<i>def var var = expr endef</i>
<i>expr</i>	<i>::=</i>	<i>let var = expr in expr</i>
		<i>var(expr)</i>
		<i>expr + expr</i>
		<i>expr - expr</i>
		<i>(expr)</i>
		<i>number</i>
		<i>~ (expr)</i>
		<i>var</i>

1) interp_expr 함수

type expr = Num of int

| Add of expr * expr

| Sub of expr * expr

| Id of string

| LetIn of string * expr * expr

| Call of string * expr list에 대해 해석하는 함수.

Num of int 형태를 가질 경우 value.ml에 정의된 NumV of int를 이용해 정수값을 그대로 반환한다.

Add of expr * expr, Sub of expr * expr 형태를 가질 경우 각 expr에 대한 interp_expr 함수를 재귀적으로 호출하여 NumV n인 정수값을 얻고, 얻은 두 expr의 값들을 더하거나 뺀 값을 반환한다.

Id of string 형태를 가질 경우 store.ml의 Value.t M.t 형태로 저장되어있는 string이 있는지 찾고 없으면 free identifier이기 때문에 error를 출력하고, 있다면 해당 string과 함께 포함된 정수값을 반환한다.

LetIn of string * expr * expr은 Let string = expr in expr을 처리하는 형태로서 let string = expr으로 정의된 부분을 로컬로 store.ml에 저장하고 in 뒤에 있는 expr 값을 계산하여 반환한다.

Call of string * expr list는 fStore.ml의 (String list * Ast.expr) M.t 형태로 저장되어있는 함수가 있는지 찾고 없으면 Undefined function이라는 error를 출력하고, 있다면 parameter의 수와 argument의 수를 비교한다. 만약 서로 다른 수면 필요한 parameter의 수와 입력된 argument의 수를 보여주며 error를 출력한다. 같은 수면 parameter와 argument에 대해 재귀적으로 조사하는 helper를 이용해 남아있는 것이 있으면 현재의 것들을 바인딩(store.ml에 저장)하고 다음 것을 처리한다. 모든 parameter 및 argument가 처리되면 env를 반환하도록 한다. 이를 이용해 새로운 환경에서 함수 본문을 실행한다.

2) interp_fundef 함수

type fundef = FunDef of string * string list * expr에 대해 해석하는 함수.

function 정의에 따라 저장해야 하는 function 이름과, parameter, body 부분을 fStore.ml에 추가한다. 함수 정의 이후 인자가 매개변수의 수와 일치하고 함수가 일치하면 expr 내에서 언제든지 사용이 가능하다.

3) interp 함수

type prog = Prog of fundef list * expr에 대해 해석하는 함수.

prog를 받아 그 안의 함수를 정의하는 fundefs 부분과 expr 부분을 추출한다.

fundefs에서 함수 정의한 부분을 global_env로 구축하며, List.fold_left를 이용해 각 함수에 대해 interp_fundef 함수를 적용하여 글로벌 환경을 만든다. 초기값은 fStore와 Store가 비어있어야 한다. 마지막으로 expr를 해석하여 이전에 구축한 글로벌 환경 global_env를 사용하고 주요 표현식을 만들어 최종적으로 Value.t 타입의 결과를 반환한다.