Homework 1

SNU 4541.664A

Due: 3/24, 24:00(program), in-class(written)

Kwangkeun Yi

- 프로그램 숙제는 OCaml로 작성하는 것을 권장합니다. 다른 언어로 구현할 경우, 숙제의 spec해 해당하는 것을 해당언어로 번역해서 제출하세요.
- 비 프로그램 숙제는 수업시간에 제출하세요.

Exercise 1 집합 $T \ni t$는 귀납적으로 다음과 같이 정의된다:

$$t \to \cdot \mid /t,t/ \mid /t,t,t/$$

모든 $t \in T$는 ,와 /의 갯수에 대한 어떤 성질을 만족한다. 그 성질을 찾고 증명하라. □

Exercise 2 식들의 집합이 귀납적으로 다음과 같이 정의된다:

$$e \to x \mid e + e \mid e * e \mid e?\, e\, e$$

"+"와 "*"는 각각 정수 더하기와 곱하기를 뜻하고 "$e_1?\, e_2\, e_3$"은 e_1의 값이 0이면 e_3의 값을, 아니면 e_2의 값을 계산한다.

다음을 증명하라: 모든 식에 대해서, 그 식에 나타나는 변수들의 값이 n의 배수이면 그 식의 값은 n의 배수이다. □

Exercise 3 CPO D위의 연속함수 f

$$f \in D \to D$$

의 최소 고정점이

$$\bigsqcup_{i \geq 0} f^i \bot$$

임을 증명하라. □

Exercise 4 CPO A에서 CPO B로 가는 연속함수로 구성된 $A \to B$가 CPO임을 증명하라. □

Exercise 5 다음 식들의 고정점을 찾아라:

- $\lambda x.1 \quad \in \mathbb{Z} \to \mathbb{Z}$
- $\lambda x.x \quad \in \mathbb{Z} \to \mathbb{Z}$
- $\lambda x.x + 1 \quad \in \mathbb{Z} \cup \{\infty\} \to \mathbb{Z} \cup \{\infty\}$
- $\lambda f(\lambda x.\text{if } x = 0\text{? } 0 : \ x + f(x-1)) \quad \in (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$
- $\lambda X.\{\epsilon\} \cup \{ax \mid x \in X\} \quad \in 2^S \to 2^S$ where S is the set of finite strings.

□

Exercise 6 Given a graph $G = (N, E)$ (N is the set of nodes, $E \subseteq N \times N$ the set of edges between the nodes), the reachable set $reach_G(X)$ of nodes from the initial node set X can be defined as the least fixpoint of a function.

The $reach_G(X)$ is the smallest set S that satisfies

- $X \subseteq S$
- If $x \in S$ then $\{y \mid x \to y \ \in E\} \subseteq S$.

Fill out the hole in the following definition:

$$reach_G(X) = lfp(\lambda S.\square)$$

□

Exercise 7 수업시간에 다룬 간단한 명령형 언어인 `KMINUS.cmd` 타입의 프로그램을 받아서 아래의 의미식 타입 `SEM.exp`의 식으로 변환하는 프로그램

$$\texttt{semantics} : \texttt{KMINUS.cmd} \to \texttt{SEM.exp}$$

을 작성하라:

```
module type KMINUS =
sig
  exception Error of string
  type id = string
  type exp = NUM of int
           | VAR of id
           | ADD of exp * exp
           | MINUS of exp
  type cmd = SKIP
           | SEQ of cmd * cmd             (* sequence *)
           | IF of exp * cmd * cmd        (* if-then-else *)
           | ASSIGN of id * exp           (* assign to variable *)
           | WHILE of exp * cmd           (* while loop *)
end

module type SEM =
sig
     type id = string
     type exp = VAR of id
              | NUM of int
              | LAM of id * exp          /* continuous function */
              | APP of exp * exp         /* function application */
              | IF of exp * exp * exp    /* if-then-else operator */
              | FIX of exp               /* fix operator */
              | NOTEQZ of exp            /* not-equal-zero operator */
              | ADD of exp * exp         /* binary + operator */
              | MINUS of exp             /* unary - operator */
              | UPDATE of exp * id * exp /* function update operator */
end
```

□