Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Fabian Huch

WS 2022/23
17.01.2023

Semantics of Programming Languages

Exercise Sheet 11

Exercise 11.1 Complete Lattice over Lists

Show that lists of the same length – ordered point-wise – form a partial order if the element type is partially ordered. Partial orders are predefined as the type class *order*.

instantiation *list* :: (*order*) *order*

Define the infimum operation for a set of lists. The first parameter is the length of the result list.

definition *Inf_list* :: *"nat ⇒ ('a::complete_lattice) list set ⇒ 'a list"*

Show that your ordering and the infimum operation indeed form a complete lattice:

interpretation
Complete_Lattice "{xs. length xs = n}" "Inf_list n" **for** *n*

Exercise 11.2 Fixed Point Theory

Let $'a$ be a complete lattice with ordering $\leq$ and $f{::}'a \Rightarrow 'a$ be a monotonic function. Moreover, let x_0 be a post-fixpoint of f, i.e., $x_0 \leq f\, x_0$. Prove:

$$\bigsqcup \{f^i(x_0) \mid i \in \mathbb{N}\} \leq \bigsqcup \{f^{i+1}\,(x_0) \mid i \in \mathbb{N}\}$$

Hint: The least upper bound satisfies the following properties

$$x \in A \Longrightarrow x \leq \bigsqcup A \qquad \text{(Sup\_upper)}$$

$$(\forall\, x \in A.\ x \leq u) \Longrightarrow \bigsqcup A \leq u \qquad \text{(Sup\_least)}$$

unbundle *lattice_syntax*

lemma
  fixes *f* :: *"'a::complete_lattice ⇒ 'a"*
  assumes *"$x_0 \leq f\, x_0$"*
  shows *"$\bigsqcup\ \{(f\widehat{\ }\widehat{\ }i)\ x_0 \mid i.\ i \in \mathbb{N}\} \leq \bigsqcup\ \{(f\widehat{\ }\widehat{\ }(i+1))\ x_0 \mid i.\ i \in \mathbb{N}\}$"*

Homework 11.1 Collecting Semantics

Submission until Monday, Jan 23, 23:59pm.

Consider the version of IMP with LOOP c UNTIL b construct. The annotations of that construct can be inferred from the following annotated command:

```
|  x := 1 {A0};
|  y := 1 {A1};
|  {A2} LOOP
|    y := y - 2; {A3}
|    x := x + y {A4}
|    {A5}
|  UNTIL x < 0
|  {A6}
```

Compute the collecting semantics: Show how the annotations change with each application of the step function, until you reach a fix-point.

Write down all entries for each column. Use the explicit ____________ value where possible. Write down states as (x,y) tuple.

definition A_0 :: *"entry list"*
definition A_1 :: *"entry list"*
definition A_2 :: *"entry list"*
definition A_3 :: *"entry list"*
definition A_4 :: *"entry list"*
definition A_5 :: *"entry list"*
definition A_6 :: *"entry list"*

Homework 11.2 Kleene fixed point theorem

Submission until Monday, Jan 23, 23:59pm. Prove the Kleene fixed point theorem. We first introduce some auxiliary definitions:

A chain is a set such that any two elements are comparable. For the purposes of the Kleene fixed-point theorem, it is sufficient to consider only countable chains. It is easiest to formalize these as ascending sequences. (We can obtain the corresponding set using the function *range* :: $('a \Rightarrow 'b) \Rightarrow 'b\ set$.)

$chain\ C = (\forall n.\ C\ n \leq C\ (Suc\ n))$

A function is continuous, if it commutes with least upper bounds of chains:

$continuous\ f = (\forall C.\ chain\ C \longrightarrow f\ (\bigsqcup\ range\ C) = \bigsqcup\ (f\ `\ range\ C))$

The following lemma may be handy:

$[\![continuous\ f;\ chain\ C]\!] \Longrightarrow f\ (\bigsqcup\ range\ C) = \bigsqcup\ (f\ `\ range\ C)$

As warm-up, show that any continuous function is monotonic:

lemma *cont_imp_mono*:
  fixes *f* :: *"'a::complete_lattice ⇒ 'b::complete_lattice"*
  assumes *"continuous f"*
  shows *"mono f"*

Hint: The relevant lemmas are

thm *mono_def monoI monoD*

Finally show the Kleene fixed point theorem. Note that this theorem is important, as it provides a way to compute least fixed points by iteration.

In this proof, you may not use metis, meson, or smt!

theorem *kleene_lfp*:
  fixes *f* :: *"'a::complete_lattice ⇒ 'a"*
  assumes *CONT*: *"continuous f"*
  shows *"lfp f = ⨆ (range (λi. (f^^i) ⊥))"*
proof −

We propose a proof structure here, however, you may deviate from this and use your own proof structure:

  let *?C = "λi. (f^^i) ⊥"*
  note *MONO=cont_imp_mono[OF CONT]*
  have *CHAIN*: *"chain ?C"*

  show *?thesis*
  proof (*rule antisym*)
    show *"⨆ (range ?C) ≤ lfp f"*

  next
    show *"lfp f ≤ Sup (range ?C)"*
      qed
qed

Hint: Some relevant lemmas are

thm *lfp_unfold lfp_lowerbound Sup_subset_mono range_eqI*