

# 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**  $Inf\_list :: "nat \Rightarrow ('a :: complete\_lattice) list set \Rightarrow 'a list"$

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

**interpretation**  $list\_complete\_lattice :$

$Complete\_Lattice \text{ “}\{xs. length\ xs = n\}” \text{ “}Inf\_list\ n” \text{ for } n$

### Exercise 11.2 Collecting Semantics

Consider a version of IMP with a DO-WHILE construct, where the command is executed first and then the condition is checked for the next iteration.

```
| x := 2 {A0};  
| DO {A1}  
|   IF x < 3 THEN {A2} x := x * x {A3}  
|     ELSE {A4} x := y / x {A5}  
|   {A6}  
| WHILE 0 < x  
| {A7}
```

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

	0	1	2	3	4	5	6	7	8	9	10
A0											
A1											
A2											
A3											
A4											
A5											
A6											
A7											

Write the states down using sets of  $(x,y)$  tuples in explicit form, i.e. sets of the shape  $\{(x_1, y_2), (x_2, y_2), \dots\}$ . If an entry does not change, leave it empty.

### Homework 11.1 Collecting Semantics

*Submission until Wednesday, Jan 15, 23:59pm.*

Consider the following program in an extended version of IMP.

```

| x := 3 {A0};
| y := x^2 {A1};
| {A2}
| WHILE x + y > 6
| DO {A3}
|   y := y div x; {A4}
|   x := x + 1; {A5}
| {A6}
| y := x + y {A7}

```

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. If some value did not change, use the \_\_\_\_\_ (8 underscores) term. Write the states down using sets of  $(x,y)$  tuples in explicit form, i.e. sets of the shape  $\{(x_1, y_2), (x_2, y_2), \dots\}$ . The first annotation is pre-defined by us.

**definition**  $A_0$  :: “entry list” where “ $A_0 = [V\{\}, V\{(3,0)\},$  \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_]”

**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”

**definition**  $A_7$  :: “entry list”

## **Homework 11.2** Formalization of Formal Language Theory/Be Creative

*Submission until Wednesday, Jan 15, 23:59pm.*

The project deadline is extended to January 15. Submit your project as follows:

- If you were assigned a formal language project, simply push your changes to your assigned repository.
- If you chose your own project, submit your file to the submission system (exercise 9\_1). If you have to submit more than one file, send a zip-file to [semantics@proof.cit.tum.de](mailto:semantics@proof.cit.tum.de)