Solutions to Selected Exercises in Complement of the Book

Principles in Abstract Interpretation

MIT Press, 2021

Patrick Cousot

New York University

May 25, 2021

Contents

1 Solutions to Selected Exercises of Chapter 2

Solution to exercise 2.6 $\mathbb{N}$ is the smallest subset of $\mathbb{R}$ containing 0 and the successor of every natural that is $\mathbb{N} = \bigcap\{S \in \wp(\mathbb{R}) \mid 0 \in S \wedge \forall n \in S \,.\, n+1 \in S\}$. $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$, $\mathbb{Z} = \mathbb{N} \cup \{-n \mid n \in \mathbb{N}^+\}$. □

Solution to exercise 2.9 Take $S = \{a, b, c\}$, $r = \{\langle a,\ c\rangle, \langle b,\ c\rangle\}$ so $r^{-1} = \{\langle c,\ a\rangle, \langle c,\ b\rangle\}$ and $r \fatsemi r^{-1} = \{\langle a,\ a\rangle, \langle a,\ b\rangle, \langle b,\ b\rangle, \langle b,\ a\rangle\} \neq \mathbb{1}_S = \{\langle a,\ a\rangle, \langle b,\ b\rangle\}$. □

Solution to exercise 2.14 We have $!0 = 1$ by definition, so $!0 \in \mathbb{N}$. Assume, by induction hypothesis, that $!m \in \mathbb{N}$ for all $m < n+1$. Then $n < n+1$ so $!n \in \mathbb{N}$ by induction hypothesis and therefore $!(n+1) = (n+1) \times !n \in \mathbb{N}$ by definition of the factorial and $\times \in \mathbb{N}^2 \rightarrow \mathbb{N}$. By recurrence, $\forall n \in \mathbb{N} \,.\, !n \in \mathbb{N}$ so $! \in \mathbb{N} \rightarrow \mathbb{N}$. □

2 Solutions to Selected Exercises of Chapter 3

Solution to exercise 3.3

$$\begin{array}{rcl}
\mathsf{vars}[\![\mathsf{1}]\!] & \triangleq & \emptyset \\
\mathsf{vars}[\![\mathsf{x}]\!] & \triangleq & \{\mathsf{x}\} \\
\mathsf{vars}[\![\mathsf{A}_1 - \mathsf{A}_2]\!] & \triangleq & \mathsf{vars}[\![\mathsf{A}_1]\!] \cup \mathsf{vars}[\![\mathsf{A}_2]\!] \\
\mathsf{vars}[\![\mathsf{A}_1 < \mathsf{A}_2]\!] & \triangleq & \mathsf{vars}[\![\mathsf{A}_1]\!] \cup \mathsf{vars}[\![\mathsf{A}_2]\!] \\
\mathsf{vars}[\![\mathsf{B}_1 \ \mathsf{nand} \ \mathsf{B}_2]\!] & \triangleq & \mathsf{vars}[\![\mathsf{B}_1]\!] \cup \mathsf{vars}[\![\mathsf{B}_2]\!]
\end{array}$$

□

3 Solutions to Selected Exercises of Chapter 4

Solution to exercise 4.3

```
$ cat iterate1.c                 $ gcc iterate1.c
#include <stdio.h>               $ ./a.out
#define tt 1                     x = 3
int main () {                    $
   int x = 0;
   x = x + 1;
   while (tt) {
     x = x + 1;
     if (x > 2) break;
   } ;
   printf(”x = %d\n”, x);
}
```

□

4 Solutions to Selected Exercises of Chapter 5

Solution to exercise 5.10

```
(* File main.ml *)

open AbstractSyntax

let rec calculate_aexpr a r = match a with
| Num i -> i
| Var v -> if List.mem_assoc v r then List.assoc v r
           else failwith (”uninitialized variable:” ^ v)
```

```
| Minus (a1, a2) -> (calculate_aexpr a1 r) - (calculate_aexpr a2 r)

let rec calculate_node s r  = match s with
    | Prog sl       -> calculate_nodelist sl r
    | Assign (v, a) -> let va = calculate_aexpr a r in ((v, va) :: r,
         va)
    | Stmtlist sl   -> calculate_nodelist sl r
    | _             -> failwith ”invalid program”
and calculate_nodelist sl r = match sl with
    | []       -> failwith ”invalid program”
    | [s]      -> calculate_node s r
    | s :: sl’ -> let (r’, va) = calculate_nodelist sl’ r in
                     calculate_node s r’;; (* nodes in inverse order
                         *)

let lexbuf = Lexing.from_channel stdin in
  try
    let (r, va) = calculate_node (Parser.prog Lexer.token lexbuf) []
        in
       print_int va; print_newline ()
  with
  | Lexer.Error msg ->
      Printf.fprintf stderr ”%s%!” msg
  | Parser.Error ->
      Printf.fprintf stderr
          ”At offset %d: syntax error.\n%!” (Lexing.lexeme_start
              lexbuf)
```

□

Solution to exercise 5.11

```
(* File interpreter.ml *)

open AbstractSyntax

let bot = 0
and neg = 1
and zero = 2
and pos = 3
and negz = 4
and nzero = 5
and posz = 6
and top = 7

let print_sign s = match s with
| 0   -> print_string ”_|_”
| 1   -> print_string ”<0”
| 2  -> print_string ”=0”
| 3   -> print_string ”>0”
| 4  -> print_string ”<=0”
| 5 -> print_string ”=/=0”
| 6  -> print_string ”>=0”
| 7   -> print_string ”T”
| _ -> failwith ”incorrect sign”

let minus_sign = Array.make 8 (Array.make 8 bot);;

Array.set minus_sign bot   [|bot;bot;bot;  bot;bot; bot;  bot; bot
    |];;
Array.set minus_sign neg   [|bot;top;neg;  neg;top; top;  neg; top
    |];;
Array.set minus_sign zero  [|bot;pos;zero; neg;posz;nzero;negz;top
    |];;
Array.set minus_sign pos   [|bot;pos;pos;  top;pos; top;  top; top
    |];;
Array.set minus_sign negz  [|bot;top;negz; neg;top; top;  negz;top
    |];;
Array.set minus_sign nzero [|bot;top;nzero;top;top; top;  top; top
    |];;
Array.set minus_sign posz  [|bot;pos;posz; top;posz;top;  top; top
    |];;
Array.set minus_sign top   [|bot;top;top;  top;top; top;  top; top
    |];;

let rec analyze_aexpr a r = match a with
| Num i -> if i < 0 then neg
           else if i = 0 then zero
           else pos
| Var v -> if List.mem_assoc v r then List.assoc v r else
           failwith (”uninitialized variable:” ^ v)
| Minus (a1, a2) -> let s1 = (analyze_aexpr a1 r)
                    and s2 = (analyze_aexpr a2 r) in
                       Array.get (Array.get minus_sign s1) s2


let rec analyze_node s r  = match s with
    | Prog sl       -> analyze_nodelist sl r
    | Assign (v, a) -> let va = analyze_aexpr a r in ((v, va) :: r,
        va)
    | Stmtlist sl   -> analyze_nodelist sl r
    | _             -> failwith ”invalid program”
and analyze_nodelist sl r = match sl with
    | []       -> failwith ”invalid program”
    | [s]      -> analyze_node s r
    | s :: sl’ -> let (r’, va) = analyze_nodelist sl’ r in
                     analyze_node s r’;; (* nodes in inverse order *)

let lexbuf = Lexing.from_channel stdin in
```

```
try
  let (r, va) = analyze_node (Parser.prog Lexer.token lexbuf) [] in
     print_sign va; print_newline ()
with
| Lexer.Error msg ->
    Printf.fprintf stderr ”%s%!” msg
| Parser.Error ->
    Printf.fprintf stderr ”At offset %d: syntax error.\n%!”
                                        (Lexing.lexeme_start lexbuf
                                            )
```

□

Solution to exercise 9.13 Assume that we have an algorithm `correct(P, f)` that always terminates and returns true if and only if $\mathsf{P}(n) = f(n)$ for all integers n for which $f(n)$ is well defined ($n \in \mathsf{dom}(f)$). We can even fix f e.g. $f(n) = n^3$.

Then the following algorithm would always terminate and return true if and only if `P` terminates on input `i`

```
let terminate(p, i) =
  let t(n) = p(i); return f(n) in
    correct(t, f);
```

`correct(t, f)` is true if and only if $\mathsf{t}(n) = f(n)$ for all integers n for which $f(n)$ is well defined, if and only if `P` terminates on input `i`, which is undecidable. □

5 Solutions to Selected Exercises of Chapter 11

Solution to exercise 11.8 $h(x) = \{f(x)\}$ and $\sqcup = \cup$. □

Solution to exercise 11.11

$$\begin{array}{lll}
 & R^*(P) \supseteq Q & \\
\Leftrightarrow & \forall y \in Q \,.\, \forall x \in P \,.\, \langle x,\ y\rangle \in R & \wr\text{definition of } \supseteq \text{ and } R^*\wr \\
\Leftrightarrow & \forall x \in P \,.\, \forall y \in Q \,.\, \langle x,\ y\rangle \in R & \wr\text{definition of } \forall\wr \\
\Leftrightarrow & P \subseteq R^\dagger(Q) & \wr\text{definition of } \subseteq \text{ and } R^\dagger\wr
\end{array}$$

□

Solution to exercise 11.12

$$\begin{array}{lll}
 & \alpha_{\mathsf{fr}}(F) \subseteq R & \\
\Leftrightarrow & \{\langle a,\ b\rangle \mid b \in F(a)\} \subseteq R & \wr\text{definition } \alpha_{\mathsf{fr}}\wr \\
\Leftrightarrow & \forall a, b \,.\, (b \in F(a)) \Rightarrow \langle a,\ b\rangle \in R & \wr\text{definition of } \subseteq\wr \\
\Leftrightarrow & \forall a \,.\, \forall b \in F(a) \,.\, \langle a,\ b\rangle \in R & \wr\text{definition of } \Rightarrow\wr \\
\Leftrightarrow & \forall a \,.\, F(a) \subseteq \{b \mid \langle a,\ b\rangle \in R\} & \wr\text{definition of } \subseteq\wr
\end{array}$$

$\Leftrightarrow F \mathrel{\dot{\subseteq}} a \mapsto \{b \mid \langle a,\ b\rangle \in R\}$ $\wr$definition of $\dot{\subseteq}$$\wr$

$\Leftrightarrow F \mathrel{\dot{\subseteq}} \gamma_{\mathsf{fr}}(R)$ $\wr$definition of $\dot{\subseteq}$$\wr$

One can check that $\gamma_{\mathsf{fr}} \circ \alpha_{\mathsf{fr}}$ is a bijection with inverse $\alpha_{\mathsf{fr}} \circ \gamma_{\mathsf{fr}}$. □

Solution to exercise 11.14 For all $w \in \Sigma^*$, $L_1, L_2 \in \wp(\Sigma^*)$, we have $L_1 \subseteq w^{-1}L_2$ if and only if $(x \in L_1 \Rightarrow wx \in L_2)$ if and only if $wL_1 \subseteq L_2$ so $wL_1 \subseteq L_2 \Leftrightarrow L_1 \subseteq w^{-1}L_2$. Moreover $w^{-1}(wL) = L$ for all $L \in \wp(\Sigma^*)$. Therefore $\langle\wp(\Sigma^*),\ \subseteq\rangle \underset{\alpha_{\overleftarrow{w}}}{\overset{\gamma_{\overleftarrow{w}}}{\substack{\twoheadleftarrow \\ \longrightarrow}}} \langle\wp(\Sigma^*),\ \subseteq\rangle$ where $\alpha_{\overleftarrow{w}}(L) = wL$ and $\gamma_{\overleftarrow{w}}(L) = w^{-1}L$. Similarly, $L_1w \subseteq L_2 \Leftrightarrow L_1 \subseteq L_2w^{-1}$ so $\langle\wp(\Sigma^*),\ \subseteq\rangle \underset{\alpha_{\overrightarrow{w}}}{\overset{\gamma_{\overrightarrow{w}}}{\leftrightarrows}} \langle\wp(\Sigma^*),\ \subseteq\rangle$ where $\alpha_{\overrightarrow{w}}(L) = Lw$ and $\gamma_{\overrightarrow{w}}(L) = Lw^{-1}$. M □

Solution to exercise 11.16 A property of a distribution is an element of $\wp(\mathbb{V} \to [0,1])$. Define $\alpha_{\mathsf{E}} \in \wp(\mathbb{V} \to [0,1]) \to \wp(\mathbb{V})$ by $\alpha_{\mathsf{E}}(\mathscr{P}) \triangleq \{\mathsf{E}(X) \mid P_X \in \mathscr{P}\}$. This is the homomorphic/partitioning abstraction of exercise 11.6 and so a Galois connection. In statistics one is often interested in properties of a given distribution P_X. Then $\alpha_{\mathsf{E}}(\{P_X\}) = \{\mathsf{E}(X)\}$ which is identified with $\mathsf{E}(X)$. The concretization is a set of distributions, so the best-guess prediction based on the expectation is valid for any of them, which can be imprecise for skewed distributions with mean far from the median. □

Solution to exercise 11.20

$\alpha \fatsemi \sqsubseteq\ =\ \leqslant \fatsemi \gamma^{-1}$

$\Leftrightarrow\ \forall P, Q : (\langle P,\ Q\rangle \in \alpha \fatsemi \sqsubseteq) \Leftrightarrow (\langle P,\ Q\rangle \in \leqslant \fatsemi \gamma^{-1})$ $\wr$def. equality of relations$\wr$

$\Leftrightarrow\ \forall P, Q : (\exists R : \langle P,\ R\rangle \in \alpha \wedge \langle R,\ Q\rangle \in \sqsubseteq) \Leftrightarrow (\exists R' : \langle P,\ R'\rangle \in \leqslant \wedge \langle R',\ Q\rangle \in \gamma^{-1})$

$\wr$def. composition of relations $r_1 \fatsemi r_2 \triangleq \{\langle x,\ z\rangle \mid \exists y : \langle x,\ y\rangle \in r_1 \wedge \langle y,\ z\rangle \in r_2\}$$\wr$

$\Leftrightarrow\ \forall P, Q : (\exists R : \langle P,\ R\rangle \in \alpha \wedge \langle R,\ Q\rangle \in \sqsubseteq) \Leftrightarrow (\exists R' : \langle P,\ R'\rangle \in \leqslant \wedge \langle Q,\ R'\rangle \in \gamma)$

$\wr$def. inverse of relations$\wr$

$\Leftrightarrow\ \forall P, Q : (\exists R : \langle P,\ R\rangle \in \alpha \wedge R \sqsubseteq Q) \Leftrightarrow (\exists R' : P \leqslant R' \wedge \langle Q,\ R'\rangle \in \gamma)$

$\wr$def. order relations$\wr$

$\Leftrightarrow\ \forall P, Q : (\exists R : R = \alpha(P) \wedge R \sqsubseteq Q) \Leftrightarrow (\exists R' : P \leqslant R' \wedge R' = \gamma(Q))$ $\wr$$\alpha$ and γ are functions$\wr$

$\Leftrightarrow\ \forall P, Q : (\alpha(P) \sqsubseteq Q) \Leftrightarrow (P \leqslant \gamma(Q))$ $\wr$simplification$\wr$

$\Leftrightarrow\ \langle C,\ \leqslant\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A,\ \sqsubseteq\rangle$ $\wr$by (11.1)$\wr$

□

Solution to exercise 11.22 For all $f \in \mathscr{D} \xrightarrow{\nearrow} \mathscr{D}$ and $y \in \mathscr{D}$,

$\alpha_p(f) \sqsubseteq y$

$\Leftrightarrow\ f(p) \sqsubseteq y$ $\wr$definition of $\alpha_p$$\wr$

$\Leftrightarrow\ \forall x \sqsubseteq p\ .\ f(x) \sqsubseteq y$ $\wr$$f$ increasing and $\sqsubseteq$ reflexive and transitive$\wr$

$\Leftrightarrow\ \forall x\ .\ f(x) \sqsubseteq \llbracket x \sqsubseteq p \mathrel{?} y \mathrel{:} \top \rrbracket$ $\wr$def. conditional and supremum $\top$$\wr$

$\Leftrightarrow\ \forall x \mathrel{.} f(x) \sqsubseteq \gamma_p(y)(x)$ ⟅by defining $\gamma_p(y)(x) \triangleq \llbracket x \sqsubseteq p \mathrel{?} y \mathrel{\mathring{:}} \top \rrbracket$⟆

$\Leftrightarrow\ f \mathrel{\dot{\sqsubseteq}} \gamma_p(y)$ ⟅pointwise⟆

□

Solution to exercise 11.23

$\quad \alpha_h(X) \mathrel{\dot{\subseteq}} Y$

$\Leftrightarrow\ \forall a \in A \mathrel{.} \alpha_h(X)\, a \subseteq Y(a)$ ⟅pointwise definition of $\dot{\subseteq}$⟆

$\Leftrightarrow\ \forall a \in A \mathrel{.} \{f(a)x \mid x \in X\} \subseteq Y(a)$ ⟅definition of α_h⟆

$\Leftrightarrow\ \forall a \in A \mathrel{.} \forall x \in X \mathrel{.} f(a)x \in Y(a)$ ⟅definition of $\subseteq$⟆

$\Leftrightarrow\ \forall x \in X \mathrel{.} \forall a \in A \mathrel{.} f(a)x \in Y(a)$ ⟅definition of $\forall$⟆

$\Leftrightarrow\ X \subseteq \{x \mid \forall a \in A \mathrel{.} f(a)x \in Y(a)\}$ ⟅definition of $\subseteq$⟆

$\Leftrightarrow\ X \subseteq \gamma_h(Y)$ ⟅by defining $\gamma_h(Y) \triangleq \{x \mid \forall a \in A \mathrel{.} f(a)x \in Y(a)\}$⟆

□

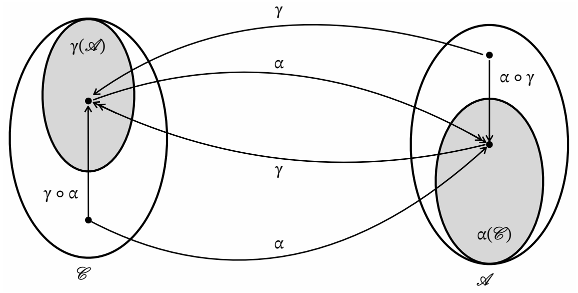
Solution to exercise 11.31 If $x \in X$ then $x \sqsubseteq_1 \sqcup_1 X$ by definition of the lub so $f(x) \sqsubseteq_2 f(\sqcup_1 X)$ because f is increasing, proving that $f(\sqcup_1 X)$ is an upper bound of $\{f(x) \mid x \in X\}$, hence $\sqcup_2\{f(x) \mid x \in X\} \sqsubseteq_2 f(\sqcup_1 X)$ by definition of an existing lub. □

Solution to exercise 11.44 Define $\alpha_y(z) = x \times y$ and $\gamma_y(x) = x \div y$. Then $\forall x, y, z \in \mathbb{N} \mathrel{.} z \times y \leqslant x \Leftrightarrow z \leqslant x \div y$ implies $\alpha_y(z) \leqslant x \Leftrightarrow z \leqslant \gamma_y(x)$ i.e. $\langle \mathbb{N}, \leqslant \rangle \underset{\alpha_y}{\overset{\gamma_y}{\leftrightarrows}} \langle \mathbb{N}, \leqslant \rangle$ which by lemma 11.42, implies $x \div y = \max\{z \mid x \times y \leqslant x\}$. □

Solution to exercise 11.45 $\gamma \circ \alpha$ is extensive so for all $Q \in \mathscr{A} : \gamma(Q) \leqslant \gamma \circ \alpha(\gamma(Q))$. $\alpha \circ \gamma$ is reductive so $\gamma \circ \alpha(\gamma(Q)) \leqslant \gamma(Q)$. By antisymmetry $\gamma \circ \alpha \circ \gamma = \gamma$. The dual is $\alpha \circ \gamma \circ \alpha = \alpha$. □

Solution to exercise 11.46 Let $x = \gamma(y) \in \gamma(\mathscr{A})$. Then $\alpha(x) = \overline{\alpha} \circ \gamma(y)$ and so, by exercise 11.45, $\overline{\gamma} \circ \overline{\alpha}(x) = \overline{\gamma} \circ \overline{\alpha} \circ \gamma(y) = \overline{\gamma} \circ \alpha \circ \gamma(y) = \gamma \circ \alpha \circ \gamma(y) = \gamma(y) = x$. Therefore $\overline{\gamma} \circ \overline{\alpha} = \mathbb{1}_{\mathscr{A}}$. The proof that $\overline{\alpha} \circ \overline{\gamma} = \mathbb{1}_{\mathscr{C}}$ is dual. □

In complement to the solution to exercise 11.46, the following figure



shows the bijection between $\gamma \circ \alpha(\mathscr{C}) = \gamma(\mathscr{A})$ and $\alpha \circ \gamma(\mathscr{A}) = \alpha(\mathscr{C})$ (in gray) where

$\gamma \circ \alpha$ is an upper closure and $\alpha \circ \gamma$ is a lower closure, which follows from $\alpha \circ \gamma \circ \alpha = \alpha$ and dually $\gamma \circ \alpha \circ \gamma = \gamma$.

Solution to exercise 11.48

$$\begin{array}{lll}
 & \gamma(a) & \\
= & \max\{c \in \mathscr{C} \mid c \sqsubseteq \gamma(a)\} & \wr\text{The max exists and is } \gamma(a) \text{ by reflexivity}\wr \\
= & \max\{c \in \mathscr{C} \mid \alpha(c) \preccurlyeq a\} & \wr\langle\mathscr{C},\ \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle\mathscr{A},\ \preccurlyeq\rangle\wr \\
= & \max\{c \in \mathscr{C} \mid \alpha(c) \in \downarrow a\} & \wr\text{definition of } \downarrow a \triangleq \{x \in \mathscr{A} \mid x \preccurlyeq a\}\wr \\
= & \max \alpha^{-1}(\downarrow a) & \wr\text{definition of } \alpha^{-1}(\downarrow a) \triangleq \{c \in \mathscr{C} \mid \alpha(c) \in \downarrow a\}\wr
\end{array}$$

$\max \alpha^{-1}(\downarrow a)$ is the lub of $\alpha^{-1}(\downarrow a)$. The dual is $\alpha(c) = \min \gamma^{-1}(\uparrow a)$. □

Solution to exercise 11.50 — If α is surjective then $\forall \overline{P} \in \mathscr{A} : \exists P \in \mathscr{C} : \alpha(P) = \overline{P}$. Therefore if $\gamma(\overline{P}) = \gamma(\overline{P}')$ then $\gamma(\alpha(P)) = \gamma(\alpha(P'))$ for some $P, P' \in \mathscr{A}$ such that $\overline{P} = \alpha(P)$ and $\overline{P}' = \alpha(P')$. By reflexivity, $\gamma(\alpha(P)) \leqslant \gamma(\alpha(P'))$ hence $P \leqslant \gamma(\alpha(P'))$ because $\gamma \circ \alpha$ is extensive. By (11.1), this implies $\alpha(P) \sqsubseteq \alpha(P')$ that is $\overline{P} \sqsubseteq \overline{P}'$. Exchanging $\overline{P}$ and $\overline{P}'$ in the previous proof, we get $\overline{P}' \sqsubseteq \overline{P}$ and so $\overline{P} = \overline{P}'$ by antisymmetry, proving γ to be injective.

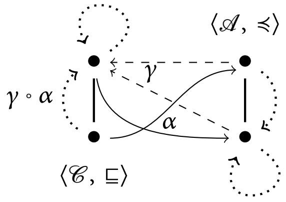
— By exercise 11.45, we have $\gamma \circ \alpha \circ \gamma(\overline{P}) = \gamma(\overline{P})$ for all $\overline{P} \in A$ so if γ is injective then $\alpha \circ \gamma(\overline{P}) = \overline{P}$.

— If $\alpha(P) = Q$ then $\alpha(P) \sqsubseteq Q$ so $P \leqslant \gamma(Q)$, proving $\gamma(Q)$ to be greater than all elements of $\{P \in \mathscr{C} \mid \alpha(P) = Q\}$. Moreover, $\alpha \circ \gamma$ is the identity on $\mathscr{A}$ so $\gamma(Q) \in \{P \in \mathscr{C} \mid \alpha(P) = Q\}$, proving $\gamma(Q)$ to be the maximum of the elements of $\{P \in \mathscr{C} \mid \alpha(P) = Q\}$.

— Finally, if $\forall Q \in A : \gamma(Q) = \max\{P \in \mathscr{C} \mid \alpha(P) = Q\}$ then given any $Q \in A$, $\gamma(Q) \in \{P \in \mathscr{C} \mid \alpha(P) = Q\}$ so $\alpha(\gamma(Q)) = Q$, proving α to be surjective.

— An isomorphism between $\mathscr{C}$ and $\mathscr{A}$ is not necessarily increasing. □

Solution to exercise 11.59 Not necessarily—here is a counterexample (α is not increasing).



$\langle\mathscr{A},\ \preccurlyeq\rangle$
γ
$\gamma \circ \alpha$
$\alpha \circ \gamma$
α
$\langle\mathscr{C},\ \sqsubseteq\rangle$

□

Solution to exercise 11.65 Let us prove $\langle\wp(\mathscr{P}),\ \subseteq\rangle \underset{\rotatebox[origin=c]{270}{$\rightsquigarrow$}}{\overset{\rotatebox[origin=c]{90}{$\rightsquigarrow$}}{\leftrightarrows}} \langle\wp(\mathscr{P}),\ \supseteq\rangle$.

$$\begin{array}{lll}
 & \rotatebox[origin=c]{270}{$\rightsquigarrow$}(X) \supseteq Y & \\
\Leftrightarrow & \forall y \in Y \mathrel{.} y \in \rotatebox[origin=c]{270}{$\rightsquigarrow$}(X) & \wr\text{definition of } \supseteq\wr \\
\Leftrightarrow & \forall y \in Y \mathrel{.} \forall x \in X \mathrel{.} y \sqsubseteq x & \wr\text{definition of } \rotatebox[origin=c]{270}{$\rightsquigarrow$} \text{ and } \in\wr
\end{array}$$

$$\Leftrightarrow\ \forall x \in X \,.\, \forall y \in Y \,.\, y \sqsubseteq x \qquad \wr\text{definition of } \forall\int$$

$$\Leftrightarrow\ X \subseteq \{x \in \mathscr{P} \mid \forall y \in Y \,.\, y \sqsubseteq x\} \qquad \wr\text{definition of } \forall\int$$

$$\Leftrightarrow\ X \subseteq \uparrow(Y) \qquad \wr\text{definition of } \uparrow\int$$

By exercise 11.57, $\uparrow \circ \downarrow$ is a closure operator. Therefore $\langle \uparrow \circ \downarrow(\wp(\mathscr{P})),\ \supseteq\rangle$ is a complete lattice by exercise 11.64. We have $\dot{\uparrow}(x) \triangleq \uparrow(\{x\}) = \{z \in \mathscr{P} \mid \forall y \in \{x\} \,.\, y \sqsubseteq z\} = \{z \in \mathscr{P} \mid x \sqsubseteq z\}$. By the Galois connection and exercise 11.45, $\dot{\uparrow}(x) = \uparrow(\{x\}) = \uparrow \circ \downarrow \circ \uparrow(\{x\}) = \uparrow \circ \downarrow(\dot{\uparrow}(x))$, proving that $\dot{\uparrow} \in \mathscr{P} \to \uparrow \circ \downarrow(\wp(\mathscr{P}))$. Moreover, if $x \sqsubseteq y$ then $\{z \mid x \sqsubseteq z\} \supseteq \{z \mid y \sqsubseteq z\}$ by transitivity, and so, $\dot{\uparrow}(x) \supseteq \dot{\uparrow}(y)$. Conversely $\dot{\uparrow}(x) \supseteq \dot{\uparrow}(y)$ implies $\{z \mid x \sqsubseteq z\} \supseteq \{z \mid y \sqsubseteq z\}$ and so, by reflexivity and definition of $\supseteq$, $y \in \{z \mid x \sqsubseteq z\}$, proving that $x \sqsubseteq y$. It follows that $\dot{\uparrow}$ is an order embedding of $\langle\mathscr{P},\ \sqsubseteq\rangle$ into $\langle \uparrow \circ \downarrow(\wp(\mathscr{P})),\ \supseteq\rangle$ such that $\forall x, y \in \mathscr{P} \,.\, x \sqsubseteq y \Leftrightarrow \dot{\uparrow}(x) \supseteq \dot{\uparrow}(y)$. So $(x = y) \Leftrightarrow (x \sqsubseteq y \wedge y \sqsubseteq x) \Leftrightarrow (\dot{\uparrow}(x) \supseteq \dot{\uparrow}(y) \wedge \dot{\uparrow}(y) \supseteq \dot{\uparrow}(x)) \Leftrightarrow (\dot{\uparrow}(x) = \dot{\uparrow}(y))$. By contraposition in section 2.4.1, $(x \neq y) \Leftrightarrow (\dot{\uparrow}(x) \neq \dot{\uparrow}(y))$ proving that $\dot{\uparrow}$ is bijective so distinct elements of $\mathscr{P}$ are be mapped to distinct elements of $\uparrow \circ \downarrow(\wp(\mathscr{P}))$. If $x, y \in \mathscr{P}$ are not comparable then $\dot{\uparrow}(x)$ and $\dot{\uparrow}(y)$ are not comparable because otherwise $\dot{\uparrow}(x) \supseteq \dot{\uparrow}(y)$ would imply $x \sqsubseteq y$, a contradiction, and inversely. Otherwise $x \sqsubseteq y$ are comparable and then $x \sqsubseteq y \Leftrightarrow \dot{\uparrow}(x) \supseteq \dot{\uparrow}(y)$ implies that they have the same ordering in $\langle \uparrow \circ \downarrow(\wp(\mathscr{P})),\ \supseteq\rangle$. Let $\dot{\uparrow}^{-1}$ be the inverse of the bijection $\dot{\uparrow} \in \mathscr{P} \rightarrowtail\!\!\!\!\twoheadrightarrow \uparrow \circ \downarrow(\wp(\mathscr{P}))$. We have $X \supseteq Y$ implies $\dot{\uparrow} \circ \dot{\uparrow}^{-1}(X) \supseteq \dot{\uparrow} \circ \dot{\uparrow}^{-1}(Y)$ implies $\dot{\uparrow}^{-1}(X) \sqsubseteq \dot{\uparrow}^{-1}(Y)$ by the embedding, proving that $\dot{\uparrow}^{-1}$ is decreasing. If $x \in \mathscr{P}$ and $Y \in \uparrow \circ \downarrow(\wp(\mathscr{P}))$ then $\dot{\uparrow}x \supseteq Y \Leftrightarrow \dot{\uparrow}^{-1} \circ \dot{\uparrow}x \sqsubseteq \dot{\uparrow}^{-1}(Y) \Leftrightarrow x \sqsubseteq \dot{\uparrow}^{-1}(Y)$, proving $\langle\mathscr{P},\ \sqsubseteq\rangle \underset{\dot{\uparrow}}{\overset{\dot{\uparrow}^{-1}}{\leftrightarrows}} \langle \uparrow \circ \downarrow(\wp(\mathscr{P})),\ \supseteq\rangle$.

The proof by MacNeille [2, THEOREM 11.9] uses the order embedding of x into cuts $\langle\{y \mid y \sqsubseteq x\},\ \{z \mid x \sqsubseteq z\}\rangle$ generalizing the cuts used by Dedekind [1] to construct the real numbers from the rational numbers, hence the name *Dedekind–MacNeille completion*. □

Solution to exercise 11.68 A hint is to use lemma 11.38 for α_a. □

Solution to exercise 11.73 See theorem 11.72. □

Solution to exercise 12.27 An execution starting with an initial environment in P, will have the following behaviors (a) $\mathsf{post}[\mathsf{S}]P \subseteq Q$, (b) $\mathsf{post}[\mathsf{S}]P \subseteq \neg Q$, (c) $\mathsf{post}[\mathsf{S}]P \subseteq \{\bot\}$, (ab) $\mathsf{post}[\mathsf{S}]P \subseteq \mathbb{Q} \setminus \{\bot\} \wedge \mathsf{post}[\mathsf{S}]P \nsubseteq Q \wedge \mathsf{post}[\mathsf{S}]P \nsubseteq \neg Q$, (ac) $\mathsf{post}[\mathsf{S}]P \subseteq Q \cup \{\bot\}$, (bc) $\mathsf{post}[\mathsf{S}]P \subseteq \neg Q \cup \{\bot\}$, (abc) $\mathsf{post}[\mathsf{S}]P \nsubseteq Q \wedge \mathsf{post}[\mathsf{S}]P \nsubseteq \neg Q \wedge \mathsf{post}[\mathsf{S}]P \nsubseteq \{\bot\}$. □

6 Solutions to Selected Exercises of Chapter 13

Solution to exercise 13.2 The smallest topology on $\mathscr{X}$ is $\{\varnothing, \mathscr{X}\}$ and the largest is $\wp(\mathscr{X})$. □

Solution to exercise 13.3 $\wp(\mathscr{X})$ is the only topology that makes every subset of $\mathscr{X}$

both an open and closed set. □

7 Solutions to Selected Exercises of Chapter 14

Solution to exercise 14.29 The fairness trace property is $F = \{\pi \in \mathbb{T}^{\infty}(\{\mathsf{a}_0, \mathsf{a}_1\}) \mid \forall i \in \{0, 1\} \;.\; \forall j \in \mathbb{N} \;.\; \exists k \geqslant j \;.\; \pi_k = \mathsf{a}_i\}$. Any finite trace $\pi_0 \in \mathbb{T}^{+}(\{\mathsf{a}_0, \mathsf{a}_1\})$ is a prefix of some trace in F so that the limit closure of the prefix closure is $\mathbb{T}^{+\infty}(\{\mathsf{a}_0, \mathsf{a}_1\})$ with complement $\varnothing$. It follows that $\mathsf{live}(F) = F$. □

Solution to exercise 14.33 The property of a program P "to be deterministic" is $\boldsymbol{\mathcal{S}}^*[\![\mathsf{P}]\!]$ is a functional relation, formally $\boldsymbol{\mathcal{S}}^*[\![\mathsf{P}]\!] \in \{\boldsymbol{\mathcal{S}} \mid \forall\langle\pi_0,\ \pi\rangle, \langle\pi_0,\ \pi'\rangle \in \boldsymbol{\mathcal{S}} \;.\; \pi = \pi'\}$. This is not a trace property hence neither a safety nor a liveness property. □

8 Solutions to Selected Exercises of Chapter 16

Solution to exercise 16.14 Because proofs are finite, only finitely many elements of the universe can be proved in a proof, so the finite set of proved elements cannot contain the infinite premise. However, the least fixpoint of the consequence operator that considers all proofs may be able to use the rule with infinite premise. Consider, for example, $R = \left\{\frac{\varnothing}{n} \,\middle|\, n \geqslant 1\right\} \cup \left\{\frac{\mathbb{N}^+}{0}\right\}$ where $\mathbb{N}^+$ is the set of strictly positive naturals.

A proof is reduced to an axiom so cannot use the rule $\frac{\mathbb{N}^+}{0}$ and so can prove any $n \in \mathbb{N}^+$ but cannot prove 0. Therefore, according to definition 16.10, the rules define $\mathbb{N}^+$.

The consequence operator is $F_R(X) = \mathbb{N}^+ \cup \{0 \mid \mathbb{N}^+ \subseteq X\}$. The iterates of F_R are $\varnothing$, $\mathbb{N}^+$, $\mathbb{N}$ which is the least fixpoint and contain 0, a counterexample to theorem 16.12 in case infinite premises would had been allowed. □

Solution to exercise 16.15 The language L defined by the context-free grammar $X ::= X\ X \mid a$ can be specified by the deductive system with axiom $a \in L$ and inference rule $\frac{\sigma_1 \in L, \sigma_2 \in L}{\sigma_1\sigma_2 \in L}$. The corresponding fixpoint definition is $L = \mathsf{lfp}^{\subseteq} F$ where $F(X) = \{a\} \cup \{\sigma_1\sigma_2 \mid \sigma_1, \sigma_2 \in L\} = \{a^n \mid n \geqslant 1\}$. □

9 Solutions to Selected Exercises of Chapter 17

Solution to exercise 17.19 The iterates $\langle \mathscr{F}^{+\infty n},\ n \in \mathbb{N}\rangle$ of $\mathscr{F}^{+\infty}[\![\mathsf{S}]\!]$ from $\ni$ are (for all initial traces $(\pi_1\ell)$)

$$\begin{aligned}
\mathscr{F}^{+\infty 0}(\pi_1\ell) &= \ni \\
\mathscr{F}^{+\infty 1}(\pi_1\ell) &= \mathsf{let}\, v = \varrho(\pi_1\ell) + 1 \text{ in } \ell \xrightarrow{\mathsf{tt}} \ell' \xrightarrow{\mathsf{x = x + 1 = }v} \ell
\end{aligned}$$

$$\mathscr{F}^{+\infty^2}(\pi_1\ell) \;=\; \text{let}\,\forall i \in [1,2]\,.\,v(i) = \varrho(\pi_1\ell) + i \text{ in}$$

$$\ell \xrightarrow{\;\mathsf{tt}\;} \ell' \xrightarrow{\;\texttt{x = x + 1 = }v(1)\;} \ell \xrightarrow{\;\mathsf{tt}\;} \ell' \xrightarrow{\;\texttt{x = x + 1 = }v(2)\;} \ell$$

...

$$\mathscr{F}^{+\infty^n}(\pi_1\ell) \;=\; \text{let}\,\forall i \in [1,n]\,.\,v(i) = \varrho(\pi_1\ell) + i \text{ in}\left(\ell \xrightarrow{\;\mathsf{tt}\;} \ell' \xrightarrow{\;\texttt{x = x + 1 = }v(i)\;} \ell\right)_{i=1}^{n}$$

$$\wr\text{induction hypothesis}\int$$

...

$$\widehat{\mathcal{S}}^{+\infty}[\![\texttt{while}\;\ell\;\texttt{(B)}\;\texttt{S}_b]\!] \;=\; \text{let}\,\forall i \in \mathbb{N}\,.\,v(i) = \varrho(\pi_1\ell) + i \text{ in}\left(\ell \xrightarrow{\;\mathsf{tt}\;} \ell' \xrightarrow{\;\texttt{x = x + 1 = }v(i)\;} \ell\right)_{i=1}^{\infty}$$

□

10 Solutions to Selected Exercises of Chapter 18

Solution to exercise 18.20 By definition of the infimum $\bot \sqsubseteq \gamma(y)$. Assume, by induction hypothesis, that $f^n(\bot) \sqsubseteq \gamma(y)$. Then $f^{n+1}(\bot) = f(f^n(\bot)) \sqsubseteq f(\gamma(y)) \sqsubseteq \gamma(\overline{f}(y)) \sqsubseteq \gamma(y)$ by semicommutation and f and γ are increasing. By recurrence, $\forall n \in \mathbb{N}\,.\, f^n(\bot) \sqsubseteq \gamma(y)$, proving, by theorem 15.21 and definition of the lub that $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq \gamma(y)$ (in particular, by reflexivity, for any fixpoint y of $\overline{f}$). □

Solution to exercise 18.31 (a) Assume $\alpha \circ f = \overline{f} \circ \alpha$. Then $\alpha \circ f = \overline{f} \circ \alpha$ then $\alpha \circ f = \alpha \circ f \circ \gamma \circ \alpha$ and so if $\alpha(x) = \alpha(y)$ then $\alpha \circ f(x) = \alpha \circ f \circ \gamma \circ \alpha(x) = \alpha \circ f \circ \gamma \circ \alpha(y) = \alpha \circ f(y)$, proving (18.32). Conversely, by the dual of exercise 11.45, $\forall x \in \mathscr{C}\,.\,\alpha(x) = \alpha(\gamma \circ \alpha(x))$ so (18.32) implies that $f(\alpha(x)) = \alpha(f(\gamma \circ \alpha(x))) = \overline{f}(\alpha(x))$.

(b) Assume that $\bigvee_{i\in\Delta}\overline{x}_i$ and $\bigsqcup_{i\in\Delta}\gamma(\overline{x}_i)$ do exist in the posets $\mathscr{A}$ and $\mathscr{C}$. Then

$$\overline{f}(\bigvee_{i\in\Delta} x_i)$$

$$= \alpha \circ f \circ \gamma(\bigvee_{i\in\Delta} x_i) \qquad \wr\text{definition of } \overline{f}\int$$

$$= \alpha \circ f(\bigsqcup_{i\in\Delta}\gamma(x_i))$$

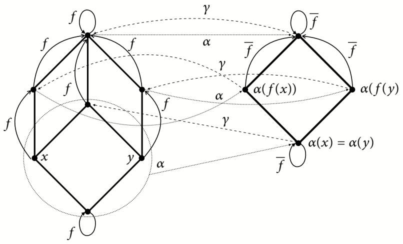
$\wr$By lemma 11.38, α preserves existing lubs and by exercise 11.50, $\alpha \circ \gamma = \mathbb{1}_{\mathscr{A}}$ so $\alpha(\bigsqcup_{i\in\Delta}\gamma(x_i)) = \bigvee_{i\in\Delta}\alpha \circ \gamma(x_i) = \bigvee_{i\in\Delta} x_i = \alpha(\gamma(\bigvee_{i\in\Delta} x_i))$ and so, by (18.32), $\alpha(f(\bigsqcup_{i\in\Delta}\gamma(x_i))) = \alpha(f(\gamma(\bigvee_{i\in\Delta} x_i)))\int$

$$= \alpha \circ \bigsqcup_{i\in\Delta} f(\gamma(x_i)) \qquad \wr\text{by hypothesis, } f \text{ preserves existing lubs}\int$$

$$= \bigvee_{i\in\Delta}\alpha \circ f \circ \gamma(x_i) \qquad \wr\text{by lemma 11.38, } \alpha \text{ preserves existing lubs}\int$$

$$= \bigvee_{i\in\Delta}\overline{f}(x_i) \qquad \wr\text{definition of } \overline{f}\int$$

(c) Here is a counterexample.



□

Solution to exercise 18.33 $\langle D \xrightarrow{\sqcup} D,\ \dot{\sqsubseteq}\rangle$ and $\langle D,\ \sqsubseteq\rangle$ are complete lattices, $\mathscr{F} \in (D \xrightarrow{\sqcup} D) \xrightarrow{\sqcup} (D \xrightarrow{\sqcup} D)$ is $\dot{\sqsubseteq}$-increasing. We have $\langle D \xrightarrow{\sqcup} D,\ \dot{\sqsubseteq}\rangle \underset{\alpha_x}{\overset{\gamma_x}{\leftrightarrows}} \langle D,\ \sqsubseteq\rangle$ by exercise 11.39 because lubs exist in a complete lattice and α_x preserves arbitrary joins:

$$
\begin{aligned}
&\alpha_x(\dot{\bigsqcup_i} f_i)\\
=\ &\mathscr{F}(\dot{\bigsqcup_i} f_i)x && \wr\text{definition of }\alpha_x\wr\\
=\ &(\dot{\bigsqcup_i}\mathscr{F}(f_i))x && \wr\mathscr{F}\text{ preserves joins}\wr\\
=\ &\bigsqcup_i(\mathscr{F}(f_i)x) && \wr\text{pointwise definition of }\dot{\bigsqcup}\wr\\
=\ &\bigsqcup_i\alpha_x(f_i) && \wr\text{definition of xs}\alpha_x\wr
\end{aligned}
$$

$F(x) \in D \xrightarrow{\sqcup} D$ is $\sqsubseteq$-increasing and we have the commutation property $\alpha_x \circ \mathscr{F} = F(x) \circ \alpha_x$. By theorem 18.23, it follows that $\mathsf{lfp}^{\sqsubseteq} F(x) = \alpha_x(\mathsf{lfp}^{\dot{\sqsubseteq}} \mathscr{F}) = \mathscr{F}(\mathsf{lfp}^{\dot{\sqsubseteq}} \mathscr{F})x = (\mathsf{lfp}^{\dot{\sqsubseteq}} \mathscr{F})x$ for all $x \in D$ so $\mathsf{lfp}^{\dot{\sqsubseteq}} \mathscr{F} = x \in D \mapsto \mathsf{lfp}^{\sqsubseteq} F(x)$. □

11 Solutions to Selected Exercises of Chapter 19

Solution to exercise 19.9 We have $\langle \wp(\mathbb{E}\mathrm{v} \times \mathbb{E}\mathrm{v}),\ \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \wp(\mathbb{E}\mathrm{v}),\ \subseteq\rangle$ with $\alpha(R) \triangleq \{\rho \mid \exists\rho_0 \in \mathbb{E}\mathrm{v}\ .\ \langle\rho_0, \rho\rangle \in R\}$ and $\gamma(r) \triangleq \{\langle\rho_0, \rho\rangle \mid \rho_0 \in \mathbb{E}\mathrm{v} \wedge \rho \in r\}$. By pointwise extension in exercise 11.21, it follows that $\langle \mathbb{L} \to \wp(\mathbb{E}\mathrm{v} \times \mathbb{E}\mathrm{v}),\ \dot{\subseteq}\rangle \underset{\dot{\alpha}}{\overset{\dot{\gamma}}{\leftrightarrows}} \langle \mathbb{L} \to \wp(\mathbb{E}\mathrm{v}),\ \dot{\subseteq}\rangle$. It follows, by theorem 11.78, that $\langle \wp(\mathbb{E}\mathrm{v}\times\mathbb{E}\mathrm{v}) \xrightarrow{\nearrow} (\mathbb{L} \to \wp(\mathbb{E}\mathrm{v}\times\mathbb{E}\mathrm{v})),\ \ddot{\subseteq}\rangle \underset{\vec{\alpha}}{\overset{\vec{\gamma}}{\leftrightarrows}} \langle \wp(\mathbb{E}\mathrm{v}) \xrightarrow{\nearrow} (\mathbb{L} \to \wp(\mathbb{E}\mathrm{v})),\ \ddot{\subseteq}\rangle$ where $\vec{\alpha} \triangleq \boldsymbol{\mathcal{S}} \mapsto \dot{\alpha} \circ \boldsymbol{\mathcal{S}} \circ \gamma$ and $\vec{\gamma} \triangleq \overline{\boldsymbol{\mathcal{S}}} \mapsto \dot{\gamma} \circ \overline{\boldsymbol{\mathcal{S}}} \circ \alpha$. Moreover, $\boldsymbol{\mathcal{S}}^{\vec{\mathsf{r}}}[\![\mathsf{S}]\!] = \vec{\alpha}(\boldsymbol{\mathcal{S}}^{\vec{\mathsf{R}}}[\![\mathsf{S}]\!])$. □

Solution to exercise 19.27 No, because of iteration. A counterexample is provided by example 19.1 □

Solution to exercise 19.31

$$
\begin{aligned}
&—\ \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_5]\!]\ \mathbb{E}\mathrm{v}\ \ell_6 && (1)\\
=\ &\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_6]\!]\ \{\rho \in \mathbb{E}\mathrm{v} \mid \rho(\mathsf{x}) = 0\}\ \ell_6 && \wr(19.22)\wr
\end{aligned}
$$

$= \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 0\}$ $\wr$(19.25)$\int$

— $\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_5]\!]\ \mathbb{Ev}\ \ell_3$ (2)

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_6]\!]\ (\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 0\})\ \ell_3 \cup \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \neq 0\}$ $\wr$(19.22)$\int$

$= \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \neq 0\}$ $\wr$(19.25)$\int$

— $(\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_3]\!]\mathbb{Ev}\ \ell_3)$ (3)

$= (\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_5]\!]\mathbb{Ev}\ \ell_3)$ $\wr$(19.24) and (19.20)$\int$

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_6]\!]\ (\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 0\})\ \ell_3 \cup \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \neq 0\}$ $\wr$(19.22)$\int$

$= \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \neq 0\}$ $\wr$(19.25)$\int$

— $\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_2]\!]\ \mathbb{Ev}\ \ell_5 \quad = \quad \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_3\ \mathsf{S}_7]\!]\ \mathbb{Ev}\ \ell_5$ (4)

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_7]\!](\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_3]\!]\mathbb{Ev}\ \ell_3)\ \ell_5$ $\wr$(19.24) and $\mathsf{at}[\![\mathsf{S}_7]\!] = \ell_3$$\int$

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_7]\!](\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \neq 0\})\ \ell_5$ $\wr$(3)$\int$

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_8]\!]\ (\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 1\})\ \ell_5 \cup \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \notin \{0, 1\}\}$ $\wr$(19.22)$\int$

$= \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \notin \{0, 1\}\}$ $\wr$(19.25)$\int$

– $\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_7]\!](\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \neq 0\})\ \ell_6$ (5)

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_8]\!]\ (\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \neq 0\} \cap \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 1\})\ \ell_6$ $\wr$(19.22)$\int$

$= \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 1\}$ $\wr$(19.25)$\int$

– $\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{P}]\!]\ \mathbb{Ev}\ \ell_6$

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_1]\!]\ \mathbb{Ev}_1\ \ell_6$ $\wr$(19.19)$\int$

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_2]\!]\mathbb{Ev}\ \ell_6 \cup \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_9]\!](\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_2]\!]\mathbb{Ev}\ \ell_5)\ \ell_6$ $\wr$(19.24)$\int$

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_3]\!]\mathbb{Ev}\ \ell_6 \cup \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_7]\!](\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_3]\!]\mathbb{Ev}\ \ell_3)\ \ell_6 \cup \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_9]\!](\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_2]\!]\mathbb{Ev}\ \ell_5)\ \ell_6$

$\wr$(19.24)$\int$

$= \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_5]\!](\mathbb{Ev})\ \ell_6 \cup \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_7]\!](\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_5]\!]\mathbb{Ev}\ \ell_3)\ \ell_6 \cup \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_9]\!](\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_2]\!]\mathbb{Ev}\ \ell_5)\ \ell_6$

$\wr$(19.24), $\ell_6 \notin \mathsf{labx}[\![\mathtt{Sl}_4]\!]$, and $\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathtt{Sl}_4]\!]\mathbb{Ev}\ \ell_1 = \mathbb{Ev}$ by (19.20)$\int$

$= \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 0\} \cup \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_7]\!](\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \neq 0\})\ \ell_6 \cup \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_9]\!](\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \notin \{0, 1\})\ \ell_6$
$\wr$(1), (2), and (4)$\int$

$= \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 0\} \cup \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 1\} \cup \widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathsf{r}}}[\![\mathsf{S}_9]\!](\{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) \notin \{0, 1\})\ \ell_6$ $\wr$(5)$\int$

$= \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 0\} \cup \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 1\} \cup \{\rho \in \mathbb{Ev} \mid \rho(\mathsf{x}) = 2\}$ $\wr$(19.12)$\int$

$= \{\rho \in \mathbb{Ev} \mid 0 \leqslant \rho(\mathsf{x}) \leqslant 2\}$ $\wr$definition of $\cup$$\int$

□

12 Solutions to Selected Exercises of Chapter 21

Solution to exercise 21.23 The $\{\bot^{¤}, \top^{¤}\}$ static analysis of the program

```
    while (0<1){ break; x=1; }
```

shows that the assignment at `l3` after the **`break`** ; statement is unreachable and that the program loop cannot be iterated (since the loop head `l1` is not reachable after executing the loop body as shown by the analysis `<l2: T ; l1:_|_; tt; l4: T >` of the loop body `Stmtlist`).

```
<l1: T ; l4: T ; ff; l0:_|_>   Prog:
<l1: T ; l4: T ; ff; l0:_|_>     (while l1: (0 < 1)
<l2: T ; l1:_|_; tt; l4: T >        Stmtlist: {
<l2: T ; l3:_|_; tt; l4: T >           l2: break;
<l3:_|_; l1:_|_; ff; l0:_|_>           l3: x = 1;
                                    } )
                               l4:
```

□

13 Solutions to Selected Exercises of Chapter 24

Solution to exercise 24.17 $\langle L, \sqsubseteq, \bot, \sqcup\rangle$ is a complete lattice so $\langle (L \longrightarrow L), \dot{\sqsubseteq}, \dot{\bot}, \dot{\sqcup}\rangle$ is a complete lattice, pointwise. The Galois connection $\langle (L \longrightarrow L), \dot{\sqsubseteq}\rangle \mathrel{\underset{\vec{F}}{\overset{\overleftarrow{F}}{\leftrightarrows}}} \langle (L \longrightarrow L), \dot{\sqsubseteq}\rangle$ implies that $\vec{F}$ preserves existing lubs by lemma 11.38 so is upper continuous proving that $\mathsf{lfp}^{\sqsubseteq}\, \vec{F}$ exists by Scott–Kleene's iterative fixpoint theorem 15.26. By duality, $\mathsf{gfp}^{\sqsubseteq}\, \overleftarrow{F}$ does exist.

Let us proof by recurrence on $n \in \mathbb{N}$ that $\vec{F}^n(X) \dot{\sqsubseteq} Y \Leftrightarrow X \dot{\sqsubseteq} \overleftarrow{F}^n(Y)$.

- for the basis $\vec{F}^0(X) = X \dot{\sqsubseteq} Y \Leftrightarrow X \dot{\sqsubseteq} Y = \overleftarrow{F}^0(Y)$;
- for the induction step,

$$
\begin{array}{ll}
 & \vec{F}^{n+1}(X) \dot{\sqsubseteq} Y \\
\Leftrightarrow & \vec{F}(\vec{F}^n(X)) \dot{\sqsubseteq} Y \qquad \wr\text{definition of the iterates}\int \\
\Leftrightarrow & \vec{F}^n(X) \dot{\sqsubseteq} \overleftarrow{F}(Y) \qquad \wr\text{Galois connection hypothesis}\int \\
\Leftrightarrow & X \dot{\sqsubseteq} \overleftarrow{F}^n(\overleftarrow{F}(Y)) \qquad \wr\text{recurrence hypothesis}\int \\
\Leftrightarrow & X \dot{\sqsubseteq} \overleftarrow{F}^{n+1}(Y) \qquad \wr\text{definition of the iterates}\int
\end{array}
$$

It follows that

$$
\begin{array}{ll}
 & (\mathsf{lfp}^{\sqsubseteq}\, \vec{F})(X) \sqsubseteq Y \\
\Leftrightarrow & (\dot{\bigsqcup_{n\in\mathbb{N}}} \vec{F}^n(\dot{\bot}))(X) \sqsubseteq Y \qquad \wr\text{Scott–Kleene's iterative fixpoint theorem 15.26}\int \\
\Leftrightarrow & \bigsqcup_{n\in\mathbb{N}}(\vec{F}^n(\dot{\bot})(X)) \sqsubseteq Y \qquad \wr\text{pointwise definition of } \dot{\bigsqcup}\int
\end{array}
$$

$$
\begin{array}{lr}
\Leftrightarrow \forall n \in \mathbb{N} \,.\, (\vec{F}^{n}(\bot)(X)) \sqsubseteq Y & \wr\text{definition of the lub } \bigsqcup\int \\
\Leftrightarrow \forall n \in \mathbb{N} \,.\, X \sqsubseteq \overleftarrow{F}^{n}(\bot)(Y) & \wr\forall n \in \mathbb{N} \,.\, \langle (L \to L),\ \dot{\sqsubseteq}\rangle \underset{\vec{F}^{n}}{\overset{\overleftarrow{F}^{n}}{\leftrightarrows}} \langle (L \to L),\ \dot{\sqsubseteq}\rangle\int \\
\Leftrightarrow X \sqsubseteq \displaystyle\prod_{n\in\mathbb{N}} \overleftarrow{F}^{n}(\bot)(Y) & \wr\text{definition of the glb } \prod\int \\
\Leftrightarrow X \sqsubseteq (\displaystyle\dot{\prod_{n\in\mathbb{N}}} \overleftarrow{F}^{n}(\bot))(Y) & \wr\text{pointwise definition of } \dot{\prod}\int \\
\Leftrightarrow X \sqsubseteq (\mathsf{gfp}^{\sqsubseteq}\, \overleftarrow{F})(Y) & \wr\text{dual of Scott–Kleene's iterative fixpoint theorem 15.26}\int
\end{array}
$$

□

14 Solutions to Selected Exercises of Chapter 32

Solution to exercise 32.6 $x = y = [1,3]$ and $z = [-3,-1]$ so that $x \otimes^{i} (y \oplus^{i} z) = [-3,3] = [-6,6]$ while $(z \otimes^{i} y) \oplus^{i} (z \otimes^{i} z) = [-8,8]$. □

15 Solutions to Selected Exercises of Chapter 33

Solution to exercise 33.4

```
let neq (lx,hx) (ly,hy) =
   if (lx=hx)&&(lx=ly)&&(ly=hy) then
      (* equal constants not != *) (infimum,infimum)
   else if (lx=hx)&&(hx=ly)&&(ly<hy) then ((lx,hx),(ly+1,hy))
   else if (lx=hx)&&(lx=hy)&&(ly<hy) then ((lx,hx),(ly,hy-1))
   else if (lx<hx)&&(hx=ly)&&(ly=hy) then ((lx,hx-1),(ly,hy))
   else if (lx<hx)&&(hy=lx)&&(ly=hy) then ((lx+1,hx),(ly,hy))
   else ((lx,hx),(ly,hy))
```

□

Solution to exercise 33.10 In order to simulate the precondition at ℓ_2 and observe the postcondition at ℓ_5, we analyze the following program:

```
if l1: (n < 1){i:T; n:T}
   {
      l2: {i:T; n:[-oo, 0]} i = n;
      while l3: (i != 1) {i:[-oo, 0]; n:[-oo, 0]}
         l4: {i:[-oo, 0]; n:[-oo, 0]} i = (i - 1);
      l5: {i:_|_; n:_|_} ;
   }
l6: {i:T; n:[1, oo]}
```

which shows that if initially $\mathsf{n} < 1$ at ℓ_2 then the program does not terminate at ℓ_5. □

16 Solutions to Selected Exercises of Chapter 34

Solution to exercise 34.9 Consider the interval analysis of the program

$$\ell_0 \texttt{ x = 0 ;}\texttt{\textbf{while}}\ \ell_1 \texttt{ (x < 1001) } \ell_2 \texttt{ x = x + 1 ;}$$

with loop invariant transformer $\mathscr{F}^i(x) = [0,0] \sqcup^i ((x \sqcap^i [-\infty, 1000]) \oplus^i [1,1])$. The iterates with widening discussed in section 33.5 converge to $[0, \infty]$. Consider the less precise transformer $\mathscr{F}^i(x) = [0,1001] \sqcup^i ((x \sqcap^i [-\infty, 1000]) \oplus^i [1,1])$ where 0 is abstracted into $[0,1001]$ instead of the more precise $[0,0]$. The iterates with widening of section 33.5 now converge to $[0,1001]$. □

Solution to exercise 34.13 For the program `b=0; x=1; while (0<1) { if (b == 0) { b=1; x=0; } else x=x+1; }}`, the successor widening yields the loop invariant `[b:[0, oo]; x:T]`, and the widening delayed 3 iterations yields `[b:[0, 1]; x:[0, oo]]`. □

Solution to exercise 36.11

— $\overline{\mathbb{P}}_1 \mathrel{\widehat{\lesseqgtr}} \overline{\mathbb{P}}_2$

$\Rightarrow \forall \overline{P}_2 \in \overline{\mathbb{P}}_2 \,.\, \exists \overline{P}_1 \in \overline{\mathbb{P}}_1 \,.\, \gamma_1(\overline{P}_1) = \gamma_2(\overline{P}_2)$ $\quad \wr$definition of $\widehat{\lesseqgtr}$$\int$

$\Rightarrow \forall P_2 \in \mathbb{P} \,.\, \exists \overline{P}_1 \in \overline{\mathbb{P}}_1 \,.\, \gamma_1(\overline{P}_1) = \gamma_2(\alpha_2(P_2))$ $\quad \wr$because $\alpha_2(P_2) \in \overline{\mathbb{P}}_2$$\int$

$\Rightarrow \forall P_2 \in \mathbb{P} \,.\, \exists \overline{P}_1 \in \overline{\mathbb{P}}_1 \,.\, \gamma_1 \circ \alpha_1 \circ \gamma_1(\overline{P}_1) = \gamma_2 \circ \alpha_2(P_2)$

$\wr \gamma_1 \circ \alpha_1 \circ \gamma_1 = \gamma_1$ in Galois connection and definition of $\circ$$\int$

$\Rightarrow \forall P_2 \in \mathbb{P} \,.\, \exists P_1 \in \mathbb{P} \,.\, \gamma_1 \circ \alpha_1(P_1) = \gamma_2 \circ \alpha_2(P_2)$ $\quad \wr$taking $P_1 = \gamma_1(\overline{P}_1)$$\int$

$\Rightarrow \gamma_2 \circ \alpha_2(\mathbb{P}) \subseteq \gamma_1 \circ \alpha_1(\mathbb{P})$ $\quad \wr$definition of $\subseteq$$\int$

— Conversely, for all $\overline{P}_2 \in \overline{\mathbb{P}}_2$ then $\gamma_2(\overline{P}_2) \in \mathbb{P}$ so

$\exists P_1 \in \mathbb{P} \,.\, \gamma_1 \circ \alpha_1(P_1) = \gamma_2 \circ \alpha_2(\gamma_2(\overline{P}_2)) = \gamma_2(\overline{P}_2)$ $\wr$hyp. and $\gamma_2 \circ \alpha_2 \circ \gamma_2 = \gamma_2$ in GC$\int$

$\Rightarrow \exists \overline{P}_1 \in \overline{\mathbb{P}}_1 \,.\, \gamma_1(\overline{P}_1) = \gamma_2(\overline{P}_2)$ $\quad \wr$choosing $\overline{P}_1 = \alpha_1(P_1)$$\int$

$\Rightarrow \overline{\mathbb{P}}_1 \mathrel{\widehat{\lesseqgtr}} \overline{\mathbb{P}}_2$ $\quad \wr$definition of $\widehat{\lesseqgtr}$$\int$

□

17 Solutions to Selected Exercises of Chapter 37

Solution to exercise 37.8 The column of $\vec{x}_i$ in the reduced row echelon form of $(\mathbf{A}|\vec{b})$ is zero except for a one in some row ℓ of $\mathbf{A}$, this row ℓ of $\mathbf{A}$ is zero but for the one in the column of $\vec{x}_i$, in which case the constant is equal to $\vec{b}_i$. □

Solution to exercise 37.20 Assume W is generated by $\langle \mathbf{B},\ \vec{x}_0 \rangle$.

$\{\vec{x} \mid \exists v \in \mathbb{F} \,.\, \vec{x}[i \leftarrow v] \in W\}$

$$= \{\vec{x} \mid \exists v \in \mathbb{F} \,.\, \exists \vec{a} \in \mathbb{F}^m \,.\, \vec{x}[i \leftarrow v] = \vec{x}_0 + \sum_{j\in[1,m]} \vec{a}_j \mathbf{B}_j\} \qquad \wr W \text{ is generated by } \langle \mathbf{B},\ \vec{x}_0\rangle \wr$$

$$= \{\vec{x} \mid \exists v \in \mathbb{F} \,.\, \exists \vec{a} \in \mathbb{F}^m \,.\, \forall k \in [1,m] \setminus \{i\} \,.\, \vec{x}_k = (\vec{x}_0 + \sum_{j\in[1,m]} \vec{a}_j \mathbf{B}_j)_k \wedge \vec{x}_i = v\}$$
$$\wr \text{definition of } \vec{x}[i \leftarrow v] \wr$$

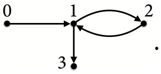
$$= \{\vec{x} \mid \exists \vec{a}_{m+1} \in \mathbb{F} \,.\, \exists \vec{a} \in \mathbb{F}^m \,.\, \forall k \in [1,m] \setminus \{i\} \,.\, \vec{x}_k = (\vec{x}_0 + \sum_{j\in[1,m]} \vec{a}_j \mathbf{B}_j)_k \wedge \vec{x}_i = (\vec{x}_0 + \sum_{j\in[1,m]} \vec{a}_j \mathbf{B}_j)_i + \vec{a}_{m+1}\} \qquad \wr \text{letting } \vec{a}_{m+1} = v - (\vec{x}_0 + \textstyle\sum_{j\in[1,m]} \vec{a}_j \mathbf{B}_j)_i \wr$$

$$= \{\vec{x}_0 + \sum_{j\in[1,m]} \vec{a}_j \mathbf{B}_j + \vec{a}_{m+1}\vec{0}[i \leftarrow 1]_j \mid \vec{a} \in \mathbb{F}^{m+1}\} \qquad \wr \text{grouping terms} \wr$$

$$= \vec{x}_0 \mathbin{\vec{+}} \mathsf{Span}(\langle (\mathbf{B}_j, \vec{0}[i \leftarrow 1]),\ j \in [1,m]\rangle) \qquad \wr \text{definition of span in section 37.2.3} \wr$$

□

18 Solutions to Selected Exercises of Chapter 39

Solution to exercise 39.29 Consider the following graph . Initially, $12 \in \widehat{\mathscr{F}_\pi}^0(1,2)$, $13 \in \widehat{\mathscr{F}_\pi}^0(1,3)$ and $21 \in \widehat{\mathscr{F}_\pi}^0(2,1)$. The next iterate is identical because there is no path through 0. The next iterate through $1 \notin \{2,3\}$ adds $21 \odot 13 = 213 \in \widehat{\mathscr{F}_\pi}^2(2,3)$. The next iterate through $2 \notin \{1,3\}$ adds $12 \odot 213 = 1213 \in \widehat{\mathscr{F}_\pi}^3(1,3)$ which is not elementary and so does not belong to $\mathsf{p}^{\ni}(1,3)$. □

Solution to exercise 39.45

```
$ cat rfw.c
#include <limits.h>
#include <stdio.h>
int main () {
#define N 3
#define INF INT_MAX
    int D[N][N] = {{INF, 1, 2}, {-1, INF, 2}, {INF, INF, 1}};
    int i,j,k,dikj,negativecycle;

    for (i=0; i<N; i++) { D[i][i] = 0; }
    for (k=0; k<N; k++)
        for (i=0; i<N; i++)
            for (j=0; j<N; j++) {
           dikj = (D[i][k]==INF | D[k][j]==INF ? INF : D[i][k]+D[k][j]);
                if (dikj < D[i][j])
                    D[i][j] = dikj;
            }
    negativecycle = 0;
    for (i=0; i<N; i++) {
        if (D[i][i]<0) negativecycle = 1;
    }
   if (negativecycle) printf(”cycle of strictly negative length”); else
        for (i=0; i<N; i++) {
            for (j=0; j<N; j++)
               (D[i][j]==INF ? printf(”oo ”) : printf(”%i ”, D[i][j]));
            printf (”\n”);
        }
}
$ gcc rfw.c
$ ./a.out
0 1 2
-1 0 1
oo oo 0
```

□

19 Solutions to Selected Exercises of Chapter 41

Solution to exercise 41.23

$$
\begin{aligned}
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{Sl}\ \ell]\!]\ L_e &\triangleq \widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{Sl}\ \ell]\!]\ \varnothing, L_e \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{x = E ;}]\!]\ L_b, L_e &\triangleq \mathbb{use}[\![\mathtt{x = E}]\!] \cup (L_e \setminus \mathbb{mod}[\![\mathtt{x = E}]\!]) \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{;}]\!]\ L_b, L_e &\triangleq L_e \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{Sl}'\ \mathtt{S}]\!]\ L_b, L_e &\triangleq \widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{Sl}']\!]\ L_b, (\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{S}]\!]\ L_b, L_e) \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\ \epsilon\ ]\!]\ L_b, L_e &\triangleq L_e \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathbf{if}\ (\mathtt{B})\ \mathtt{S}_t]\!]\ L_b, L_e &\triangleq \mathbb{use}[\![\mathtt{B}]\!] \cup (L_e \cap \widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{S}_t]\!]\ L_b, L_e) \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathbf{if}\ (\mathtt{B})\ \mathtt{S}_t\ \mathbf{else}\ \mathtt{S}_f]\!]\ L_b, L_e &\triangleq \mathbb{use}[\![\mathtt{B}]\!] \cup (\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{S}_t]\!]\ L_b, L_e \cap \widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{S}_f]\!]\ L_b, L_e) \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathbf{while}\ (\mathtt{B})\ \mathtt{S}_b]\!]\ L_b, L_e &\triangleq \mathbb{use}[\![\mathtt{B}]\!] \cup (L_e \cap \widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{S}_b]\!]\ L_b, L_e) \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathbf{break}\ \mathtt{;}]\!]\ L_b, L_e &\triangleq L_b \\
\widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\{\ \mathtt{Sl}\ \}]\!]\ L_b, L_e &\triangleq \widehat{\mathcal{S}}^{\forall\mathbb{l}}[\![\mathtt{Sl}]\!]\ L_b, L_e
\end{aligned}
\tag{6}
$$

For (un)soundness, notice that $\mathbb{use}[\![\mathtt{B}]\!]$ does *not* guarantee that a variable is used in `B` (e.g. for `(x-x)==0`). Only a semantic underapproximation would be formally correct.
□

Solution to exercise 41.11 By choosing $v \neq \rho(\mathtt{y})$, we have $\rho(\mathtt{y}) = \mathcal{A}[\![\mathtt{A}]\!]\,\rho \neq \mathcal{A}[\![\mathtt{A}]\!]\,\rho[\mathtt{y} \leftarrow v] = v$. So use$[\![\mathtt{x = y}]\!]\,\rho \triangleq \{\mathtt{y} \mid \rho(\mathtt{x}) \neq \rho(\mathtt{y})\}$ because when $\rho(\mathtt{x}) = \rho(\mathtt{y})$ the assignment can be skipped. □

20 Solutions to Selected Exercises of Chapter 43

Solution to exercise 43.8

program $\mathsf{P} ::= \mathtt{Sl}\ \ell$

$$\widehat{\mathcal{S}}^{\tau}[\![\mathsf{P}]\!] \quad = \quad \widehat{\mathcal{S}}^{\tau}[\![\mathtt{Sl}]\!] \tag{7}$$

empty statement list $\mathtt{Sl} ::= \epsilon$

$$\widehat{\mathcal{S}}^{\tau}[\![\mathtt{Sl}]\!] \quad = \quad \varnothing \tag{8}$$

skip statement $\mathsf{S} ::= \ell;$

$$\widehat{\mathcal{S}}^{\tau}[\![\mathsf{S}]\!] \quad = \quad \{\langle \ell,\ \rho\rangle \longrightarrow \langle \mathsf{after}[\![\mathsf{S}]\!],\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} \tag{9}$$

conditional statement $\mathsf{S} ::= \mathtt{if}\ \ell\ (\mathsf{B})\ \mathsf{S}_t$

$$\begin{aligned}\widehat{\mathcal{S}}^{\tau}[\![\mathsf{S}]\!] \quad = \quad & \{\langle \ell,\ \rho\rangle \longrightarrow \langle \mathsf{after}[\![\mathsf{S}]\!],\ \rho\rangle \mid \mathcal{B}[\![\mathsf{B}]\!]\,\rho = \mathrm{ff}\} \cup \\ & \{\langle \ell,\ \rho\rangle \longrightarrow \langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle \mathbin{\widehat{\cdot}} \pi_2 \mid \mathcal{B}[\![\mathsf{B}]\!]\,\rho = \mathrm{tt}\} \cup \widehat{\mathcal{S}}^{\tau}[\![\mathsf{S}_t]\!]\end{aligned} \tag{10}$$

conditional statementS $::= \mathtt{if}\ \ell\ (\mathsf{B})\ \mathsf{S}_t\ \ \mathtt{else}\ \ \mathsf{S}_f$

$$\begin{aligned}\widehat{\mathcal{S}}^{\tau}[\![\mathsf{S}]\!] \quad = \quad & \{\langle \ell,\ \rho\rangle \longrightarrow \langle \mathsf{at}[\![\mathsf{S}_t]\!],\ \rho\rangle \mid \mathcal{B}[\![\mathsf{B}]\!]\,\rho = \mathrm{tt}\} \cup \widehat{\mathcal{S}}^{\tau}[\![\mathsf{S}_t]\!] \cup \\ & \{\langle \ell,\ \rho\rangle \longrightarrow \langle \mathsf{at}[\![\mathsf{S}_f]\!],\ \rho\rangle \mathbin{\widehat{\cdot}} \pi_2 \mid \mathcal{B}[\![\mathsf{B}]\!]\,\rho = \mathrm{ff}\} \cup \widehat{\mathcal{S}}^{\tau}[\![\mathsf{S}_f]\!]\end{aligned} \tag{11}$$

break statement $\mathsf{S} ::= \ell\ \mathtt{break}\ ;$

$$\widehat{\mathcal{S}}^{\tau}[\![\mathsf{S}]\!] \quad = \quad \{\langle \ell,\ \rho\rangle \longrightarrow \langle \mathsf{break\text{-}to}[\![\mathsf{S}]\!],\ \rho\rangle \mid \rho \in \mathbb{E}\mathbb{v}\} \tag{12}$$

compound statement $\mathsf{S} ::= \{\ \mathtt{Sl}\ \}$

$$\widehat{\mathcal{S}}^{\tau}[\![\mathsf{S}]\!] \quad = \quad \widehat{\mathcal{S}}^{\tau}[\![\mathtt{Sl}]\!] \tag{13}$$

□

Solution to exercise 43.12 The float interval transition semantics $\widehat{\mathcal{S}}^{\tau}_{\mathbb{P}^i_{\mathbb{F}}}$ is similar to $\widehat{\mathcal{S}}^{\tau}$ except in the nondeterministic handling of tests in conditional and iteration statements. For example,

$$\begin{aligned}\widehat{\mathcal{S}}^{\tau}_{\mathbb{P}^i_{\mathbb{F}}}[\![\mathtt{while}\ \ell\ (\mathsf{B})\ \mathsf{S}_b]\!] \quad = \quad & \{\langle \ell,\ \overline{\rho}\rangle \longrightarrow \langle \mathsf{after}[\![\mathsf{S}]\!],\ \overline{\rho}_{\mathrm{ff}}\rangle \mid \exists \overline{\rho}_{\mathrm{tt}}\ .\ \mathcal{B}^i_{\mathbb{F}}[\![\mathsf{B}]\!]\overline{\rho} = \langle \overline{\rho}_{\mathrm{tt}},\ \overline{\rho}_{\mathrm{ff}}\rangle\} \\ \cup\, & \{\langle \ell,\ \overline{\rho}\rangle \longrightarrow \langle \mathsf{at}[\![\mathsf{S}_b]\!],\ \overline{\rho}_{\mathrm{tt}}\rangle \mid \exists \overline{\rho}_{\mathrm{ff}}\ .\ \mathcal{B}^i_{\mathbb{F}}[\![\mathsf{B}]\!]\overline{\rho} = \langle \overline{\rho}_{\mathrm{tt}},\ \overline{\rho}_{\mathrm{ff}}\rangle\} \cup \widehat{\mathcal{S}}^{\tau}_{\mathbb{P}^i_{\mathbb{F}}}[\![\mathsf{S}_b]\!]\end{aligned} \tag{14}$$

□

21 Solutions to Selected Exercises of Chapter 44

Solution to exercise 44.20

Proof of lemma 44.19 The proof that $\mathsf{R}' \in \mathbb{R}^{\dagger}$ is $\mid$-free is by structural on R, observing that the definition (44.18) of fstnxt involves no alternative $\mid$. The proof that $\mathsf{R} \Bumpeq \mathtt{L}\ :\ \mathsf{B} \bullet \mathsf{R}'$ that is $\mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!] = \mathcal{S}^{\mathsf{r}}[\![\mathtt{L}\ :\ \mathsf{B} \bullet \mathsf{R}']\!]$ is by structural on R.

- Let us first prove that $\ni$ is the neutral element of $\bullet$.

$$\begin{aligned} & \mathcal{S}^{\mathsf{r}}[\![\mathsf{R} \bullet \varepsilon]\!] \\ = \ & \{\langle \underline{\varrho},\ \pi \cdot \pi'\rangle \mid \langle \underline{\varrho},\ \pi\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!] \wedge \langle \underline{\varrho},\ \pi'\rangle \in \mathcal{S}^{\mathsf{r}}[\![\varepsilon]\!]\} && \text{≀(44.7)≀} \\ = \ & \{\langle \underline{\varrho},\ \pi \cdot \ni\rangle \mid \langle \underline{\varrho},\ \pi\rangle \in \mathcal{S}^{\mathsf{r}}[\![\mathsf{R}]\!]\} && \text{≀because } \pi' = \ni \text{ by (44.7)≀}\end{aligned}$$

$= \boldsymbol{\mathcal{S}}^{\mathsf{r}}[\![\mathsf{R}]\!]$ $\qquad\wr$definition of concatenation $\cdot$ and $\in\int$

Similarly $\varepsilon \bullet \mathsf{R} \cdot \mathsf{R}$ and this extends to all $\mathsf{R}' \in \mathbb{R}_{\varepsilon}$.

- It follows that lemma 44.19 holds for $\mathsf{fstnxt}(\mathsf{L} : \mathsf{B})$ and $\mathsf{fstnxt}(\mathsf{R}_1\mathsf{R}_2)$ when $\mathsf{R}_1 \in \mathbb{R}_{\varepsilon}$.
- For $\mathsf{fstnxt}(\mathsf{R}_1\mathsf{R}_2)$ when $\mathsf{R}_1 \notin \mathbb{R}_{\varepsilon}$, there are two cases.
  - Either $\mathsf{R}_1^n \in \mathbb{R}_{\varepsilon}$ and then

    $\mathsf{R}_1^f \bullet \mathsf{R}_2$

    $\Bumpeq \mathsf{R}_1^f \bullet \mathsf{R}_1^n \bullet \mathsf{R}_2$ $\qquad\wr$because $\mathsf{R}_1^n \in \mathbb{R}_{\varepsilon}$ so $\mathsf{R}_1^f \bullet \mathsf{R}_1^n \Bumpeq \mathsf{R}_1^f\int$

    $\Bumpeq \mathsf{R}_1 \bullet \mathsf{R}_2$ $\qquad\wr$by induction hypothesis because $\langle \mathsf{R}_1^f,\ \mathsf{R}_1^n\rangle = \mathsf{fstnxt}(\mathsf{R}_1)$, *Q.E.D.*$\int$
  - Otherwise $\mathsf{R}_1^n \notin \mathbb{R}_{\varepsilon}$ and then

    $\mathsf{R}_1^f \bullet \mathsf{R}_1^n \bullet \mathsf{R}_2$

    $\Bumpeq \mathsf{R}_1 \bullet \mathsf{R}_2$ $\qquad\wr$by induction hypothesis because $\langle \mathsf{R}_1^f,\ \mathsf{R}_1^n\rangle = \mathsf{fstnxt}(\mathsf{R}_1)$, *Q.E.D.*$\int$
- For $\mathsf{fstnxt}(\mathsf{R}^+)$, let $\langle \mathsf{R}^f,\ \mathsf{R}^n\rangle = \mathsf{fstnxt}(\mathsf{R})$. There are two cases.
  - Either $\mathsf{R}^n \in \mathbb{R}_{\varepsilon}$ and then

    $\mathsf{R}^f \bullet \mathsf{R}^*$

    $\Bumpeq \mathsf{R}^f \bullet \mathsf{R}^n \bullet \mathsf{R}^*$ $\qquad\wr$because $\boldsymbol{\mathcal{S}}^{\mathsf{r}}[\![\mathsf{R}^n]\!] = \boldsymbol{\mathcal{S}}^{\mathsf{r}}[\![\varepsilon]\!]\int$

    $\Bumpeq \mathsf{R} \bullet \mathsf{R}^*$ $\qquad\wr$induction hypothesis because $\langle \mathsf{R}^f,\ \mathsf{R}^n\rangle = \mathsf{fstnxt}(\mathsf{R})\int$

    $\Bumpeq \mathsf{R}^+$ $\qquad\wr$definition (44.7) of $\boldsymbol{\mathcal{S}}^{\mathsf{r}}[\![\mathsf{R}^*]\!]$ and $\boldsymbol{\mathcal{S}}^{\mathsf{r}}[\![\mathsf{R}^+]\!]\int$
  - Otherwise $\mathsf{R}^n \notin \mathbb{R}_{\varepsilon}$ and then $\mathsf{R}^f \bullet \mathsf{R}^n \bullet \mathsf{R}^* \Bumpeq \mathsf{R}$, as shown previously.
- The last case for $\mathsf{fstnxt}((\mathsf{R}))$ follows by structural induction from $\boldsymbol{\mathcal{S}}^{\mathsf{r}}[\![(\mathsf{R})]\!] \triangleq \boldsymbol{\mathcal{S}}^{\mathsf{r}}[\![\mathsf{R}]\!]$.

□

Solution to exercise 44.33 Define $\gamma_{\mathscr{M}^{t}\langle\underline{\varrho},\,\mathsf{R}\rangle}(M) \triangleq \{\pi \mid \forall \mathsf{R}' \in \mathbb{R}\ .\ (\langle \mathsf{tt},\ \mathsf{R}'\rangle = \mathscr{M}^{t}\langle\rho,\ \mathsf{R}\rangle(\pi)) \Rightarrow (\pi \in M)\}$. □

Solution to exercise 44.54

— Let us first prove that $X \mapsto \vec{\tau} \frown X$ preserves arbitrary joins. If $\vec{\tau}$ is $\varnothing$, this is $\varnothing$ whichever is X. Because $\tau \in \wp(\mathbb{S} \times \mathbb{S})$, we cannot have $\tau = \ni$. Otherwise, if X is empty then $\vec{\tau} \frown \varnothing = \varnothing$. For $\Delta = \varnothing$, $\vec{\tau} \frown \bigcup_{i\in\varnothing} X_i = \vec{\tau} \frown \varnothing = \varnothing = \bigcup_{i\in\varnothing} \vec{\tau} \frown X_i$. Otherwise, assuming $\Delta \neq \varnothing$, we have

$$\vec{\tau} \frown (\bigcup_{i\in\Delta} X_i)$$

$$= \{\vec{\tau} \mid \ni \in \bigcup_{i\in\Delta} X_i\} \cup \{\sigma\sigma'\pi \mid \langle\sigma,\ \sigma'\rangle \in \tau \wedge \sigma'\pi \in \bigcup_{i\in\Delta} X_i\} \qquad \wr\text{definitions of} \frown \text{and } \vec{\tau}\int$$

$$= \bigcup_{i\in\Delta}\{\vec{\tau} \mid \ni\, \in X_i\} \cup \bigcup_{i\in\Delta}\{\sigma\sigma' \mathbin{\overset{\frown}{\cdot}} \sigma'\pi \mid \langle\sigma,\, \sigma'\rangle \in \tau \wedge \sigma'\pi \in X_i\} \qquad \wr\,\text{definitions of }\bigcup\text{ and }\mathbin{\overset{\frown}{\cdot}}\,\int$$

$$= \bigcup_{i\in\Delta}\big(\{\vec{\tau} \mid \ni\, \in X_i\} \cup \{\sigma\sigma' \mathbin{\overset{\frown}{\cdot}} \sigma'\pi \mid \langle\sigma,\, \sigma'\rangle \in \tau \wedge \sigma'\pi \in X_i\}\big) \qquad \wr\,\text{definition of }\bigcup\,\int$$

$$= \bigcup_{i\in\Delta}(\vec{\tau} \mathbin{\overset{\frown}{\cdot}} X_i) \qquad \wr\,\text{definitions of }\mathbin{\overset{\frown}{\cdot}}\text{ and }\vec{\tau}\,\int$$

It follows that $X \mapsto \mathbb{S}^1 \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} X$ preserves nonempty joins.

$$\mathbb{S}^1 \cup (\vec{\tau} \mathbin{\overset{\frown}{\cdot}} \bigcup_{i\in\Delta} X_i)$$

$$= \mathbb{S}^1 \cup \bigcup_{i\in\Delta}(\vec{\tau} \mathbin{\overset{\frown}{\cdot}} X_i) \qquad \wr\,\text{as shown previously}\,\int$$

$$= \bigcup_{i\in\Delta}(\mathbb{S}^1 \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} X_i) \qquad \wr\,\bigcup\text{ associative}\,\int$$

It does not preserve empty joins because $\mathbb{S}^1 \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} \bigcup_{i\in\varnothing} X_i = \mathbb{S}^1 \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} \varnothing = \mathbb{S}^1 \neq \varnothing = \bigcup_{i\in\varnothing}(\mathbb{S}^1 \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} X_i)$.

— By recurrence on n.

– for $n = 0$,

$$X^0$$

$$= \varnothing \qquad \wr\,\text{definition of iterates from }\varnothing\,\int$$

$$= \bigcup_{0}\varnothing \qquad \wr\,\text{definition of }\bigcup\,\int$$

$$= \bigcup_{i=1}\boldsymbol{\mathcal{S}}^i_{\mathrm{t}}[\![\tau]\!] \qquad \wr\,\text{definition of }\textstyle\bigcup_{i=1}^{j} x_i = \varnothing\text{ when } j < i\,\int$$

– for $n = 1$,

$$X^1$$

$$= \mathbb{S}^1 \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} X^0 \qquad \wr\,\text{definition of the iterates}\,\int$$

$$= \mathbb{S}^1 \qquad \wr\, X^0 = \varnothing\text{ and definition of }\mathbin{\overset{\frown}{\cdot}}\,\int$$

$$= \underset{1}{\boldsymbol{\mathcal{S}}^1_{\mathrm{t}}}[\![\tau]\!] \qquad \wr\,\boldsymbol{\mathcal{S}}^1_{\mathrm{t}}[\![\tau]\!] = \mathbb{S}^1 \triangleq \{\pi \in \mathbb{S}^1 \mid \pi_0 = \iota_0\}\,\int$$

$$= \bigcup_{i=1}\boldsymbol{\mathcal{S}}^i_{\mathrm{t}}[\![\tau]\!] \qquad \wr\,\text{definition of }\textstyle\bigcup_{i=1}^{j} x_i = x_1\text{ with } j = i\,\int$$

— for the induction, assume that $X^n = \bigcup_{i=1}^{n} \boldsymbol{\mathcal{S}}^i_{\mathrm{t}}[\![\tau]\!]$, by induction hypothesis Then

$$X^{n+1}$$

$$= \mathbb{S}^1 \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} X^n \qquad \wr\,\text{definition of the iterates}\,\int$$

$$= \mathbb{S}^1 \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} (\bigcup_{i=1}^{n}\boldsymbol{\mathcal{S}}^i_{\mathrm{t}}[\![\tau]\!]) \qquad \wr\,\text{induction hypothesis}\,\int$$

$$= \mathbb{S}^1 \cup \bigcup_{i=1}^{n}(\vec{\tau} \mathbin{\overset{\frown}{\cdot}} \boldsymbol{\mathcal{S}}^i_{\mathrm{t}}[\![\tau]\!]) \qquad \wr\, X \mapsto \vec{\tau} \mathbin{\overset{\frown}{\cdot}} X\text{ preserves arbitrary joins, as shown previously}\,\int$$

$$
\begin{aligned}
&= \boldsymbol{\mathcal{S}}^{1}_{\mathrm{t}}[\![\tau]\!] \cup \bigcup_{i=1}^{n}(\boldsymbol{\mathcal{S}}^{i+1}_{\mathrm{t}}[\![\tau]\!]) && \wr\, \mathbb{S}^{1} = \boldsymbol{\mathcal{S}}^{1}_{\mathrm{t}}[\![\tau]\!] \text{ and } \boldsymbol{\mathcal{S}}^{i+1}_{\mathrm{t}}[\![\tau]\!] = \boldsymbol{\mathcal{S}}^{i}_{\mathrm{t}}[\![\tau]\!] \mathbin{\overset{\frown}{\cdot}} \vec{\tau} \,\int \\
&= \boldsymbol{\mathcal{S}}^{1}_{\mathrm{t}}[\![\tau]\!] \cup \bigcup_{j=2}^{n+1}(\boldsymbol{\mathcal{S}}^{j}_{\mathrm{t}}[\![\tau]\!]) && \wr\, \text{letting } j = i + 1 \,\int \\
&= \bigcup_{j=1}^{n+1}(\boldsymbol{\mathcal{S}}^{j}_{\mathrm{t}}[\![\tau]\!]) && \wr\, \text{incorporating the term } j = 1 \,\int
\end{aligned}
$$

— Let us apply Scott–Kleene's iterative fixpoint theorem 15.26.

$$
\begin{aligned}
\vec{X}^{\infty} &\triangleq \bigcup_{n\in\mathbb{N}} \vec{X}^{n} && \wr\, \text{definition of the iterates } \vec{X}^{n} \text{ of } X \mapsto \mathbb{S}^{1} \cup X \mathbin{\overset{\frown}{\cdot}} \vec{\tau} \text{ from } \varnothing \,\int \\
&= \bigcup_{n\in\mathbb{N}} \bigcup_{i=1}^{n} \boldsymbol{\mathcal{S}}^{i}_{\mathrm{t}}[\![\tau]\!] && \wr\, X^{n} = \textstyle\bigcup_{i=1}^{n} \boldsymbol{\mathcal{S}}^{i}_{\mathrm{t}}[\![\tau]\!], \text{ as shown previously} \,\int \\
&= \bigcup_{n\in\mathbb{N}} \boldsymbol{\mathcal{S}}^{n}_{\mathrm{t}}[\![\tau]\!] && \wr\, \text{definition of } \textstyle\bigcup \,\int \\
&= \boldsymbol{\mathcal{S}}_{\mathrm{t}}[\![\tau]\!] && \wr\, \text{definition of } \boldsymbol{\mathcal{S}}_{\mathrm{t}}[\![\tau]\!] \text{ in (44.55)} \,\int
\end{aligned}
$$

which is $\mathsf{lfp}^{\subseteq}\, X \mapsto \mathbb{S}^{1} \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} X$ by Scott–Kleene's iterative fixpoint theorem 15.26 knowing that $X \mapsto \mathbb{S}^{1} \cup \vec{\tau} \mathbin{\overset{\frown}{\cdot}} X$ is preserves nonempty joins and therefore is continuous and $\langle \mathbb{S}^{*}, \subseteq\rangle$ is a complete lattice hence a CPO. □

Solution to exercise 44.60 We have $\alpha^{\mathscr{T}}(\varnothing)\langle\sigma, \Sigma\rangle = \mathtt{tt}$ and $\langle\sigma, \Sigma\rangle \mapsto \mathtt{tt}$ is the infimum for $\stackrel{.}{\Leftarrow}$. Otherwise, for $\Delta \neq \varnothing$,

$$
\begin{aligned}
&\alpha^{\mathscr{T}}((\bigcup_{i\in\Delta} X_{i})\langle\sigma, \Sigma\rangle \\
\Leftrightarrow\ & (\{\pi \in \bigcup_{i\in\Delta} X_{i} \mid \pi_{0} = \sigma\} \subseteq \alpha^{\mathbb{T}}(\{P \in \boldsymbol{\mathcal{S}}[\![T]\!] \mid P_{0} = \Sigma\})) \Leftarrow b && \wr\, \text{definition (44.62) of } \alpha^{\mathscr{T}} \,\int \\
\Leftrightarrow\ & (\bigcup_{i\in\Delta}\{\pi \in X_{i} \mid \pi_{0} = \sigma\} \subseteq \alpha^{\mathbb{T}}(\{P \in \boldsymbol{\mathcal{S}}[\![T]\!] \mid P_{0} = \Sigma\})) \Leftarrow b && \wr\, \text{definition of } \textstyle\bigcup \,\int \\
\Leftrightarrow\ & \bigwedge_{i\in\Delta}(\{\pi \in X_{i} \mid \pi_{0} = \sigma\} \subseteq \alpha^{\mathbb{T}}(\{P \in \boldsymbol{\mathcal{S}}[\![T]\!] \mid P_{0} = \Sigma\}) \Leftarrow b) && \wr\, \text{definition of} \subseteq \,\int \\
\Leftrightarrow\ & \bigwedge_{i\in\Delta} \alpha^{\mathscr{T}}(X_{i})\langle\sigma, \Sigma\rangle && \wr\, \text{definition (44.62) of } \alpha^{\mathscr{T}} \,\int
\end{aligned}
$$

proving $X \mapsto \alpha^{\mathscr{T}}(X)\langle\sigma, \Sigma\rangle$ preserves arbitrary joins in the complete lattice $\langle\mathbb{B}, \Leftarrow, \mathtt{tt}, \mathtt{ff}, \bigwedge, \bigvee\rangle$, hence by exercise 11.39, $\forall\sigma \in \mathbb{S} \,.\, \forall\Sigma \in \wp(\mathbb{S}) \,.\, \langle\wp(\mathbb{S})^{*}, \subseteq\rangle \xleftarrow[X\mapsto\alpha^{\mathscr{T}}(X)\langle\sigma,\Sigma\rangle]{Y\mapsto\gamma^{\mathscr{T}}(Y)\langle\sigma,\Sigma\rangle} \langle\mathbb{B}, \Leftarrow\rangle$. The pointwise extension $\langle(\mathbb{S} \times \wp(\mathbb{S})) \to \wp(\mathbb{S})^{*}, \subseteq\rangle \xleftarrow[\alpha^{\mathscr{T}}]{\gamma^{\mathscr{T}}} \langle(\mathbb{S} \times \wp(\mathbb{S})) \to \mathbb{B}, \stackrel{.}{\Leftarrow}\rangle$ follows by exercise 11.21. □

Solution to exercise 44.63

— $\alpha^{\mathcal{T}}(\mathbb{S}^1 \cup \vec{\tau} \,\overset{\frown}{\cdot}\, X)\langle\sigma,\ \Sigma\rangle$

$= \alpha^{\mathcal{T}}(\mathbb{S}^1)\langle\sigma,\ \Sigma\rangle \wedge \alpha^{\mathcal{T}}(\vec{\tau} \,\overset{\frown}{\cdot}\, X)\langle\sigma,\ \Sigma\rangle$ ≀$X \mapsto \alpha^{\mathcal{T}}(X)\langle\sigma,\ \Sigma\rangle$ preserves joins∫ (A)

The first term of (A) is

$\alpha^{\mathcal{T}}(\mathbb{S}^1)\langle\sigma,\ \Sigma\rangle$

$= \{\pi \in \mathbb{S}^1 \mid \pi_0 = \sigma\} \subseteq \alpha^{\mathbb{T}}(\{P \in \boldsymbol{S}\llbracket T\rrbracket \mid P_0 = \Sigma\})$ ≀definition (44.62) of $\alpha^{\mathcal{T}}$∫

$= \{\pi \in \mathbb{S}^1 \mid \pi_0 = \sigma\} \subseteq \bigcup\{\alpha^{\mathbb{T}}(P) \mid P \in \{P \in \boldsymbol{S}\llbracket T\rrbracket \mid P_0 = \Sigma\}\}$ ≀definition (44.59) of $\alpha^{\mathbb{T}}$∫

$= \{\pi \in \mathbb{S}^1 \mid \pi_0 = \sigma\} \subseteq \bigcup\{\alpha^{\mathbb{T}}(P) \mid P \in \boldsymbol{S}\llbracket T\rrbracket \wedge P_0 = \Sigma\}$ ≀definition of $\in$∫

$= \{\pi \in \mathbb{S}^1 \mid \pi_0 = \sigma\} \subseteq \{\pi \in \mathbb{S}^n \mid n \in \mathbb{N}^+ \wedge \exists P \in \boldsymbol{S}\llbracket T\rrbracket \,.\, P_0 = \Sigma \wedge \forall i \in [0, n[\,.\, \pi_i \in P_i\}$

≀definition (44.58) of $\alpha^{\mathbb{T}}(P) \triangleq \bigcup_{n\in\mathbb{N}^+}\{\pi \in \mathbb{S}^n \mid \forall i \in [0, n[\,.\, \pi_i \in P_i\}$∫

$= \{\pi \in \mathbb{S}^1 \mid \pi_0 = \sigma\} \subseteq \{\pi \in \mathbb{S}^1 \mid \exists P \in \boldsymbol{S}\llbracket T\rrbracket \,.\, P_0 = \Sigma \wedge \pi_0 \in P_0\}$

≀definition of $\subseteq$ so that the traces must have the same length∫

$= \exists P \in \boldsymbol{S}\llbracket T\rrbracket \,.\, \sigma \in P_0 = \Sigma$ ≀definitions of $\Rightarrow$ and $\subseteq$∫

$= \exists P \in \{P \in \wp(\mathbb{S})^\infty \mid \forall i \in \mathbb{N} \,.\, \langle P_i,\ P_{i+1}\rangle \in T\} \,.\, \sigma \in P_0 = \Sigma$ ≀definition (44.57) of $\boldsymbol{S}\llbracket T\rrbracket$∫

$= \sigma \in \Sigma$ ≀T is total and $\mathbb{S}$ not empty∫

The second term of (A) is

$\alpha^{\mathcal{T}}(\vec{\tau} \,\overset{\frown}{\cdot}\, X)\langle\sigma,\ \Sigma\rangle$

$= \alpha^{\mathcal{T}}(\{\sigma'\sigma''\pi \mid \langle\sigma',\ \sigma''\rangle \in \tau \wedge \sigma''\pi \in X\})\langle\sigma,\ \Sigma\rangle$

≀definitions of $\vec{\tau}$ and $\overset{\frown}{\cdot}$ in exercise 44.54∫

$= \{\pi \in \{\sigma'\sigma''\pi \mid \langle\sigma',\ \sigma''\rangle \in \tau \wedge \sigma''\pi \in X\} \mid \pi_0 = \sigma\} \subseteq \alpha^{\mathbb{T}}(\{P \in \boldsymbol{S}\llbracket T\rrbracket \mid P_0 = \Sigma\})$

≀definition (44.62) of $\alpha^{\mathcal{T}}$∫

$= \{\sigma'\sigma''\pi \mid \sigma' = \sigma \wedge \langle\sigma',\ \sigma''\rangle \in \tau \wedge \sigma''\pi \in X\} \subseteq \alpha^{\mathbb{T}}(\{P \in \boldsymbol{S}\llbracket T\rrbracket \mid P_0 = \Sigma\})$ ≀definition of $\in$∫

$= \{\sigma\sigma''\pi \mid \langle\sigma,\ \sigma''\rangle \in \tau \wedge \sigma''\pi \in X\} \subseteq \{\pi' \in \mathbb{S}^n \mid n \in \mathbb{N}^+ \wedge \exists P \in \boldsymbol{S}\llbracket T\rrbracket \,.\, P_0 = \Sigma \wedge \forall i \in [0, n[\,.\, \pi'_i \in P_i\}$ ≀definition (44.58) of $\alpha^{\mathbb{T}}(P) \triangleq \bigcup_{n\in\mathbb{N}^+}\{\pi \in \mathbb{S}^n \mid \forall i \in [0, n[\,.\, \pi_i \in P_i\}$∫

$= \bigwedge_{\langle\sigma,\ \sigma''\rangle\in\tau} \forall \sigma''\pi \in X \,.\, \exists P \in \boldsymbol{S}\llbracket T\rrbracket \,.\, P_0 = \Sigma \wedge \sigma \in P_0 \wedge \sigma'' \in P_1 \wedge \forall i \in [0, |\pi|[\,.\, \pi_i \in P_{i+1}$

≀definition of $\subseteq$ where $\pi' = \sigma\sigma''\pi$∫

$= \bigwedge_{\langle\sigma,\ \sigma''\rangle\in\tau} \forall \sigma''\pi \in X \,.\, \exists \Sigma'', P' \,.\, \langle\Sigma,\ \Sigma''\rangle \in T \wedge \Sigma''P' \in \boldsymbol{S}\llbracket T\rrbracket \wedge \sigma \in \Sigma \wedge \sigma'' \in \Sigma'' \wedge \forall i \in [0, |\pi|[\,.\, \pi_i \in P'_i$

≀by definition (44.57) of $\boldsymbol{S}\llbracket T\rrbracket$, $P \in \boldsymbol{S}\llbracket T\rrbracket$ if and only if $\exists \Sigma', \Sigma'', P' \,.\, \langle\Sigma',\ \Sigma''\rangle \in T \wedge \Sigma''P' \in \boldsymbol{S}\llbracket T\rrbracket \wedge P = \Sigma''P'$, so $\Sigma' = \Sigma$ because $P_0 = \Sigma$ and $P_{i+1} = P'_i$∫

$$= \bigwedge_{\langle\sigma,\,\sigma''\rangle\in\tau}\ \bigvee_{\langle\Sigma,\,\Sigma''\rangle\in T} \sigma\in\Sigma\wedge\sigma''\in\Sigma''\wedge\big(X\subseteq\{\sigma''\pi \mid \sigma''\in\Sigma''\wedge\exists P'\,.\,\Sigma''P'\in\boldsymbol{\mathcal{S}}[\![T]\!]\wedge\forall i\in [0,|\pi|[\,.\,\pi_i\in P'_i\} \qquad \wr\text{definitions of }\bigcup\text{ and }\subseteq\int$$

$$= \bigwedge_{\langle\sigma,\,\sigma''\rangle\in\tau}\ \bigvee_{\langle\Sigma,\,\Sigma''\rangle\in T} \sigma\in\Sigma\wedge\sigma''\in\Sigma''\wedge\big(X\subseteq\{\pi' \mid (\pi'_0=\sigma'')\Rightarrow(\pi'_0\in\Sigma''\wedge\exists P\,.\,\pi'_0\in P_0=\Sigma''\wedge P\in\boldsymbol{\mathcal{S}}[\![T]\!]\wedge\forall i\in[0,|\pi'|-1[\,.\,\pi'_i\in P_{i+1})\} \qquad \wr\text{letting }\pi'=\sigma''\pi\text{ and }P=\Sigma''P'\int$$

$$= \bigwedge_{\langle\sigma,\,\sigma''\rangle\in\tau}\ \bigvee_{\langle\Sigma,\,\Sigma''\rangle\in T} \sigma\in\Sigma\wedge\sigma''\in\Sigma''\wedge\big(X\subseteq\{\pi' \mid (\pi'_0=\sigma'')\Rightarrow(\exists P\in\boldsymbol{\mathcal{S}}[\![T]\!]\,.\,P_0=\Sigma''\wedge\forall i\in[0,|\pi'|[\,.\,\pi'_i\in P_i)\} \qquad \wr\text{including }\pi'_0\in P_0\text{ in }\forall i\in[0,|\pi'|-1[\,.\,\pi'_i\in P_{i+1}\int$$

$$= \bigwedge_{\langle\sigma,\,\sigma''\rangle\in\tau}\ \bigvee_{\langle\Sigma,\,\Sigma''\rangle\in T} \sigma\in\Sigma\wedge\sigma''\in\Sigma''\wedge\alpha^{\mathscr{T}}(X)\langle\sigma'',\,\Sigma''\rangle$$

because

$$\alpha^{\mathscr{T}}(X)\langle\sigma'',\,\Sigma''\rangle$$

$$= \{\pi\in X \mid \pi_0=\sigma''\}\subseteq\alpha^{\mathbb{T}}(\{P\in\boldsymbol{\mathcal{S}}[\![T]\!] \mid P_0=\Sigma''\} \qquad \wr(44.62)\int$$

$$= \{\pi\in X \mid \pi_0=\sigma''\}\subseteq\{\pi\in X \mid \pi_0=\sigma''\}\subseteq\bigcup\{\alpha^{\mathbb{T}}(P) \mid P\in\{P\in\boldsymbol{\mathcal{S}}[\![T]\!] \mid P_0=\Sigma''\}\} \qquad \wr\text{definition (44.59) of }\alpha^{\mathbb{T}}\int$$

$$= \{\pi\in X \mid \pi_0=\sigma''\}\subseteq\bigcup\{\bigcup_{n\in\mathbb{N}^+}\{\pi\in\mathbb{S}^n \mid \forall i\in[0,n[\,.\,\pi_i\in P_i\} \mid P\in\{P\in\boldsymbol{\mathcal{S}}[\![T]\!] \mid P_0=\Sigma''\}\} \qquad \wr\text{def (44.58) of }\alpha^{\mathbb{T}}\int$$

$$= \{\pi\in X \mid \pi_0=\sigma''\}\subseteq\{\pi\in\mathbb{S}^* \mid \exists P\in\boldsymbol{\mathcal{S}}[\![T]\!]\,.\,P_0=\Sigma''\wedge\forall i\in[0,|\pi|[\,.\,\pi_i\in P_i\} \qquad \wr\text{definitions of }\in\text{ and }\cup\int$$

$$= X\subseteq\{\pi\in\mathbb{S}^* \mid (\pi_0=\sigma'')\Rightarrow(\exists P\in\boldsymbol{\mathcal{S}}[\![T]\!]\,.\,P_0=\Sigma''\wedge\forall i\in[0,|\pi|[\,.\,\pi_i\in P_i)\} \qquad \wr\text{definition of }\Rightarrow\int$$

(44.65) follows by grouping the two terms of (A) together, renaming, and factorizing the condition $\sigma\in\Sigma$.

— We have $\alpha^{\mathscr{T}}(\langle\sigma,\,\Sigma\rangle\mapsto\varnothing)=\langle\sigma,\,\Sigma\rangle\mapsto\mathsf{ff}$ and commutation, as shown previously, so by the exact fixpoint abstraction theorem 18.23 in a complete lattice, we have

$$\mathsf{mc}$$

$$\triangleq \alpha^{\mathscr{T}}(\boldsymbol{\mathcal{S}}_{\mathrm{t}}[\![\tau]\!]) \qquad \wr\text{definition (44.64) of }\mathsf{mc}\int$$

$$= \alpha^{\mathscr{T}}(\mathsf{lfp}^{\subseteq}\,X\mapsto\mathbb{S}^1\cup\vec{\tau}\mathbin{\frown}X) \qquad \wr\text{exercise 44.54}\int$$

$$= \mathsf{lfp}^{\dot{\Leftarrow}}\,X\mapsto\langle\sigma,\,\Sigma\rangle\mapsto\big((\sigma\in\Sigma)\wedge\bigwedge_{\langle\sigma,\,\sigma'\rangle\in\tau}\ \bigvee_{\langle\Sigma,\,\Sigma'\rangle\in T} X(\sigma',\Sigma')\big) \qquad \wr\text{exercise 44.60 and theorem 18.23}\int$$

$$= \mathsf{gfp}^{\dot{\Rightarrow}}\ X \mapsto \langle\sigma,\ \Sigma\rangle \mapsto \big((\sigma \in \Sigma) \wedge \bigwedge_{\langle\sigma,\,\sigma'\rangle\in\tau} \bigvee_{\langle\Sigma,\,\Sigma'\rangle\in T} X(\sigma',\Sigma')\big) \qquad \wr\text{order-duality}\int$$

□

22 Solutions to Selected Exercises of Chapter 47

Solution to exercise 47.10 $\mathsf{x} \not\rightsquigarrow^{\ell_1} \mathsf{y}$, $\mathsf{x} \not\rightsquigarrow^{\ell_2} \mathsf{y}$, and $\mathsf{x} \rightsquigarrow^{\ell_3} \mathsf{y}$. □

Solution to exercise 47.14 If the initial value x_0 of x at ℓ_0 is positive then the infinite sequence of values of y at ℓ_5 is $1 \cdot 2 \cdot 3 \cdot \ldots$ while it is $1 \cdot 1 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot \ldots$ when the initial value x_0 of x at ℓ_0 is strictly negative. They have a common prefix but differ at position 2 so y depends upon the initial value of x at ℓ_5.

The situation is different at ℓ_4, because in both cases the sequence of values of y is $0 \cdot 1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots$ so y does not depend upon the initial value of x at ℓ_4.

With the iteration condition $\mathsf{i} < 5$, the sequence of values taken by y at ℓ_4 is $0{\cdot}1{\cdot}2{\cdot}3{\cdot}4$ when the initial value x_0 of x at ℓ_0 is positive whereas it is $0 \cdot 1 \cdot 2$ when x_0 is strictly negative. These sequences do not involve differences on values stored in variable y but differences on their lengths linked to the rate of termination. There is a timing channel but not a dependency. □

Solution to exercise 47.31 In S, $\mathsf{x} = \mathsf{y} = 1$ at ℓ_2 so x and y depend on no other variable. For S′ changing the initial value of x e.g. from 2 to 3 will change the value of x and y at ℓ_2 so both depend upon the initial value of x. □

Solution to exercise 47.38 The dependency of the first assignment should be "x and y at ℓ_1 depend on x at ℓ_0 and $\mathsf{x} = \mathsf{y}$." The dependency of the second assignment should be "x and y at ℓ_2 depend on x at ℓ_1 if and only if $\mathsf{x} \neq \mathsf{y}$." The composition is then "x at ℓ_2 depends on x at ℓ_0." More generally, the information provided by the relational semantics $\widehat{\boldsymbol{\mathcal{S}}}^{\vec{\mathrm{R}}}$ of chapter 19 must not be abstracted away. □

Solution to exercise 47.43

Proof of (47.42)

$$\alpha^{\mathrm{d}}(\{\boldsymbol{\mathcal{S}}^{+\infty}[\![\mathsf{S}]\!]\})\ (\ell)$$

$$= \alpha^{\mathrm{d}}(\{\boldsymbol{\mathcal{S}}^{*}[\![\mathsf{S}]\!]\})\ \ell \qquad \wr\text{lemma 47.23}\int$$

$$= \{\langle \mathsf{x}',\ \mathsf{y}\rangle \mid \boldsymbol{\mathcal{S}}^{*}[\![\mathsf{S}]\!] \in \mathcal{D}(\ell)\langle \mathsf{x}',\ \mathsf{y}\rangle\} \qquad \wr\text{definition (47.25) of } \alpha^{\mathrm{d}}\int$$

$$= \{\langle \mathsf{x}',\ \mathsf{y}\rangle \mid \exists\langle\pi_0,\ \pi_1\rangle, \langle\pi'_0,\ \pi'_1\rangle \in \boldsymbol{\mathcal{S}}^{*}[\![\mathsf{S}]\!]\ .\ (\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}'\}\ .\ \boldsymbol{\varrho}(\pi_0)\mathsf{z} = \boldsymbol{\varrho}(\pi'_0)\mathsf{z}) \wedge \mathsf{diff}(\mathsf{seqval}[\![\mathsf{y}]\!](\ell)(\pi_0, \pi_1), \mathsf{seqval}[\![\mathsf{y}]\!](\ell)(\pi'_0, \pi'_1))\} \qquad \wr\text{definition (47.19) of } \mathcal{D}(\ell)\langle \mathsf{x}',\ \mathsf{y}\rangle\int$$

$$= \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \exists\langle\pi_0,\ \pi_1\rangle, \langle\pi'_0,\ \pi'_1\rangle \in \boldsymbol{\mathcal{S}}^{*}[\![\mathsf{S}]\!]\ .\ (\forall \mathsf{z} \in \mathbb{V}\setminus\{\mathsf{x}'\}\ .\ \boldsymbol{\varrho}(\pi_0)\mathsf{z} = \boldsymbol{\varrho}(\pi'_0)\mathsf{z}) \wedge \mathsf{diff}(\text{э}, \text{э})\}$$

≀definition of $\boldsymbol{\mathcal{S}}^{*}[\![\mathtt{S}]\!]$ so that if $\langle\pi,\ \pi'\rangle\ \in\ \boldsymbol{\mathcal{S}}^{*}[\![\mathtt{S}]\!]$ then π ends $\mathsf{at}[\![\mathtt{S}]\!]$ and π' contains only labels of $\mathsf{labx}[\![\mathtt{S}]\!]$ so that, by definition (47.16) of $\mathsf{seqval}[\![\mathtt{y}]\!]$, $\mathsf{seqval}[\![\mathtt{y}]\!](\ell)(\pi_0,\pi_1) = \mathsf{seqval}[\![\mathtt{y}]\!](\ell)(\pi'_0,\pi'_1) = \backepsilon$∫

$= \varnothing$ ≀definition (47.18) of $\mathsf{diff}(\omega,\omega')$ which implies $\omega \neq \backepsilon$ and $\omega' \neq \backepsilon$∫ □

□

Solution to exercise 47.46 We can define $\widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\mathtt{1}]\!] \triangleq \varnothing$, $\widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\mathtt{x}]\!] \triangleq \{\mathtt{x}\}$, and $\widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\mathtt{A}_1 - \mathtt{A}_2]\!] \triangleq \{\mathtt{y} \in \mathbb{vars}[\![\mathtt{A}_1]\!] \cup \mathbb{vars}[\![\mathtt{A}_2]\!] \mid \mathtt{A}_1 \neq \mathtt{A}_2\}$. This handles the case $\widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\mathtt{x} - \mathtt{x}]\!] = \varnothing$ while $\mathbb{vars}[\![\mathtt{x} - \mathtt{x}]\!] = \{\mathtt{x}\}$. Even more precision can be achieved by considering reachable environments only (see remark 47.39). For example, using a constant propagation, an interval, or a zone/octagon analysis, $\mathtt{y} \in \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\mathtt{A}_1 - \mathtt{A}_2]\!]$ only if this analysis cannot prove that $\mathtt{A}_1 - \mathtt{A}_2$ is constant. This can be implemented by a reduced product. □

Solution to exercise 47.61 $\widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\mathtt{Sl}]\!]\ \ell_2 = \{\langle\mathtt{x},\ \mathtt{y}\rangle\} \cup \{\langle\mathtt{z},\ \mathtt{z}\rangle \mid \mathtt{z} \in \mathbb{V} \setminus \{\mathtt{y}\}\}$. This proves that $\mathtt{y}$ at ℓ_2 does not depend on its initial value at ℓ_0 but not that $\mathtt{y}$ at ℓ_2 does not depend on $\mathtt{x}$ at ℓ_0 (which would require to take values of variables into account, for example, by a linear equality analysis of chapter 38). □

Solution to exercise 47.66

$\widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\mathtt{S}_b]\!]\ \ell_0$

$= \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\{\ell_1\ \mathtt{y = z\ ;}\ell_2\ \mathtt{z = x\ ;}\}]\!]\ \ell_0$ ≀definition of $\mathtt{S}_b$∫

$= \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_1\ \mathtt{y = z\ ;}\ell_2\ \mathtt{z = x\ ;}]\!]\ \ell_0$ ≀compound statement (47.57)∫

$= \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_1\ \mathtt{y = z\ ;}]\!]\ \ell_2 \mathbin{⨟} \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_2\ \mathtt{z = x\ ;}]\!]\ \ell_0$ ≀(47.60.b) where $\mathtt{Sl}' = \ell_1\ \mathtt{y = z\ ;}$∫

$= \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ \epsilon\ ]\!]\ \ell_1 \mathbin{⨟} \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_1\ \mathtt{y = z\ ;}]\!]\ \ell_2 \mathbin{⨟} \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_2\ \mathtt{z = x\ ;}]\!]\ \ell_0$ ≀(47.60.b) where $\mathtt{Sl}' = \epsilon\ \ell_1$∫

$= \mathbb{1}_{\mathbb{V}} \mathbin{⨟} \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_1\ \mathtt{y = z\ ;}]\!]\ \ell_2 \mathbin{⨟} \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_2\ \mathtt{z = x\ ;}]\!]\ \ell_0$ ≀(47.54)∫

$= \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_1\ \mathtt{y = z\ ;}]\!]\ \ell_2 \mathbin{⨟} \widehat{\overline{\boldsymbol{\mathcal{S}}}}^{\exists}_{\mathsf{diff}}[\![\ell_2\ \mathtt{z = x\ ;}]\!]\ \ell_0$ ≀$\mathbb{1}_{\mathbb{V}}$ neutral element of ⨟∫

$= \{\langle\mathtt{z},\ \mathtt{y}\rangle, \langle\mathtt{z},\ \mathtt{z}\rangle, \langle\mathtt{x},\ \mathtt{x}\rangle\} \mathbin{⨟} \{\langle\mathtt{x},\ \mathtt{z}\rangle, \langle\mathtt{x},\ \mathtt{x}\rangle, \langle\mathtt{y},\ \mathtt{y}\rangle\}$ ≀(47.44)∫

$= \{\langle\mathtt{x},\ \mathtt{x}\rangle, \langle\mathtt{x},\ \mathtt{z}\rangle, \langle\mathtt{z},\ \mathtt{y}\rangle\}$ ≀definition of ⨟∫

□

Solution to exercise 48.62 — The proof is by structural induction on $\boldsymbol{\tau}$.

- If $\boldsymbol{\tau} = \alpha \in \mathbb{V}_{\mathfrak{t}}$ then $\boldsymbol{\tau}[\beta \in \mathbb{vars}[\![\boldsymbol{\tau}]\!] \leftarrow \vartheta(\beta)] = \alpha[\beta \in \mathbb{vars}[\![\boldsymbol{\tau}]\!] \leftarrow \vartheta(\beta)] = \vartheta(\alpha) = \vartheta(\boldsymbol{\tau})$ (which is α when $\alpha \notin \mathsf{dom}(\vartheta)$ because then $\vartheta(\alpha) = \alpha$);
- Otherwise, $\boldsymbol{\tau} = f(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n)$ so that $\boldsymbol{\tau}[\beta \in \mathbb{vars}[\![\boldsymbol{\tau}]\!] \leftarrow \vartheta(\beta)]$

  $= f(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n)[\beta \in \bigcup_{i=1}^{n} \mathbb{vars}[\![\boldsymbol{\tau}_i]\!] \leftarrow \vartheta(\beta)]$ ≀definition (48.3) of $\mathbb{vars}$∫

  $= f(\boldsymbol{\tau}_1[\beta \in \mathbb{vars}[\![\boldsymbol{\tau}]\!] \leftarrow \vartheta(\beta)], \ldots, \boldsymbol{\tau}_n[\beta \in \mathbb{vars}[\![\boldsymbol{\tau}]\!] \leftarrow \vartheta(\beta)])$ ≀(48.61)∫

$$
\begin{aligned}
&= f(\vartheta(\boldsymbol{\tau}_1), \ldots, \vartheta(\boldsymbol{\tau}_n)) && \wr\text{induction hypothesis}\int \\
&= \vartheta(f(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n)) && \wr\text{def (48.30) of substitution applications}\int
\end{aligned}
$$

□

Solution to exercise 48.60 — The proof is by structural induction on $\boldsymbol{\tau}'$.

- If $\boldsymbol{\tau}' = \alpha \in \mathbb{V}_{\mathfrak{t}}$ then $\{\langle\alpha, \boldsymbol{\tau}\rangle\}(\boldsymbol{\tau}') = \{\langle\alpha, \boldsymbol{\tau}\rangle\}(\alpha) = \boldsymbol{\tau}$ by definition of function application. On the other hand, $\boldsymbol{\tau}[\alpha \leftarrow \boldsymbol{\tau}'] = \boldsymbol{\tau}[\alpha \leftarrow \alpha] = \boldsymbol{\tau}$ by (48.5);

- If $\alpha \neq \boldsymbol{\tau}' = \beta \in \mathbb{V}_{\mathfrak{t}}$ then $\{\langle\alpha, \boldsymbol{\tau}\rangle\}(\boldsymbol{\tau}') = \{\langle\alpha, \boldsymbol{\tau}\rangle\}(\beta) = \beta$ by (48.30) and $\alpha \notin \mathsf{dom}(\{\langle\alpha, \boldsymbol{\tau}\rangle\}) = \{\alpha\}$. This is equal to $\boldsymbol{\tau}'[\alpha \leftarrow \boldsymbol{\tau}] = \beta[\alpha \leftarrow \boldsymbol{\tau}] = \beta$, by (48.5);

- Otherwise, $\boldsymbol{\tau}' = f(\boldsymbol{\tau}'_1, \ldots, \boldsymbol{\tau}'_n)$ so that, by (48.30), induction hypothesis, and (48.5), we have $\{\langle\alpha, \boldsymbol{\tau}\rangle\}(\boldsymbol{\tau}') = \{\langle\alpha, \boldsymbol{\tau}\rangle\}(f(\boldsymbol{\tau}'_1, \ldots, \boldsymbol{\tau}'_n)) = f(\{\langle\alpha, \boldsymbol{\tau}\rangle\}(\boldsymbol{\tau}'_1), \ldots, \{\langle\alpha, \boldsymbol{\tau}\rangle\}(\boldsymbol{\tau}'_n)) = f(\boldsymbol{\tau}'_1[\alpha \leftarrow \boldsymbol{\tau}], \ldots, \boldsymbol{\tau}'_n[\alpha \leftarrow \boldsymbol{\tau}]) = f(\boldsymbol{\tau}'_1, \ldots, \boldsymbol{\tau}'_n)[\alpha \leftarrow \boldsymbol{\tau}] = \boldsymbol{\tau}'[\alpha \leftarrow \boldsymbol{\tau}]$.

□

23 Solutions to Selected Exercises of Chapter 49

Solution to exercise 49.6

```
(* syntax of dynamic types *)

type dtype =
    Dbool
  | Dint
  | Dnil
  | Dpair of dtype * dtype
  | Dlist of dtype
  | Derr

(* equivalent up to Nil for lists *)

let rec equivalent dt1 dt2 =
  match dt1, dt2 with
  | Dlist dt, Dlist dt’ ->
      equivalent dt dt’
  | Dpair (dt1, dt2), Dpair (dt3, dt4) ->
      (equivalent dt1 dt3) && (equivalent dt2 dt4)
  | Dlist dt, Dnil -> true
  | Dnil, Dlist dt -> true
  | _, _ -> dt1 = dt2

(* values *)

type value =
        Vbool of bool
  | Vint of int
  | Vnil
  | Vpair of value * value
  | Vlist of value * value
  | Vderr
  | Vserr

(* dynamic type of values *)

let rec dtypeof v =
  match v with
  | Vbool b -> Dbool
  | Vint i  -> Dint
  | Vnil -> Dnil
  | Vpair (v1,v2) ->
      let dt1 = dtypeof(v1) and dt2 = dtypeof(v2) in
        if (dt1 = Derr) || (dt2 = Derr) then Derr
        else Dpair (dt1, dt2)
  | Vlist (h,t) ->
      (match dtypeof h, dtypeof t with
       | Derr, Derr -> Derr
       | dh, Dnil -> Dlist dh
       | dh, Dlist dt ->
           if (equivalent dh dt) then Dlist dh
           else Derr
       | _, _ -> Derr)
  | Vderr -> Derr
  | Vserr -> Derr

# dtypeof (Vlist (Vnil, Vnil));;
- : dtype = Dlist Dnil
# dtypeof (Vlist (Vpair (Vint 1, Vlist (Vint 1, Vnil)), Vnil));;
- : dtype = Dlist (Dpair (Dint, Dlist Dint))
```

□

Solution to exercise 49.10

```
(* syntax of expressions *)

type program_variable = string
type expression =
    One
  | Var of program_variable
  | Minus of expression * expression
  | Nil
  | Pair of expression * expression
  | Cons of expression * expression
  | Hd of expression
  | Tl of expression
  | Less of expression * expression
  | Isnil of expression
  | Nand of expression * expression

(* environments *)

type environment = (program_variable * value) list

let rec valueof r x =
   match r with
```

```
     [] -> Vserr
   | (y, v) :: t ->
         if (y = x) then v
         else valueof t x

(* evaluation of expressions *)

let rec eval e r =
  match e with
    One ->  Vint 1
  | Var x -> valueof r x
  | Minus (e1, e2) ->
        (match (eval e1 r, eval e2 r) with
         | Vserr, _ -> Vserr
         | _, Vserr -> Vserr
         | _, Vderr -> Vderr
         | Vderr, _ -> Vderr
         | Vint i1, Vint i2 -> Vint (i1 - i2)
         | _, _ -> Vserr)
  | Nil -> Vnil
  | Pair (e1, e2) ->
        (match (eval e1 r, eval e2 r) with
         | Vserr, _ -> Vserr
         | _, Vserr -> Vserr
         | _, Vderr -> Vderr
         | Vderr, _ -> Vderr
         | v1, v2 -> Vpair (v1, v2))
  | Cons (e1, e2) ->
        (match (eval e1 r, eval e2 r) with
         | Vserr, _ -> Vserr
         | _, Vserr -> Vserr
         | _, Vderr -> Vderr
         | Vderr, _ -> Vderr
         | v1, v2 ->
             let l = Vlist (v1, v2) in
               if (dtypeof l) <> Derr then l
               else Vserr)
  | Hd e1 -> let v1 = eval e1 r in
               (match dtypeof v1 with
                | Dlist dh ->
                    (match v1 with
                     | Vnil -> Vderr
                     | Vlist (h,t) -> h
                     | _ -> Vserr)
                | _ -> Vserr)
 | Tl e1 -> let v1 = eval e1 r in
               (match dtypeof v1 with
                | Dlist dh ->
                    (match v1 with
                     | Vnil -> Vderr
                     | Vlist (h,t) -> t
                     | _ -> Vserr)
                | _ -> Vserr)
  | Less (e1, e2) ->
        (match (eval e1 r, eval e2 r) with
         | Vserr, _ -> Vserr
         | _, Vserr -> Vserr
         | _, Vderr -> Vderr
         | Vderr, _ -> Vderr
         | Vint i1, Vint i2 -> Vbool (i1 < i2)
         | _, _ -> Vserr)
  | Isnil e1 ->
        (match (eval e1 r) with
         | Vserr -> Vserr
         | Vderr -> Vderr
         | Vnil  -> (Vbool true)
         | v1  -> (match dtypeof v1 with
                   | Dlist dh -> (Vbool false)
                   | _ -> Vserr))
  | Nand (e1, e2) ->
        (match (eval e1 r, eval e2 r) with
         | Vserr, _ -> Vserr
         | _, Vserr -> Vserr
         | _, Vderr -> Vderr
         | Vderr, _ -> Vderr
         | Vbool i1, Vbool i2 -> Vbool (not (i1 && i2))
         | _, _ -> Vserr)

# eval (Cons ((Pair (One, (Cons (One, Nil)))), Nil)) [];;
- : value = Vlist (Vpair (Vint 1, Vlist (Vint 1, Vnil)), Vnil)
# eval (Cons (Nil, (Var ”x”))) [(”x”, (Vint 1))];;
- : value = Vserr
```

□

Solution to exercise 49.9

$$\mathscr{A}[\![\mathsf{x}]\!]\rho \quad \triangleq \quad \text{let}\, v = \rho(\mathsf{x}) \text{ in } (\!|\, \tau^{\delta}(v) = \boldsymbol{err} \mathrel{?} \Omega^{\delta} \mathrel{:} v \,|\!)$$

This is a dynamic error because initial values or inputs must be checked at runtime. □

Solution to exercise 49.43

```
(* monotypes with variables *)

type type_variable = string
type monotypevar =
  | Tvar of type_variable
  | Tbool
  | Tint
  | Tpair of monotypevar * monotypevar
  | Tlist of monotypevar

(* occurrence of a variable in a type with variables *)

let rec occurrence alpha tv =
  match tv with
  | Tvar beta -> alpha = beta
  | Tbool -> false
  | Tint -> false
  | Tpair (tv1, tv2) -> occurrence alpha tv1 || occurrence alpha tv2
  | Tlist tv1 -> occurrence alpha tv1
```

```
(* Substitutions *)

type substitution = (type_variable * monotypevar) list

let identity : substitution = []

(* application of a substitution to a monotype with variables *)

let rec apply (s:substitution) (tv:monotypevar) =
  match tv with
  | Tvar alpha -> (try List.assoc alpha s
                   with Not_found -> Tvar alpha)
  | Tbool -> tv
  | Tint -> tv
  | Tpair (tv1, tv2) -> Tpair (apply s tv1, apply s tv2)
  | Tlist tv1 -> Tlist (apply s tv1)

(* composition of substitutions *)

let rec domain (s:substitution) =
  match s with
    [] -> []
  | (x, tv) :: tl -> x :: domain tl

let rec union l1 l2 =
  match l1 with
    [] -> l2
  | hd :: tl -> union tl (if List.mem hd l2 then l2 else hd :: l2)

let rec apply_s2_to (s1:substitution) (s2:substitution) : substitution =
   match s1 with
   | [] -> []
   | (a, tv) :: s1’ ->
        if (apply s2 tv) = (Tvar a)
        then apply_s2_to s1’ s2
        else (a, (apply s2 tv)) :: (apply_s2_to s1’ s2)

let rec remove (s2:substitution) d : substitution =
   match s2 with
   | [] -> []
   | (a, tv) :: s2’ ->
        if List.mem a d
        then remove s2’ d
        else (a, tv) :: (remove s2’ d)

let compose (s1:substitution) (s2:substitution) : substitution =
   List.append (apply_s2_to s1 s2)  (remove s2 (domain s1))

(* systems of equations *)

type equations = (monotypevar * monotypevar) list

(* is alpha free in tv? *)

let rec free_in alpha (tv:monotypevar) =
   match tv with
  | Tvar beta -> (alpha = beta)
  | Tbool -> false
  | Tint -> false
  | Tpair (tv1, tv2) -> (free_in alpha tv1) || (free_in alpha tv2)
  | Tlist tv1 -> free_in alpha tv1

(* is alpha in the range of the equations eqns? *)

let rec in_range alpha eqns =
  match eqns with
  | [] -> false
  | (tv, tv’) :: eqns’ -> (free_in alpha tv’) || (in_range alpha eqns’)

(* is tv not a type variable? *)

let not_var tv =
  match tv with
  | (Tvar x) -> false
  | _ -> true

(* apply a substitution to a system of equations *)

let apply_subst_to_eqns (s:substitution) (eqns:equations) : equations =
  List.map (fun (tv1, tv2) -> (apply s tv1, apply s tv2)) eqns

exception NotTypable

(* apply the transformation rule first in eqnsR to equations {eqnsL, eqnsR} *)
(* and report a change if any.                                              *)

let rec apply_rule change (eqnsL:equations) (eqnsR:equations) : bool * equations * equations =
  match eqnsR with
     [] -> (change, eqnsL, [])
  | (tv, tv’) :: eqnsR’ when (tv = tv’) -> (true, eqnsL, eqnsR’)
  | (Tvar alpha, tv) :: eqnsR’ when (occurrence alpha tv) -> raise NotTypable
  | (Tvar alpha, tv) :: eqnsR’ when (in_range alpha eqnsL) || (in_range alpha eqnsR’) ->
    (true, (List.append (apply_subst_to_eqns [(alpha, tv)] eqnsL) [(Tvar alpha, tv)]), (apply_subs
  | (tv, Tvar beta) :: eqnsR’ when (not_var tv) -> (true, eqnsL, ((Tvar beta, tv) :: eqnsR’))
  | (Tbool, Tbool) :: eqnsR’ -> (true, eqnsL, eqnsR’)
  | (Tbool, tv) :: eqnsR’ when (not_var tv) -> raise NotTypable
  | (tv, Tbool) :: eqnsR’ when (not_var tv)  -> raise NotTypable
  | (Tint, Tint) :: eqnsR’ -> (true, eqnsL, eqnsR’)
  | (Tint, tv) :: eqnsR’ when (not_var tv)  -> raise NotTypable
  | (tv, Tint) :: eqnsR’ when (not_var tv)  -> raise NotTypable
  | (Tpair (tv1, tv2), Tpair (tv1’, tv2’)) :: eqnsR’ ->
      (true, eqnsL, ((tv1, tv1’) :: (tv2, tv2’) :: eqnsR’))
  | (Tpair (tv1, tv2), tv) :: eqnsR’ when (not_var tv)  -> raise NotTypable
  | (tv, Tpair (tv1’, tv2’)) :: eqnsR’ when (not_var tv)  -> raise NotTypable
  | (Tlist tv1, Tlist tv1’) :: eqnsR’ ->
      (true, eqnsL, ((tv1, tv1’) :: eqnsR’))
  | (Tlist tv1, tv) :: eqnsR’ when (not_var tv)  -> raise NotTypable
  | (tv, Tlist tv1’) :: eqnsR’ when (not_var tv)  -> raise NotTypable
  | (tv, tv’) :: eqnsR’ -> apply_rule change (List.append eqnsL [(tv, tv’)]) eqnsR’

(* transform solved equations into a substitution *)
```

```
let rec subst_of_eqns (eqns:equations) : substitution =
   match eqns with
   | [] -> []
   | ((Tvar x),tv)::eqns’ -> (x,tv)::(subst_of_eqns eqns’)
   | _ -> failwith ”equations not solved”

(* most general unifier: apply the rule to the equations until no change *)
let rec mgu (eqns:equations) =
  let (change, eqnsL, eqnsR) = (apply_rule false [] eqns) in
    if change then (mgu (List.append eqnsL eqnsR))
    else (subst_of_eqns eqnsL)

# mgu [(Tvar ”a”, Tvar ”b”); (Tvar ”b”, Tvar ”c”); (Tvar ”c”, Tvar ”a”)];;
- : substitution = [(”a”, Tvar ”c”); (”b”, Tvar ”c”)]
# mgu [(Tlist (Tpair (Tint, Tvar ”a”)), (Tlist (Tvar ”a”)))];;
Exception: NotTypable.
```

□

Solution to exercise 49.44

```
(* type environment *)
type type_env = (program_variable * monotypevar) list

(* apply a substitution to a type environment *)
let apply_env (s:substitution) (env:type_env) : type_env =
  List.map (fun (x, tv) -> (x, apply s tv)) env

(* merge environments with different variables *)
let rec merge (env1:type_env) (env2:type_env) : type_env =
   match env1 with
   | [] -> env2
   | (v, tv) :: env1’ ->
       if List.mem_assoc v env2
       then (v, tv) :: (List.remove_assoc v env2)
       else (v, tv) :: (merge env1’ env2)

(* most general unifier of type environments *)
let rec mgu_env (env1:type_env) (env2:type_env) : substitution =
   match env1 with
   | [] -> identity
   | (v, tv) :: env1’ ->
       try let tv’ = List.assoc v env2 in
         let s =  mgu [(tv, tv’)] in
           compose (mgu_env env1’ (List.remove_assoc v env2)) s
       with Not_found -> (mgu_env env1’ env2)

(* fresh variables *)
let next_var = ref 0

let fresh () =
  incr next_var;
  (Tvar (”a” ^ string_of_int !next_var))

let rec infer e =
  match e with
  | One -> ([], Tint)
  | Var x ->  let a = fresh () in ([(x, a)], a)
  | Minus (e1, e2) ->
      let (env1, tv1) = infer e1 and (env2, tv2) = infer e2 in
        let s = (compose (mgu_env env1 env2) (mgu [(tv1,tv2); (tv2,Tint)])) in
          (apply_env s (merge env1 env2), Tint)
  | Nil -> let a = fresh () in ([], Tlist a)
  | Pair (e1, e2) ->
      let (env1, tv1) = infer e1 and (env2, tv2) = infer e2 in
        let a = fresh () and b = fresh () in
       let s = (compose (mgu_env env1 env2) (mgu [((Tpair (tv1,b)),(Tpair (a,tv2)))])) in
            (apply_env s (merge env1 env2), (Tpair (apply s tv1, apply s tv2)))
  | Cons (e1, e2) ->
      let (env1, tv1) = infer e1 and (env2, tv2) = infer e2 in
         let s = (compose (mgu_env env1 env2) (mgu [(Tlist tv1, tv2)])) in
           (apply_env s (merge env1 env2), Tlist (apply s tv1))
  | Hd e1 ->
      let (env1, tv1) = infer e1 in
         let a = fresh () in
            let s = mgu [(tv1,Tlist a)] in
              (apply_env s env1, apply s a)
  | Tl e1 ->
      let (env1, tv1) = infer e1 in
         let a = fresh () in
            let s = mgu [(tv1,Tlist a)] in
              (apply_env s env1, apply s tv1)
  | Less (e1, e2) ->
      let (env1, tv1) = infer e1 and (env2, tv2) = infer e2 in
        let s = (compose (mgu_env env1 env2) (mgu [(tv1,tv2); (tv2,Tint)])) in
          (apply_env s (merge env1 env2), Tbool)
  | Isnil e1 ->
      let (env1, tv1) = infer e1 in
         let a = fresh () in
            let s = mgu [(tv1,Tlist a)] in
              (apply_env s env1, Tbool)
  | Nand  (e1, e2) ->
      let (env1, tv1) = infer e1 and (env2, tv2) = infer e2 in
        let s = (compose (mgu_env env1 env2) (mgu [(tv1,tv2); (tv2,Tbool)])) in
          (apply_env s (merge env1 env2), Tbool)
```

```
# infer (Cons ((Var ”x”),(Var ”y”)));;
- : type_env * monotypevar =
([(”x”, Tvar ”a1”); (”y”, Tlist (Tvar ”a1”))], Tlist (Tvar ”a1”))
# infer (Isnil (Var ”x”));;
- : type_env * monotypevar = ([(”x”, Tlist (Tvar ”a4”))], Tbool)
```

□

24 Solutions to Selected Exercises of Chapter 50

Solution to exercise 50.37 The result of the static analysis

```
while l1: (n > 0) {n:_|_}
   l2: {n:_|_} n = (n - 1);
l3: {n:_|_}
```

states that the invariance specification is unsatisfiable (no execution can reach a program point ℓ in a state satisfying $\mathscr{P}_f(\ell)$). □

Solution to exercise 50.53 We could define $\boldsymbol{\mathcal{S}}^{\tilde{\overleftarrow{\varrho}}}[\![\mathsf{S}]\!] \triangleq \varnothing$ for noncompilable programs. Then $\forall \pi \;.\; \boldsymbol{\mathcal{S}}[\![\mathsf{S}]\!]\,\pi = \varnothing$ implies $\boldsymbol{\mathcal{S}}^{\tilde{\overleftarrow{\varrho}}}[\![\mathsf{S}]\!]\,\mathscr{P}_f = \mathbb{E}\mathbb{v}^{\varrho} \not\subseteq \boldsymbol{\mathcal{S}}^{\overleftarrow{\varrho}}[\![\mathsf{S}]\!]\,\mathscr{P}_f = \varnothing$. However, for the semantics of chapter 6, “Structural Deductive Stateless Prefix Trace Semantics,” and chapter 7, “Maximal Trace Semantics,” we always have $\forall \pi \;.\; \boldsymbol{\mathcal{S}}[\![\mathsf{S}]\!]\,\pi \neq \varnothing$ and so $\widehat{\boldsymbol{\mathcal{S}}}^{\tilde{\overleftarrow{\varrho}}}[\![\mathsf{S}]\!]\,\mathscr{P}_f \subseteq \widehat{\boldsymbol{\mathcal{S}}}^{\overleftarrow{\varrho}}[\![\mathsf{S}]\!]\,\mathscr{P}_f$. □

25 Solutions to Selected Exercises of Chapter 51

Solution to exercise 51.6

```
*** Labeled program:
l1: x = y;
l2:
*** programspec:
l2:{ x:[42,42] }
*** Result of the backward-forward extremal static analysis:
    <{x:T; y:[42, 42]},
     [l1: {x:_|_; y:_|_}; l2: {x:[42, 42]; y:[42, 42]}]>
*** Result of the forward-backward extremal static analysis:
    <{x:T; y:[42, 42]},
     [l1: {x:_|_; y:_|_}; l2: {x:[42, 42]; y:T}]>
```

□

26 Bibliography

[1] Richard Dedekind. *Stetigkeit und irrationale Zahlen.* Braunschweig : F. Vieweg, 1892 (p. 10).

[2] Holbrook M. MacNeille. “Partially Ordered Sets.” *Trans. Amer. Math. Soc.* 42.3 (1937), pp. 416–460 (p. 10).