On lattice theory and program semantics

Jan L.A. van de Snepscheut
Spring 1993

This note is to record some lattice theory. It is a theory concerned with the properties of an uninterpreted partial order \( \leq \), to be read as “at most”, “below”, “implies”, or “is contained in”.

In the first half of the last century, Boole’s formalization of propositional logic led to the concept of boolean algebra. Pierce and Schröder investigated axiomatic definitions of boolean algebras and introduced the notion of a lattice. Dedekind’s work on ideals of algebraic numbers led independently to the same notion.

The general development of lattice theory started in the 1930’s with Birkhoff. His *Lattice Theory* (cf. [2]) is still the standard reference work. A more recent book is [7]. Some results are from [5] and some are from [6].

In a later section we study an operational semantics of a program notation. This semantics is based on [9], and so is the link to the subsequent sections. The axiomatic semantics that follows is largely based on [6]. The refinement ordering is from ([1]); ([13]) studies refinement in a lattice-theoretical setting.

1 Partial orders

Let me start with the definition of a partial order. It is a binary relation \( \leq \) on a set \( Z \) and has three properties:

\[
\forall(x :: x \leq x) \tag{1}
\]

\[
\forall(x, y :: x \leq y \land y \leq x \Rightarrow x = y) \tag{2}
\]

\[
\forall(x, y, z :: x \leq y \land y \leq z \Rightarrow x \leq z) \tag{3}
\]

known as *reflexivity*, *antisymmetry*, and *transitivity* respectively. All bound variables are taken from set \( Z \) but this restriction has been omitted from the formulae. Pair \((Z, \leq)\) is called a *poset*, for partially ordered set. If \( \leq \) is understood, we may refer to \( Z \) as a poset.

Formula \( x \leq y \) is also written as \( y \geq x \). Relation \( \geq \) is called the dual of \( \leq \). If a relation is a partial order, then so is its dual. Both \( \leq \) and \( \geq \) have a higher binding power than has \( = \).
Formula (1) is equivalent to
\[
\forall(x, y :: x \leq y \iff \forall(z :: y \leq z \Rightarrow x \leq z))
\]  

(4)

Proof
(1) $\Rightarrow$ (4)
\[
\forall(z :: y \leq z \Rightarrow x \leq z)
\]
\[
\Rightarrow \{\text{ instantiate } z := y \}\]
\[
y \leq y \Rightarrow x \leq y
\]
\[
= \{ (1) \}
\]
\[
z \leq y
\]

(4) $\Rightarrow$ (1)
\[
x \leq x
\]
\[
\iff \{ (4)[y := x] \}
\]
\[
\forall(z :: x \leq z \Rightarrow x \leq z)
\]
\[
= \text{true}
\]

Formula (3) is equivalent to
\[
\forall(x, y :: x \leq y \Rightarrow \forall(z :: y \leq z \Rightarrow x \leq z))
\]  

(5)

Proof
\[
\forall(x, y, z :: x \leq y \land y \leq z \Rightarrow x \leq z)
\]
\[
= \forall(x, y, z :: x \leq y \Rightarrow (y \leq z \Rightarrow x \leq z))
\]
\[
= \forall(x, y :: x \leq y \Rightarrow \forall(z :: y \leq z \Rightarrow x \leq z))
\]

Combining formulae (4) and (5), we get
\[
\forall(x, y :: x \leq y = \forall(z :: y \leq z \Rightarrow x \leq z))
\]  

(6)
We also have

\[ \forall(x, y :: x \leq y = \forall(z :: z \leq x \Rightarrow z \leq y)) \]  

(7)

Proof

\[
\begin{align*}
x \leq y &= \forall(z :: z \leq x \Rightarrow z \leq y) \\
= & \quad \{ \text{switch to dual} \} \\
y \geq x &= \forall(z :: x \geq z \Rightarrow y \geq z) \\
= & \quad \{ (6) [x, y, \leq ::= y, x, \geq] \text{ since } \geq \text{ is a partial order} \} \\
& \quad \text{true}
\end{align*}
\]

We also have

\[ \forall(x, z :: \forall(y :: (x \leq y) = (z \leq y)) = (x = z)) \]  

(8)

as shown by

\[
\begin{align*}
x &= z \\
\Rightarrow & \quad \forall(y :: (x \leq y) = (z \leq y)) \\
\Rightarrow & \quad \forall(y :: (x \leq y) \Rightarrow (z \leq y)) \land \forall(y :: (z \leq y) \Rightarrow (x \leq y)) \\
= & \quad \{ (6) \} \\
& \quad z \leq x \land x \leq z \\
= & \quad \{ (2): \text{antisymmetry} \} \\
x &= z
\end{align*}
\]

Switching to the dual, we have

\[ \forall(x, y :: \forall(y :: (y \leq x) = (y \leq z)) = (x = z)) \]  

(9)

2 Functions

Function \( f \) from \( Y \) to \( Z \) is called monotonic if

\[ \forall(x, y :: x \leq y \Rightarrow f.x \leq f.y) \]  

(10)
Function application is written with infix operator \( \cdot \) instead of surrounding the argument with parentheses. The ordering relation between \( x \) and \( y \) is taken from \( Y \) whereas the ordering between \( f.x \) and \( f.y \) is taken from \( Z \). They need not be the same orderings. We use the same symbol nevertheless and resolve potential ambiguities by looking at the types of the quantities involved.

Function composition is written with infix operator \( \circ \) and for mapping a function over a set is we use the same infix operator that we use for function application.

\[
\forall (z : z \in Z : (f \circ g).z = f.(g.z))
\]

\[
\forall (Y : Y \subseteq Z : f.Y = \{ y : y \in Y : f.y \})
\]

The two operators \( \cdot \) and \( \circ \) have the highest binding power of all infix operators. Operator \( \circ \) is associative, but \( \cdot \) is not.

**Theorem**

Composition of monotonic functions is monotonic.

**Proof**

For monotonic \( f \) and \( g \), we have

\[
f \circ g \text{ is monotonic}
\]

\[
= \{ (10): \text{definition of monotonicity} \}
\]

\[
\forall (x, y :: x \leq y \Rightarrow (f \circ g).x \leq (f \circ g).y)
\]

\[
= \{ (11): \text{definition of } \circ \}
\]

\[
\forall (x, y :: x \leq y \Rightarrow f.(g.x) \leq f.(g.y))
\]

\[
\Leftarrow \{ f \text{ is monotonic} \}
\]

\[
\forall (x, y :: x \leq y \Rightarrow g.x \leq g.y)
\]

\[
= \{ g \text{ is monotonic} \}
\]

true

**Theorem**

\[
f.(g.Y) = (f \circ g).Y
\]

Next we lift the partial ordering to functions. Let \( Y \) be any set. For functions \( f, g : Y \rightarrow Z \) we define

\[
f \leq g \Leftrightarrow \forall (y : y \in Y : f.y \leq g.y)
\]
Observe that $\leq$ in the right-hand side is $\leq$ on $Z$.

**Theorem** Function composition is monotonic

$f \circ g$ is monotonic in $f$

Proof

$$f \circ g \leq f' \circ g$$

$$= \quad \{ (15) \}$$

$$\forall(x :: (f \circ g).x \leq (f' \circ g).x)$$

$$\Leftarrow \quad \{ \text{switch from } g.x \text{ to } y \}$$

$$\forall(y :: f.y \leq f'.y)$$

$$= \quad \{ (15) \}$$

$$f \leq f'$$

\[\square\]

**Theorem** Function composition is monotonic

$f \circ g$ is monotonic in $g$ if $f$ is monotonic

Proof

$$f \circ g \leq f \circ g'$$

$$= \quad \{ (15) \}$$

$$\forall(x :: (f \circ g).x \leq (f \circ g').x)$$

$$\Leftarrow \quad \{ f \text{ is monotonic} \}$$

$$\forall(x :: g.x \leq g'.x)$$

$$= \quad \{ (15) \}$$

$$g \leq g'$$

\[\square\]

3 Upper and lower bounds

An upper bound of a subset $Y$ of $Z$ is an element $z \in Z$ such that every element in $Y$ is below $z$. The lowest upper bound is an upper bound below every other upper bound. We give a formal definition. Any $x$ for which

$$\forall(z : z \in Z : x \leq z = \forall(y : y \in Y : y \leq z))$$

(18)
is called a lowest upper bound. Similarly, the highest lower bound is defined as any $x$ for which

$$\forall(z : z \in Z : z \leq x = \forall(y : y \in Y : z \leq y))$$

(19)

First, we show that such an equation defines $x$ uniquely. Let both $x$ and $x'$ satisfy (18), then for all $z$

$$\begin{align*}
x &\leq z \\
= & \quad \{ \text{$x$ satisfies (18)} \} \\
\forall(y : y \in Y : y \leq z) \\
= & \quad \{ \text{$x'$ satisfies (18)} \} \\
x' &\leq z
\end{align*}$$

from which, on account of (8), $x = x'$ follows. This allows us to introduce a functional notation. We write $\uparrow Y$ for the lowest upper bound and $\downarrow Y$ for the highest lower bound. These prefix operators are partial functions and have higher binding power than any infix operator. From (18) and (19) we obtain

$$\begin{align*}
\forall(x : x \geq \uparrow Y) &= \forall(y : y \in Y : x \geq y)) \quad (20) \\
\forall(x : x \leq \downarrow Y) &= \forall(y : y \in Y : x \leq y)) \quad (21)
\end{align*}$$

Next, we show why $\uparrow Y$ and $\downarrow Y$ are called the lowest upper bound and highest lower bound of $Y$.

**Theorem**

$$\begin{align*}
\forall(x : (x = \uparrow Y) &= (\forall(y : y \in Y : y \leq x) \land \forall(z : \forall(y : y \in Y : y \leq z) \Rightarrow x \leq z))) \quad (22) \\
\forall(x : (x = \downarrow Y) &= (\forall(y : y \in Y : y \geq x) \land \forall(z : \forall(y : y \in Y : y \geq z) \Rightarrow x \geq z))) \quad (23)
\end{align*}$$

Proof

$$\begin{align*}
x &\equiv \uparrow Y \\
= & \quad \{ \text{definition (18)} \} \\
\forall(z : z \leq x \equiv \forall(y : y \in Y : y \leq z)) \\
= & \quad \{ \text{instantiate with } z := x \} \\
\forall(z : z \leq x \equiv \forall(y : y \in Y : y \leq z)) \land x \leq x \equiv \forall(y : y \in Y : y \leq x) \\
= & \quad \{ (1) : \text{reflexivity} \} \\
\forall(z : z \leq x \equiv \forall(y : y \in Y : y \leq z)) \land \forall(y : y \in Y : y \leq x) \\
= & \quad \{ (3) : y \leq x \land x \leq z \Rightarrow y \leq z \} \\
\forall(z : z \leq x \equiv \forall(y : y \in Y : y \leq z)) \land \forall(y : y \in Y : y \leq x) \\
\end{align*}$$
The first conjunct in (22) expresses that $x$ is an upper bound. The second conjunct expresses that it is the lowest upper bound.

**Theorem**

$$\forall(x :: x \in Y \land \forall(y : y \in Y : y \leq x) \Rightarrow x = \uparrow Y)$$  \hspace{1cm} (24)

$$\forall(x :: x \in Y \land \forall(y : y \in Y : y \geq x) \Rightarrow x = \downarrow Y)$$  \hspace{1cm} (25)

**Proof**

\[
x = \uparrow Y
\]

\[
= \{ \text{(18)} \}
\]

\[
\forall(z : z \in Z : x \leq z = \forall(y : y \in Y : y \leq z))
\]

\[
= \{ \text{instantiate } z := x; \text{ (1): reflexivity} \}
\]

\[
\forall(z : z \in Z : x \leq z = \forall(y : y \in Y : y \leq z)) \land \forall(y : y \in Y : y \leq x)
\]

\[
\quad \Leftarrow \{ x \in Y \land \forall(y : y \in Y : y \leq x) \Rightarrow x \leq z \}
\]

\[
\forall(z : z \in Z : x \leq z \Rightarrow \forall(y : y \in Y : y \leq z)) \land \forall(y : y \in Y : y \leq x) \land x \in Y
\]

\[
= \{ \text{(3): transitivity} \}
\]

\[
\forall(y : y \in Y : y \leq x) \land x \in Y
\]

**Theorem**

$$\forall(x :: \exists(y : y \in Y : x \leq y) \Rightarrow x \leq \uparrow Y)$$  \hspace{1cm} (26)

$$\forall(x :: \exists(y : y \in Y : x \geq y) \Rightarrow x \geq \downarrow Y)$$  \hspace{1cm} (27)

**Proof**

\[
\forall(x :: \exists(y : y \in Y : x \leq y) \Rightarrow x \leq \uparrow Y)
\]

\[
= \{ \text{predicate calculus} \}
\]

\[
\forall(x, y : y \in Y : x \leq y \Rightarrow x \leq \uparrow Y)
\]

\[
\Leftarrow \{ \text{(3): transitivity} \}
\]

\[
\forall(x, y : y \in Y : y \leq \uparrow Y)
\]

\[
= \{ \text{(22)} \}
\]

true
Theorem

$$\forall(x :: \upharpoonright\{y : x \leq y : y\} = x = \downarrow\{y : y \leq x : y\})$$

(28)

It is not necessarily the case that $\uparrow Y$ or $\downarrow Y$ is defined for every set $Y$: it may be that no $x$ satisfies (18) or (19). A poset is called a lattice if both $\uparrow Y$ and $\downarrow Y$ are defined for every finite, nonempty set $Y$. Of course, this is identical to saying that poset $Z$ is a lattice just when $\uparrow\{x, y\}$ and $\downarrow\{x, y\}$ are defined for all $x, y \in Z$. A poset is called a complete lattice if both $\uparrow Y$ and $\downarrow Y$ are defined for every set $Y$.

Two elements of $Z$ have names, provided they exist: top $\top$ and bottom $\bot$. They are defined as

$$\top = \uparrow Z$$

(29)

$$\bot = \downarrow Z$$

(30)

Theorem

$$\downarrow \emptyset$$

(31)

$$\top = \downarrow \emptyset$$

(32)

Proof

$$\downarrow \emptyset = \downarrow Z$$

$$\downarrow \emptyset = \{ (19)[Y := \emptyset, x := \uparrow Z] \}$$

$$\forall(z : z \in Z : z \leq \uparrow Z = \forall(y : y \in \emptyset : z \leq y))$$

$$\downarrow \emptyset = \{ \forall(y : false : p) \}$$

$$\forall(z : z \in Z : z \leq \uparrow Z)$$

$$\downarrow \emptyset = \{ (22)[Y := Z] \}$$

true

Theorem

$$\uparrow\{x\} = x$$

(33)

$$\downarrow\{x\} = x$$

(34)

Proof

For all $y$
\[ \forall \{x\} \leq y \]
\[ = \{ (18) \} \]
\[ \forall (z : z \in \{ x \} : z \leq y) \]
\[ = \]
\[ x \leq y \]

\[ \text{Theorem} \]
\[ \forall (X, y : X \neq \emptyset : y \uparrow X = \uparrow \{ x : x \in X : y \uparrow x \}) \]  \hspace{1cm} (35)
\[ \forall (X, y : X \neq \emptyset : y \downarrow X = \downarrow \{ x : x \in X : y \downarrow x \}) \]  \hspace{1cm} (36)

\[ \text{Theorem} \]
\[ \forall (X : \uparrow \{ x : x \in X : \bot \} = \bot \]  \hspace{1cm} (37)
\[ \forall (X : \downarrow \{ x : x \in X : \top \} = \top \]  \hspace{1cm} (38)

\[ \text{Theorem} \text{ the extreme bounds are monotonic} \]
\[ X \subseteq Y \implies \uparrow X \leq \uparrow Y \]  \hspace{1cm} (39)
\[ X \subseteq Y \implies \downarrow X \geq \downarrow Y \]  \hspace{1cm} (40)

\[ \text{Proof} \]
\[ \downarrow X \geq \downarrow Y \]
\[ = \{ (7) \} \]
\[ \forall (z : z \leq \downarrow X \iff z \leq \downarrow Y) \]
\[ = \{ (21) \} \]
\[ \forall (z : \forall (x : x \in X : z \leq x) \iff \forall (y : y \in Y : z \leq y)) \]
\[ \iff \{ \text{predicate calculus} \} \]
\[ X \subseteq Y \]

\[ \text{Theorem} \text{ the extreme bounds are monotonic} \]
\[ \forall (y : y \in Y : \text{f} \cdot y \leq g \cdot y) \implies \uparrow (f \cdot Y) \leq \uparrow (g \cdot Y) \]  \hspace{1cm} (41)
\[ \forall (y : y \in Y : \text{f} \cdot y \leq g \cdot y) \implies \downarrow (f \cdot Y) \leq \downarrow (g \cdot Y) \]  \hspace{1cm} (42)

\[ \text{Proof} \]
\( \downarrow (f.Y) \leq \downarrow (g.Y) \)

\[
= \quad \{ \text{(7)} \}
\]

\[
\forall (z: \downarrow (f.Y) \leq z) \iff (g.Y \leq z)
\]

\[
= \quad \{ \text{(20)} \}
\]

\[
\forall (z: \forall (y: y \in Y: f.y \leq z) \iff \forall (y: y \in Y: g.y \leq z))
\]

\[
\iff \quad \{ \text{monotonicity of } \forall \}
\]

\[
\forall (z: \forall (y: y \in Y: f.y \leq z) \iff g.y \leq z))
\]

\[
= \quad \{ \text{swap quantifiers} \}
\]

\[
\forall (y: y \in Y: \forall (z: f.y \leq z) \iff g.y \leq z))
\]

\[
= \quad \{ \text{(7)} \}
\]

\[
\forall (y: y \in Y: f.y \leq g.y)
\]

\[\square\]

\textbf{Theorem}

For monotonic \( f \)

\[
\downarrow (f.X) \leq f. \downarrow X
\]

\[\text{(43)}\]

\[
f. \downarrow X \leq \downarrow (f.X)
\]

\[\text{(44)}\]

\textbf{Proof}

\[
f. \downarrow X \leq \downarrow (f.X)
\]

\[
= \quad \{ \text{(21)} [x := f. \downarrow X, Y := (f.X)]; \text{predicate calculus} \}
\]

\[
\forall (x: x \in X: f. \downarrow X \leq f.x)
\]

\[
\iff \quad \{ \text{f is monotonic} \}
\]

\[
\forall (x: x \in X: \downarrow X \leq x)
\]

\[
= \quad \{ \text{(23)} \}
\]

\[\text{true}\]

\[\square\]

\textbf{Theorem}

For monotonic \( f \)

\[
\downarrow X \in X \Rightarrow \downarrow (f.X) = f. \downarrow X
\]

\[\text{(45)}\]

\[
\downarrow X \in X \Rightarrow f. \downarrow X = \downarrow (f.X)
\]

\[\text{(46)}\]

\textbf{Proof}
Theorem
If \( Z \) is a lattice, then

\[
\uparrow\{ Y : Y \in V \uparrow Y \} = \uparrow\cup(Y : Y \in V : Y) \tag{47}
\]

\[
\downarrow\{ Y : Y \in V \downarrow Y \} = \downarrow\cup(Y : Y \in V : Y) \tag{48}
\]

for every finite, nonempty set \( V \) of subsets of \( Z \). If \( Z \) is complete, it holds for all \( V \).

Proof
For all \( z \)

\[
\uparrow\{ Y : Y \in V : \uparrow Y \} \leq z
\]

\[
= \{ (18) \}
\]

\[
\forall(Y : Y \in V : \uparrow Y \leq z)
\]

\[
= \{ (18) \}
\]

\[
\forall(Y : Y \in V : \forall(y : y \in Y : y \leq z))
\]

\[
= \{ (18) \}
\]

\[
\uparrow\{y, Y : y \in Y \land Y \in V : y \leq z \}
\]

\[
= \uparrow\cup(Y : Y \in V : Y) \leq z
\]

\[
\]
If $Y$ and $Z$ are lattices, then their Cartesian product $Y \times Z$ consisting of pairs $(y, z)$ ordered by
\[(y, z) \leq (y', z') = y \leq y' \land z \leq z'\]
is a lattice. If $Y$ and $Z$ are complete, $Y \times Z$ is complete.

4 Lattice algebra

For sets of two elements, we introduce an infix operator for $\uparrow$, also written as $\uparrow$.

\[x \uparrow y = \uparrow\{x, y\}\]

Similarly for $\downarrow$. Both $\uparrow$ and $\downarrow$ have a binding power higher than $\leq$ and $\geq$ but lower than $\cdot$ and $\circ$. They have the following five properties.

\[\forall(x :: x \uparrow x = x \land x \downarrow x = x)\]
\[\forall(x, y :: x \uparrow y = y \uparrow x \land x \downarrow y = y \downarrow x)\]
\[\forall(x, y, z :: (x \uparrow y) \uparrow z = x \uparrow(y \uparrow z) \land (x \downarrow y) \downarrow z = x \downarrow(y \downarrow z))\]
\[\forall(x, y :: x \uparrow(x \downarrow y) = x = x \downarrow(x \uparrow y))\]
\[\forall(x, y :: (x = x \downarrow y) = (x \leq y) = (x \uparrow y = y))\]

known as idempotence, symmetry, associativity, absorption, and consistency respectively.

Proof

Idempotence:

\[x \uparrow x = x\]
\[= \{ \text{definition of } \uparrow \}\]
\[\forall(z :: x \leq z = \forall(y :: y \in \{x, x\} : y \leq z))\]
\[= \forall(z :: x \leq z = x \leq z)\]
\[= \text{true}\]

Symmetry:
\[ h = x \uparrow y \]
\[
= \{ \text{definition of \( \uparrow \)} \}
\]
\[
\forall (z :: h \leq z = \forall (u : u \in \{x, y\} : u \leq z))
\]
and the result follows from the fact that the latter is symmetric in \( x \) and \( y \).

**Associativity:**

For all \( h \), we have

\[
(x \uparrow y) \uparrow z \leq h
\]
\[
= \{ (18) \}
\]
\[
x \uparrow y \leq h \land z \leq h
\]
\[
= \{ (18) \}
\]
\[
x \leq h \land y \leq h \land z \leq h
\]

and the result follows from the fact that the latter is symmetric in \( x \), \( y \), and \( z \).

**Absorption:**

\[
x \mid (x \uparrow y) = x
\]
\[
= \{ (54): \text{consistency} \}
\]
\[
x \leq x \uparrow y
\]
\[
= \{ (22) \}
\]
\[
\text{true}
\]

**Consistency:**

\[
x = x \downarrow y
\]
\[
= \{ \text{definition of \( \downarrow \)} \}
\]
\[
\forall (z :: z \leq x = \forall (u : u \in \{x, y\} : z \leq u))
\]
\[
= \forall (z :: z \leq x = z \leq x \land z \leq y)
\]
\[
= \{ \text{predicate calculus} \}
\]
\[
\forall (z :: z \leq x \Rightarrow z \leq y)
\]
\[
= \{ (7) \}
\]
\[
x \leq y
\]
We started our development with \( \leq \) and derived \( \uparrow \) and \( \downarrow \). We could also have proceeded the other way round. If \( \uparrow \) and \( \downarrow \) are arbitrary infix operators that satisfy (50) through (53), we first show 

\[
(x = x \uparrow y) = (x \downarrow y = y)
\]

\[
z = x \uparrow y
\]

\[
= \quad \{ \text{(53): absorption} \}
\]

\[
x \uparrow (x \downarrow y) = x \uparrow y
\]

\[
\Leftrightarrow
\]

\[
x \downarrow y = y
\]

\[
= \quad \{ \text{(51): symmetry, (53): absorption} \}
\]

\[
y \downarrow x = y \downarrow (x \uparrow y)
\]

\[
\Leftrightarrow
\]

\[
z = x \uparrow y
\]

and then define \( \leq \) by either half of this equality, show that \( \leq \) is a partial order, and finally show that \( x \uparrow y \) and \( x \downarrow y \) are the lowest upper bound and highest lower bound of \( \{x, y\} \).

**Theorem**

Absorption implies idempotence.

**Proof**

\[
z \downarrow x
\]

\[
= \quad \{ \text{(53): } (x = x \uparrow (x \downarrow y)) \}
\]

\[
x \downarrow (x \uparrow (x \downarrow y))
\]

\[
= \quad \{ \text{(53): } (x = x \downarrow (x \uparrow y)) \}[y := x \downarrow x]
\]

\[
z
\]

As a result, a lattice is characterized by the three conditions (51) through (53). A characterization of lattices with only two axioms is due to [8]. The existence of a one-axiom system is shown in [10].

**Theorem**  **Kalman**

The conjunction of conditions

\[
\forall(a, b :: a = (b \uparrow a) \uparrow a)
\]

\[
\forall(a, b, c, d, e, f :: (((a \downarrow b) \downarrow c) \downarrow d) \uparrow e = (((b \downarrow c) \downarrow a) \downarrow e) \uparrow ((f \uparrow d) \downarrow d))
\]
is equivalent to the conjunction of (51) through (53).

**Theorem**

\[ \forall(x, y, x', y' :: x \leq y \land x' \leq y' \Rightarrow x \downarrow x' \leq y \downarrow y' \land x \uparrow x' \leq y \uparrow y') \quad (55) \]

**Proof**

\[
\begin{align*}
  x \leq y \land x' \leq y' \\
  = \quad \{ \text{consistency} \} \\
  x \downarrow y = x \land x' \downarrow y' = x' \\
  \Rightarrow \\
  x \downarrow y \downarrow x' \downarrow y' = x \downarrow x' \\
  = \quad \{ \downarrow \text{ is symmetric, associative; consistency} \} \\
  x \downarrow x' \leq y \downarrow y'
\end{align*}
\]

Here are a few more properties for a lattice in which \( \bot \) and \( \top \) are defined.

\[
\begin{align*}
x \uparrow \top &= \top \\
x \downarrow \bot &= \bot \\
x \uparrow \bot &= x \\
x \downarrow \top &= x
\end{align*}
\]

From (20) and (21) we obtain

\[
\begin{align*}
\forall(x, y, z :: x \geq y \uparrow z = (x \geq y \land x \geq z)) \\
\forall(x, y, z :: x \leq y \downarrow z = (x \leq y \land x \leq z))
\end{align*}
\]

From (22) and (23) we obtain

\[
\forall(x, y :: x \downarrow y \leq x \leq x \uparrow y)
\]

**Theorem**

Both \( \uparrow \) and \( \downarrow \) are monotonic.

**Proof**
\[ x \downarrow y \leq x \downarrow z \]

= \{ (54): consistency \}

\[ x \downarrow y = x \downarrow y \uparrow x \downarrow z \]

= \{ (51): symmetry, (50): idempotence \}

\[ x \uparrow y = x \downarrow y \uparrow z \]

\[ \iff \{ \text{Leibniz} \} \]

\[ y = y \uparrow z \]

= \{ (54): consistency \}

\[ y \leq z \]

Theorem

If \( Z \) is a complete lattice,

the set of functions \( Y \to Z \) forms a complete lattice; \( \quad (64) \)

the set of monotonic functions \( Y \to Z \) forms a complete lattice. \( \quad (65) \)

The lifting of \( \leq \) to functions induces operators \( \uparrow \) and \( \downarrow \) on functions also.

Theorem

If \( W \) is a set of functions from \( Y \) to \( Z \) then

\[ (\uparrow W).y = \uparrow\{f : f \in W : f.y\} \]

\[ (\downarrow W).y = \downarrow\{f : f \in W : f.y\} \] \( \quad (66) \)

\( (67) \)

Proof

\[ \uparrow W \leq g \]

= \{ (18) \}

\[ \forall(f : f \in W : f \leq g) \]

= \{ (15) \}

\[ \forall(f, y : f \in W \land y \in Y : f.y \leq g.y) \]

= \{ (18) \}

\[ \forall(y : y \in Y : \uparrow\{f : f \in W : f.y\} \leq g.y) \]

= \{ (15) \}

\[ \lambda(y : y \in Y : \uparrow\{f : f \in W : f.y\}) \leq g \]
A special case that we will need is the following.

**Theorem**

For all $y \in Y$, we have

\[
(f \uparrow g) \cdot y = f \cdot y \uparrow g \cdot y \tag{68}
\]

\[
(f \downarrow g) \cdot y = f \cdot y \downarrow g \cdot y \tag{69}
\]

Notice that the above properties can also be written in so-called point-free notation.

**Theorem**

If $W$ is a set of functions from $Y$ to $Z$ and $g : X \to Y$ then

\[
(\uparrow W) \circ g = \uparrow \{f : f \in W : f \circ g\} \tag{70}
\]

\[
(\downarrow W) \circ g = \downarrow \{f : f \in W : f \circ g\} \tag{71}
\]

**Theorem**

If $W$ is a set of functions $Y \to Z$ that are $\uparrow$-distributive over $V$ then

\[
\uparrow W \text{ is } \uparrow \text{-distributive over } V . \tag{72}
\]

If $W$ is a set of functions $Y \to Z$ that are $\downarrow$-distributive over $V$ then

\[
\downarrow W \text{ is } \downarrow \text{-distributive over } V . \tag{73}
\]

**Proof**

\[
(\downarrow W) \cdot \downarrow V
\]

\[
= \quad \{ \text{ (67) } \}
\]

\[
\downarrow \{f : f \in W : f \cdot \downarrow V\}
\]

\[
= \quad \{ \text{ $f$ is } \downarrow \text{-distributive over } V \} \}
\]

\[
\downarrow \{f : f \in W : \downarrow \{y : y \in V : f \cdot y\}\}
\]

\[
= \quad \{ \text{ (48) } \}
\]

\[
\downarrow \{y : y \in V : \downarrow \{f : f \in W : f \cdot y\}\}
\]

\[
= \quad \{ \text{ (67) } \}
\]

\[
\downarrow \{y : y \in V : (\downarrow W) \cdot y\}
\]

\[
= \quad \{ \text{ (12) } \}
\]

\[
\downarrow ((\downarrow W) \cdot V)
\]
Theorem

\[ \uparrow (X \times Y) = (\uparrow X, \uparrow Y) \] (74)

\[ \downarrow (X \times Y) = (\downarrow X, \downarrow Y) \] (75)

Proof

\[ \uparrow (X \times Y) \leq (a, b) \]

\[ = \{ (20) \} \]

\[ \forall (x, y : x \in X \land y \in Y : (x, y) \leq (a, b)) \]

\[ = \{ (49) \} \]

\[ \forall (x, y : x \in X \land y \in Y : x \leq a \land y \leq b) \]

\[ = \forall (x : x \in X : x \leq a) \land \forall (y : y \in Y : y \leq b) \]

\[ = \{ (20) \} \]

\[ \uparrow X \leq a \land \uparrow Y \leq b \]

\[ = \{ (49) \} \]

\[ (\uparrow X, \uparrow Y) \leq (a, b) \]

(*** Lifting is insufficiently explored in what follows and, consequently, we have more dummies than needed ***)

5 Distributivity

In any lattice, we have

Theorem

\[ \forall(x, y, z : x \uparrow (y \downarrow z) \leq (x \uparrow y) \downarrow (x \uparrow z)) \] (76)

\[ \forall(x, y, z : x \downarrow (y \uparrow z) \geq (x \downarrow y) \uparrow (x \downarrow z)) \] (77)

Proof
\[ x \downarrow (y \uparrow z) \geq (x \downarrow y) \uparrow (x \downarrow z) \]
\[
= \{ (60) \}
\]
\[ x \downarrow (y \uparrow z) \geq x \downarrow y \quad \land \quad x \downarrow (y \uparrow z) \geq x \downarrow z \]
\[ \Leftarrow \{ \downarrow \text{ is monotonic} \} \]
\[ y \uparrow z \geq y \quad \land \quad y \uparrow z \geq z \]
\[ = \{ (62) \} \]
\[ \text{true} \]

\[ \blacksquare \]

**Theorem**

\[ \forall (x, y, z :: (x \downarrow y) \uparrow (y \downarrow z) \downarrow (z \downarrow x)) \leq (x \downarrow y) \downarrow (y \downarrow z) \downarrow (z \downarrow x)) \]  \hspace{1cm} (78)

**Proof**

We show \( x \downarrow y \leq \text{rhs} \) and then the result follows by (61) and symmetry.

\[ x \downarrow y \leq (x \uparrow y) \downarrow (y \uparrow z) \downarrow (z \downarrow x) \]
\[ \Leftarrow \{ (76) \} \]
\[ x \downarrow y \leq (x \uparrow (y \downarrow z)) \downarrow (y \uparrow z) \]
\[ = \{ (61) \} \]
\[ x \downarrow y \leq x \uparrow (y \downarrow z) \quad \land \quad x \downarrow y \leq y \uparrow z \]
\[ = \{ (62) \} \]
\[ \text{true} \]

\[ \blacksquare \]

**Theorem**

\[ \forall (x, y, z :: (x \downarrow y) \uparrow (x \downarrow z) \leq x \downarrow (y \uparrow (x \downarrow z))) \]  \hspace{1cm} (79)

\[ \forall (x, y, z :: (x \uparrow y) \downarrow (x \uparrow z) \geq x \uparrow (y \downarrow (x \uparrow z))) \]  \hspace{1cm} (80)

**Proof**

\[ (x \downarrow y) \uparrow (x \downarrow z) \]
\[ = \{ (50): \text{idempotence of } \downarrow \} \]
\[ (x \downarrow y) \uparrow (x \downarrow x \downarrow z) \]
\[ \leq \{ (77)[z := x \downarrow z] \} \]
\[ x \downarrow (y \uparrow (x \downarrow z)) \]
Theorem

\( \forall (x, y, z :: x \geq z = x \uparrow (y \downarrow z) \geq (x \downarrow y) \uparrow z) \)  
\( \forall (x, y, z :: x \leq z = x \downarrow (y \uparrow z) \leq (x \uparrow y) \downarrow z) \)

Proof

\[
x \downarrow (y \downarrow z) \leq (x \uparrow y) \downarrow z
= \{ (61) \}
\]
\[
x \downarrow (y \downarrow z) \leq x \downarrow y \land x \uparrow (y \downarrow z) \leq z
= \{ (60) \}
\]
\[
x \leq x \downarrow y \land y \downarrow z \leq x \downarrow y \land x \leq z \land y \downarrow z \leq z
= \{ (62) \}
\]
\[
x \leq z
\]

\[\]

We do not necessarily have that \( \uparrow \) distributes through \( \downarrow \) or the other way round, but the two operators are equally distributive.

Theorem

Properties

(a) \( \forall (x, y, z :: x \downarrow (y \uparrow z) = (x \downarrow y) \uparrow (x \downarrow z)) \)
(b) \( \forall (x, y, z :: x \uparrow (y \downarrow z) = (x \uparrow y) \downarrow (x \uparrow z)) \)
(c) \( \forall (x, y, z :: (x \downarrow y \leq z \land x \leq y \uparrow z) = x \leq z) \)
(d) \( \forall (x, y, z :: (x \uparrow y) \downarrow z \leq x \uparrow (y \downarrow z)) \)
(e) \( \forall (x, y, z :: (x \uparrow y) \downarrow (y \uparrow z) \downarrow (z \uparrow x) = (x \downarrow y) \uparrow (y \downarrow z) \downarrow (z \downarrow x)) \)

are equivalent.

Proof

(a) \( \Rightarrow \) (b):

\[
(x \uparrow y) \downarrow (x \uparrow z)
= \{ (a) [x, y, z := x \uparrow y, x, z] \} 
\]
\[(x \uparrow y) \downarrow z \leq x \uparrow (y \downarrow z)\]

\[= \quad \{ (c); \text{absorption, (51): symmetry } \}
\]
\[x \uparrow (z \downarrow (x \uparrow y))\]

\[= \quad \{ (a)[x, y, z := z, x, y], \text{associativity } \}
\]
\[x \uparrow (z \downarrow x) \uparrow (z \downarrow y)\]

\[= \quad \{ (53): \text{absorption, (51): symmetry } \}
\]
\[z \downarrow (y \downarrow z)\]

(b) \implies (c):

\[x \downarrow y \leq z \land x \leq y \downarrow z\]

\[= \quad \{ (54): \text{consistency } \}
\]
\[(x \downarrow y) \uparrow z = z \land x \uparrow y \downarrow z = y \downarrow z\]

\[= \quad \{ (b) \} \]
\[(x \uparrow z) \downarrow (y \downarrow z) = z \land x \uparrow y \downarrow z = y \downarrow z\]

\[= \quad \{ (53): \text{absorption } \}
\]
\[x \uparrow z = z \land x \uparrow y \downarrow z = y \downarrow z\]

\[= \quad \{ (54): \text{consistency } \}
\]
\[x \leq z\]

(c) \implies (d):

\[(x \uparrow y) \downarrow z \leq x \uparrow (y \downarrow z)\]

\[= \quad \{ (c)[x, z := (x \uparrow y) \downarrow z, (x \uparrow y) \uparrow z] \} \]
\[(x \uparrow y) \downarrow z \leq x \uparrow (y \downarrow z) \land (x \uparrow y) \downarrow z \leq y \uparrow x \uparrow (y \downarrow z)\]

\[= \quad \{ (53): \text{absorption } \}
\]
\[z \downarrow y \leq x \uparrow (y \downarrow z) \land (x \uparrow y) \downarrow z \leq y \uparrow x\]

\[= \quad true \]
(d) $\Rightarrow$ (e):

\[(x \uparrow y) \uparrow (y \uparrow z) \uparrow (z \downarrow x)\]
\[\geq \quad \{ \text{ (d) } \}\]
\[(x \uparrow y) \uparrow (y \uparrow z) \uparrow z \downarrow x\]
\[= \quad \{ (53): \text{ absorption } \}\]
\[(x \uparrow y) \uparrow z \downarrow x\]
\[\geq \quad \{ \text{ (d) } \}\]
\[(x \uparrow (y \uparrow z)) \downarrow x\]
\[= \quad \{ (50): \text{ idempotence } \}\]
\[x \downarrow (y \uparrow z)\]

and hence

\[(x \uparrow y) \uparrow (y \uparrow z) \uparrow (z \downarrow x)\]
\[\geq \quad \{ \text{ above step, (20) } \}\]
\[(x \uparrow (y \uparrow z)) \uparrow (y \uparrow (z \uparrow x)) \uparrow (z \downarrow (x \uparrow y))\]
\[\geq \quad \{ \text{ (d) } \}\]
\[(x \uparrow (y \uparrow z)) \uparrow (y \uparrow (z \uparrow x)) \uparrow z \downarrow (x \uparrow y)\]
\[= \quad \{ (50): \text{ idempotence, (51): symmetry } \}\]
\[(z \uparrow (x \uparrow (y \uparrow z))) \uparrow z \uparrow (y \uparrow (z \uparrow x))) \uparrow (x \uparrow y)\]
\[\geq \quad \{ \text{ (d), monotonicity } \}\]
\[(z \uparrow (x \uparrow (y \uparrow z))) \uparrow (z \uparrow (y \uparrow (z \uparrow x))) \uparrow (x \uparrow y)\]
\[= \quad \{ (51): \text{ symmetry, (50): idempotence } \}\]
\[(x \uparrow y) \uparrow (y \uparrow z) \downarrow (z \uparrow x)\]
\[\geq \quad \{ (78) \}\]
\[x \uparrow y) \uparrow (y \uparrow z) \downarrow (z \downarrow x)\]

(e) $\Rightarrow$ (a): First we show $x \geq z \Rightarrow (x \uparrow y) \uparrow z = x \downarrow (y \uparrow z)$. Assuming $z$, we have

\[\text{true}\]
\[= \quad \{ \text{ (e) } \}\]
\[(x \uparrow y) \downarrow (y \uparrow z) \downarrow (z \uparrow x) = (x \uparrow y) \downarrow (y \uparrow z) \downarrow (z \downarrow x)\]
\[
\begin{align*}
&= \{ x \geq z \} \\
&\quad (x \uparrow y) \downarrow (y \uparrow z) \downarrow x = (x \downarrow y) \uparrow (y \downarrow z) \uparrow z \\
&= \{ (53): \text{absorption}, (51): \text{symmetry} \} \\
&\quad x \downarrow (y \uparrow z) = (x \downarrow y) \uparrow z
\end{align*}
\]

Next, we derive \[
\text{true}
\]
\[
\Rightarrow \quad \{ \text{e} \} \\
&\quad x \downarrow (x \uparrow y) \downarrow (y \uparrow z) \downarrow (z \uparrow x) = x \downarrow ((x \downarrow y) \uparrow (y \downarrow z) \downarrow (z \downarrow x)) \\
\Rightarrow \quad \{ x \geq (x \downarrow y) \uparrow (z \downarrow x) \text{ and use previous result} \} \\
&\quad x \downarrow (x \uparrow y) \downarrow (y \uparrow z) \downarrow (z \uparrow x) = (x \downarrow y \downarrow z) \uparrow (x \downarrow y) \uparrow (z \downarrow x) \\
&= \quad \{ (53): \text{absorption, } (51): \text{symmetry} \} \\
&\quad x \downarrow (y \uparrow z) = (x \downarrow y) \uparrow (z \downarrow x)
\]

\[\square\]

Conditions (79) and (80) with their inequalities replaced by equalities are strictly weaker than the above four conditions. A lattice in which a distribution property holds (and, hence, also the other four) is called a \textit{distributive lattice}.

\textbf{Theorem}

In a distributive lattice
\[
\forall (x, y, z :: (x \uparrow z = y \uparrow z \land x \downarrow z = y \downarrow z) = (x = y)) \tag{84}
\]

Proof
\[
\begin{align*}
x &= y \\
&= \quad \{ \text{(53): absorption} \} \\
&\quad x \uparrow (x \downarrow z) = y \uparrow (y \downarrow z) \\
\Leftarrow &\quad x \uparrow (y \downarrow z) = y \uparrow (x \downarrow z) \land x \downarrow z = y \downarrow z \\
&= \quad \{ \text{distribution} \} \\
&\quad (x \uparrow y) \downarrow (x \uparrow z) = (y \uparrow x) \downarrow (y \downarrow z) \land x \downarrow z = y \downarrow z \\
\Leftarrow &\quad x \uparrow y = y \uparrow x \land x \uparrow z = y \uparrow z \land x \downarrow z = y \downarrow z \\
&= \quad \{ \text{(51): symmetry} \}
\end{align*}
\]
\[ x \uparrow z = y \uparrow z \land x \downarrow z = y \downarrow z \]

\[ \iff \]

\[ x = y \]

We say that a lattice is \emph{universally distributive} if

\[ \forall (x, Y :: x \downarrow Y = \up\{y : y \in Y : x \downarrow y\}\}) \]

\[ \forall (x, Y :: x \uparrow Y = \down\{y : y \in Y : x \uparrow y\}\}) \]

Not every distributive lattice is universally distributive.

\section{Boolean lattices}

For a lattice with \( \top \) and \( \bot \), a complement of element \( x \) is any element \( y \) such that

\[ x \downarrow y = \bot \land x \uparrow y = \top \]

A boolean lattice is a distributive lattice with \( \top \) and \( \bot \) in which every element has a complement. From theorem (84), it follows that complements are unique. We write \( \neg x \) for the complement of \( x \).

\textbf{Theorem}

\[ x \downarrow \neg x = \bot, \quad x \uparrow \neg x = \top \]

\[ x = \neg\neg x \]

\textbf{Proof}

(88) is immediate from the definition of \( \neg \). From theorem (84) it follows that complements are unique and by the symmetry of the definition of \( \neg \) it follows that \( x \) is the complement of \( \neg x \), hence \( x = \neg\neg x \).

\[ \square \]

\textbf{Theorem \textit{De Morgan}}

\[ \neg(x \downarrow y) = \neg x \uparrow \neg y \]

\[ \neg(x \uparrow y) = \neg x \downarrow \neg y \]

\textbf{Proof}

We verify (90) by checking \((x \downarrow y) \downarrow (\neg x \uparrow \neg y) = \bot \) and \((x \uparrow y) \uparrow \neg x \uparrow \neg y = \top \).
\[(x \uparrow y) \uparrow (\neg x \downarrow y)\]  
\[\downarrow \]

\[= \{ \text{distribution} \}\]  
\[= \{ \text{definition } \bot \}\]  
\[= \{ \text{definition } \top \}\]  

\[\bot\]

\[\forall (x, y, z :: (\neg x \uparrow y) \uparrow (x \downarrow z) = (x \downarrow y) \uparrow (\neg x \downarrow z))\]  

\[\forall (x, y :: x \downarrow y \leq z = x \leq \neg y \uparrow z)\]
Proof

 lh$\Rightarrow$ rhs:

\[
x = \{ \text{(59)} \} \\
x \Downarrow \top \\
= \{ \text{(88)} \} \\
x \Downarrow (\neg y \uparrow z \downarrow (\neg y \uparrow z)) \\
= \{ \text{(91)} \} \\
x \Downarrow (\neg y \uparrow z)(y \downarrow \neg z) \\
= \{ \text{distribution } \} \\
(x \Downarrow (\neg y \uparrow z))(x \Downarrow y \downarrow \neg z) \\
= \{ x \downarrow y \leq z \Rightarrow x \downarrow y \neg z = \bot \} \\
(x \Downarrow (\neg y \uparrow z)) \downarrow \bot \\
= \{ \text{(58)} \} \\
x \Downarrow (\neg y \uparrow z)
\]

 lh$\Leftarrow$ rhs:

\[
x \downarrow y \leq z \\
= \{ \text{consistency } \} \\
x \downarrow y \downarrow z = x \downarrow y \\
= \{ x \leq \neg y \uparrow z \Rightarrow (x = x \downarrow (\neg y \uparrow z)) \} \\
x \downarrow y \downarrow z = x \downarrow (\neg y \uparrow z) \downarrow y \\
= \{ \text{(93)} \} \\
\text{true}
\]

\[\square\]

From (95) and (89) we have

**Theorem**  *contrapositive*

\[
\forall(x, y:: x \leq y = \neg y \leq \neg x)
\]  \hspace{1cm} (96)

**Theorem**

\[
\forall(x, y:: x \leq y = (x \downarrow y = \bot) = (\neg x \uparrow y = \top))
\]  \hspace{1cm} (97)
Theorem A complete boolean lattice is universally distributive

In a complete boolean lattice

\[ x \uparrow \downarrow Y = \{ \{ y : y \in Y : x \downarrow y \} \} \quad (98) \]

\[ x \uparrow \downarrow Y = \{ \{ y : y \in Y : x \downarrow y \} \} \quad (99) \]

Proof

We omit range \( y \in Y \).

\[ \uparrow \{ y :: x \downarrow y \} \leq x \uparrow \downarrow Y \]
\[ = \} \{ (20) \} \]
\[ \forall(y :: x \downarrow y \leq x \uparrow \downarrow Y) \]
\[ \Leftarrow \} \{ \text{monotonicity} \} \]
\[ \forall(y :: y \leq \uparrow Y) \]
\[ = \} \{ (20) \} \]
\[ \text{true} \]

Let \( u \) be such that \( \uparrow \{ y :: x \downarrow y \} = u \). Then

\[ y \]
\[ = \]
\[ y \downarrow (x \uparrow \neg x) \]
\[ = \} \{ \text{distribution} \} \]
\[ (y \downarrow x) \downarrow (y \downarrow \neg x) \]
\[ \leq \} \{ (22) \} \]
\[ u \downarrow (y \downarrow \neg x) \]
\[ \leq \} \{ (62) \} \]
\[ u \downarrow \neg x \]

and hence

\[ x \uparrow \downarrow Y \]
\[ \leq \} \{ y \leq u \uparrow \neg x, (42) \} \]
\[ x \uparrow \downarrow \{ y :: u \uparrow \neg x \} \]
\[ = \} \{ \text{check this step; do I need } Y \neq \emptyset ? \} \]
\
\[
\begin{align*}
  x \uparrow (u \uparrow \neg x) \\
  = & \{ \text{(93): complement rule } \} \\
  x \uparrow u \\
  \leq & \{ \text{(62) } \} \\
  u
\end{align*}
\]

from which we conclude \( x \uparrow Y \leq \uparrow \{ y : x \uparrow y \} \).

\[ \square \]

**Theorem**  *generalized De Morgan*

In a complete boolean lattice \( Z \)

\[
\forall (Y : Y \subseteq Z : - \uparrow Y = \uparrow \{ y : y \in Y : -y \}) \tag{100}
\]

\[
\forall (Y : Y \subseteq Z : - \uparrow Y = \uparrow \{ y : y \in Y : -y \}) \tag{101}
\]

7  Distributivity of functions

Function \( f \) is said to be \( \uparrow \)-distributive over \( V \) just when \( \uparrow (f \cdot V) = f \cdot \uparrow V \). If \( f \) is \( \uparrow \)-distributive over all \( V \) then \( f \) is called universally \( \uparrow \)-distributive. If \( f \) is \( \uparrow \)-distributive over all nonempty \( V \) then \( f \) is called positively \( \uparrow \)-distributive. If \( f \) is \( \uparrow \)-distributive over all nonempty, finite \( V \) then \( f \) is called finitely \( \uparrow \)-distributive. A chain is a set \( V \) with the property that all elements can be arranged in a sequence such that each element is below the next element in the sequence. If \( f \) is \( \uparrow \)-distributive over every chain then \( f \) is called universally \( \uparrow \)-continuous. If \( f \) is \( \uparrow \)-distributive over every nonempty chain then \( f \) is called positively \( \uparrow \)-continuous. If \( f \) is \( \uparrow \)-distributive over every nonempty, finite chain then \( f \) is called finitely \( \uparrow \)-continuous. Note that finitely \( \uparrow \)-continuous is equivalent to monotonic:

\[ f \text{ is unanimously } \uparrow \text{-continuous} \]

\[ = \]

\[ f \text{ distributes over all chains of length one or two} \]

\[ = \]

\[
\forall (x, y : x \leq y : f \cdot x \uparrow f \cdot y = f \cdot (x \uparrow y))
\]

\[ = \{ \ x \leq y \ = \ (x \uparrow y = y) \ \} \]

\[
\forall (x, y : x \leq y : f \cdot x \uparrow f \cdot y = f \cdot y)
\]

\[ = \{ \ f \cdot x \leq f \cdot y \ = \ (f \cdot x \uparrow f \cdot y = f \cdot y) \ \} \]

\[
\forall (x, y : x \leq y : f \cdot x \leq f \cdot y)
\]

\[ =\]

\[ f \text{ is monotonic} \]
We have

\[ f \text{ is universally } \uparrow \text{-distributive} \Rightarrow f \text{ is universally } \downarrow \text{-continuous} \]
\[ \downarrow \]
\[ f \text{ is positively } \uparrow \text{-distributive} \Rightarrow f \text{ is positively } \downarrow \text{-continuous} \]
\[ \downarrow \]
\[ f \text{ is finitely } \uparrow \text{-distributive} \Rightarrow f \text{ is finitely } \downarrow \text{-continuous} = f \text{ is monotonic} \]

Similar definitions can be given for \( \downarrow \). Notice that monotonicity does not distinguish between \( \uparrow \)-continuity and \( \downarrow \)-continuity. An interesting theorem, due to J.C.S.P. van der Woude, is the following.

**Theorem**

\[( f \text{ is finitely } \downarrow \text{-distributive } \land f \text{ is positively } \uparrow \text{-continuous} ) \Rightarrow f \text{ is positively } \downarrow \text{-continuous} \]

A generalization of (13) is the following theorem. Let \( X, Y, \) and \( Z \) be posets, \( f : Y \to Z, \) \( g : X \to Y \).

**Theorem**

Function \( f \circ g \) has every distributivity or continuity property shared by \( f \) and \( g \).

(102)

**Theorem**

If \( W \) is a set of functions from \( Y \) to \( Z \) then

\[ \uparrow W \text{ has each type of } \uparrow \text{-distributivity shared by functions in } W \]

(103)

\[ \downarrow W \text{ has each type of } \downarrow \text{-distributivity shared by functions in } W \]

(104)

**Proof**

Let all functions in \( V \) be \( \uparrow \)-distributive over \( V \).

\[
(\uparrow W). \uparrow V \\
= \{ \text{(66)} \}
\]

\[ \uparrow \{ f : f \in W : f. \uparrow V \} \]

\[ = \{ \text{ all functions in } V \text{ are } \uparrow \text{-distributive over } V \} \]

\[ \uparrow \{ f : f \in W : \uparrow \{ v : v \in V : f. v \} \} \]

\[ = \{ \text{(47)} \} \]

\[ \uparrow \{ v : v \in V : \{ f : f \in W : f. v \} \} \]

\[ = \{ \text{(66)} \} \]

\[ \uparrow \{ v : v \in V : (\uparrow W). v \} \]

\[ \square \]

**Theorem**

If \( f \cdot x = g \cdot x \downarrow y \) then \( f \) inherits all \( \uparrow \)-distributivity properties of \( g \)

(105)

If \( f \cdot x = g \cdot x \uparrow y \) then \( f \) inherits all \( \downarrow \)-distributivity properties of \( g \)

(106)
8 Extreme solutions of equations

We write

\[ x : b.x \]  \hspace{1cm} (107)  

for an equation in variable \( x \). The set of solutions of this equation is the set of values \( x \) such that \( b.x \) holds. Consider also

\[ x : b.x \wedge \forall(y : b.y : x \leq y) \]  \hspace{1cm} (108)  

which is a more restrictive equation, and define \( q \) as follows.

\[ q = \bot\{y : b.y : y\} \]  \hspace{1cm} (109)  

Assuming that \( q \) exists, we calculate

\[
\begin{align*}
(108) & \\
& = b.x \wedge \forall(y : b.y : x \leq y) \\
& = \{ (21) \} \\
& = b.x \wedge x \leq \bot\{y : b.y : y\} \\
& = \{ (109) \} \\
& = b.x \wedge x \leq q \\
& = \{ \text{from (109) } b.x \Rightarrow q \leq x \} \\
& = b.x \wedge x = q \\
& = b.q \wedge x = q
\end{align*}
\]

and conclude that (108) has at most one solution, viz. \( q \). If a solution exists, it solves (107) as well and is the lowest solution of (107). We write \([x : b.x]\) for the lowest solution, and \([x : b.x]\) for the highest solution.

**Theorem**

\[ [x : b.x] \text{ exists } = b.q \]  \hspace{1cm} (110)  

**Proof**

\( \text{true} \)
\[
\begin{align*}
\forall (y :: ([x : b] \text{ exists and equals } y) = (b.q \land y = q)) \\
\Rightarrow \quad \{ \text{existential quantification over } y \} \\
[x : b] \text{ exists } = \exists (y :: b.q \land y = q) \\
\Rightarrow \quad [x : b] \text{ exists } = b.q
\end{align*}
\]

\[\Box\]

**Theorem**

\[ [x : b] = \downarrow \{ x : b.x : x \} \tag{111} \]
\[ [x : b] = \downarrow \{ x : b.x : x \} \tag{112} \]
\[ b.[x : b] \land \forall (y :: b.y \Rightarrow [x : b] \leq y) \tag{113} \]
\[ b.[x : b] \land \forall (y :: b.y \Rightarrow y \leq [x : b.x]) \tag{114} \]

We now turn to equations of the form

\[ x : f.x \leq g.x \]

**Theorem**

\[ [x : f.x \leq g.x] \text{ exists if } f \text{ is monotonic and } g \text{ is } \downarrow -\text{distributive over } \{ x : f.x \leq g.x : x \} \tag{115} \]
\[ [x : f.x \leq g.x] \text{ exists if } g \text{ is monotonic and } f \text{ is } \downarrow -\text{distributive over } \{ x : f.x \leq g.x : x \} \tag{116} \]

**Proof**

Because of (110), \([x : f.x \leq g.x] \text{ exists just when } \downarrow \{ y :: f.y \leq g.y : y \} \text{ exists and solves } x : f.x \leq g.x \).

\[ f. \downarrow \{ y :: f.y \leq g.y : y \} \]
\[ \leq \quad \{ \text{ } f \text{ is monotonic: (44) } \} \]
\[ \downarrow \{ y :: f.y \leq g.y : f.y \} \]
\[ \leq \quad \{ \text{ (42) } \} \]
\[ \downarrow \{ y :: f.y \leq g.y : g.y \} \]
\[ \quad \{ g \text{ is } \downarrow -\text{distributive over } \{ x : f.x \leq g.x : x \} \} \]
\[ g. \downarrow \{ y :: f.y \leq g.y : y \} \]
Since the identity function is both \(-\) and \(-\)-distributive over every set, and since \(\uparrow V\) and \(\downarrow V\) exist for every \(V\) in a complete lattice, we have the following result.

**Theorem**

In a complete lattice, \([x : f.x \leq x]\) and \([x : x \leq f.x]\) exist if \(f\) is monotonic. \(\tag{117}\)

This theorem suggests that we also have a look at equation \(x : x = f.x\) and this is what we do next.

## 9 Fixpoints

Let \(f\) be a function on \(Z\). Element \(x\) of \(Z\) is called a fixpoint of \(f\) just when \(f.x = x\). The lowest fixpoint of \(f\) is denoted by \(\mu f\). The highest fixpoint of \(f\) is denoted by \(\nu f\).

\[
\mu f = [x : x = f.x] \tag{118}
\]

\[
\nu f = [x : x = f.x] \tag{119}
\]

From (113) and (114), we have

\[
f.\mu f = \mu f \tag{120}
\]

\[
f.\nu f = \nu f \tag{121}
\]

If \(x \leq f.x\) then \(x\) is called a prefixpoint of \(f\). If \(f.x \leq x\) then \(x\) is called a postfixpoint of \(f\). The following theorem is due to Knaster and Tarski (cf. [12]). It relates extreme prefixpoints, postfixpoints, and fixpoints.

**Theorem**  **Knaster-Tarski**

For monotonic \(f\)

if \(\uparrow\{x : f.x \leq x : x\}\) exists then \(\mu f\) exists and the two are equal \(\tag{122}\)

if \(\uparrow\{x : x \leq f.x : x\}\) exists then \(\nu f\) exists and the two are equal \(\tag{123}\)

**Proof**

Let \(q = \uparrow\{x : x \leq f.x : x\}\).

\[
f.q
= \{ \text{definition of } q \}
= f.\uparrow\{x : x \leq f.x : x\}
\]
\[ \geq \{ (43) \} \]
\[ \uparrow \{ x : x \leq f.x : f.x \} \]
\[ \geq \{ (41) : \uparrow \text{ is monotonic} \} \]
\[ \uparrow \{ x : x \leq f.x : x \} \]
\[ = \{ \text{definition of } q \} \]
\[ q \]
\[ f.q \leq q \]
\[ \Leftarrow \{ (20) \} \]
\[ f.q \in \{ x : x \leq f.x : x \} \]
\[ = \]
\[ f.q \leq f.(f.q) \]
\[ \Leftarrow \{ f \text{ is monotonic} \} \]
\[ q \leq f.q \]

Hence, \( f.q = q \). From (39) we conclude \( q = \{ x : x = f.x \} = \nu f \).

\[ \square \]

Observe that both \( \downarrow \{ x : x \leq f.x : x \} \) and \( \uparrow \{ x : x \geq f.x : x \} \) exist, and hence both \( \mu f \) and \( \nu f \) exist, if the lattice is complete.

The converse to Knaster-Tarski’s theorem is due to [4].

**Theorem** *Davis*

If every monotonic function on a lattice has a fixpoint then the lattice is complete.

The following theorem is part of the folklore. Roland Backhouse dubbed it \( \mu \)-fusion.

**Theorem** *\( \mu \)-fusion*

For monotonic functions \( f \) and \( g \) on a complete lattice,

\[ f.\mu(g \circ f) = \mu(f \circ g) \] (124)

**Proof**

From

\[ f.\mu(g \circ f) \]
\[ = \{ \text{definitions of } \mu \text{ and } \circ \} \]
\[ f.\downarrow \{ x : x = g.(f.x) : x \} \]
\[ \leq \{ (44) \text{ since } f \text{ is monotonic} \} \]
Theorem

Let $f$ be a monotonic function on a complete lattice, and let $x, z$ be such that

$x \leq f.x \leq f.z \leq z$

then

$\exists (y : x \leq y \leq z : f.y = y)$ \hspace{1cm} (125)

10 Functions of two variables

Consider function $f$ of two arguments on a complete lattice. We write $f.(x, y)$ when we want to emphasize that the ordering of $f$'s arguments is as given by (49), and we write $f.x.y$ otherwise. One may view the latter as the curried version of the former. In this section, let $f$ be monotonic. Define $l$ and $h$ by

$l.y = [x : f.x.y \leq x]$ \hspace{1cm} (126)
\[ h.y = [x : x \leq f.x.y] \]  

From (126) we have

\[ f.x.y \leq x \Rightarrow l.y \leq x \]  

and

\[ f.(l.x).x = l.x \]  

**Theorem**

\[ l \text{ and } h \text{ are monotonic functions} \]  

Proof

\[
l.y \leq l.z \\
= \{ (126),(111) \} \\
\leq \{ x : f.x.y \leq x : x \} \leq \{ x : f.x.z \leq x : x \} \\
<= \{ (40) \} \\
\forall(x : f.x.y \leq x \iff f.x.z \leq x) \\
= \{ (6) \} \\
f.x.y \leq f.x.z \\
<= \{ f \text{ is monotonic in its second argument} \} \\
y \leq z
\]

\[ \square \]

**Theorem**

\[ l \text{ inherits all } ^\uparrow -\text{distributivity and } ^\uparrow -\text{continuity of } f \]  

\[ h \text{ inherits all } \downarrow -\text{distributivity and } \downarrow -\text{continuity of } f \]  

Proof

We need to show that \( l \) is \( ^\uparrow \)-distributive over set \( V \) if \( f \) is \( ^\uparrow \)-distributive over a set of pairs of the same type as \( V \). Since \( l \) is monotonic, the type of \( l.V \times V \) is the same as the type of \( V \).

\[
l.\uparrow V = \uparrow(l.V) \\
= \{ (130) \text{ hence (43)} \} 
\]
\[ l.\uparrow V \leq \uparrow(l. V) \]
\[ \iff \{ (128) \} \]
\[ f.\uparrow(l. V).\uparrow V \leq \uparrow(l. V) \]
\[ \iff \{ (129) \} \]
\[ f.\uparrow(l. V).(\uparrow V) \leq \uparrow(f.(l. V).V) \]
\[ \iff \{ f \uparrow \text{-distributes over } l. V \times V \} \]
\[ \text{true} \]

\[ \text{Theorem} \]
\[ l.(x \downarrow y) = l.x \downarrow h.y \quad \text{if } f \text{ is finitely } \downarrow \text{-distributive} \tag{133} \]
\[ h.(x \uparrow y) = h.x \uparrow l.y \quad \text{if } f \text{ is finitely } \uparrow \text{-distributive} \tag{134} \]

\text{Proof}
\text{Ping:}
\[ l.(x \downarrow y) \leq l.x \downarrow h.y \]
\[ \iff \{ (128) \} \]
\[ f.(l.x \downarrow h.y).(x \downarrow y) \leq l.x \downarrow h.y \]
\[ \iff \{ f \text{ is finitely } \downarrow \text{-distributive} \} \]
\[ f.(l.x).x \downarrow f.(h.y).y \leq l.x \downarrow h.y \]
\[ \iff \{ (129) \text{ and its dual} \} \]
\[ \text{true} \]

\text{Pong:}
\[ l.x \downarrow h.y \leq l.(x \downarrow y) \]
\[ \iff \{ (95): \text{shunting} \} \]
\[ l.x \leq \neg h.y \uparrow l.(x \downarrow y) \]
\[ \iff \{ (128) \} \]
\[ f.(-h.y \uparrow l.(x \downarrow y)).x \leq \neg h.y \uparrow l.(x \downarrow y) \]
\[ \iff \{ (95): \text{shunting} \} \]
\[ f.(-h.y \uparrow l.(x \downarrow y)).x \downarrow h.y \leq l.(x \downarrow y) \]
\[ \begin{align*}
= & \quad \{ \text{dual of (129)} \} \\
& f.(\neg h.y \uparrow l.(x \downarrow y)).x \downarrow f.(h.y).y \leq l.(x \downarrow y) \\
= & \quad \{ f \text{ is finitely } \downarrow \text{-distributive} \} \\
& f.((\neg h.y \uparrow l.(x \downarrow y)) \downarrow h.y).(x \downarrow y) \leq l.(x \downarrow y) \\
= & \quad \{ (93) \text{: complement rule} \} \\
& f.(l.(x \downarrow y) \downarrow h.y).(x \downarrow y) \leq l.(x \downarrow y) \\
= & \quad \{ \text{see above: } l.(x \downarrow y) \leq h.y \} \\
& f.(l.(x \downarrow y)).(x \downarrow y) \leq l.(x \downarrow y) \\
= & \quad \{ (129) \} \\
& \text{true}
\end{align*} \]

As a result hereof, we find

**Theorem**

Let the lattice be boolean.

\[ l \text{ is finitely } \downarrow \text{-distributive if } f \text{ is} \quad (135) \]

\[ h \text{ is finitely } \uparrow \text{-distributive if } f \text{ is} \quad (136) \]

Proof

\[ l.(x \downarrow y) \]
\[ = \quad \{ (133) \} \\
& h.(x \downarrow y) \downarrow l.y \\
= & \quad \{ (132) \} \\
& h.x \downarrow h.y \downarrow l.y \\
= & \quad \{ (133) \} \\
& l.x \downarrow l.y \]

**Theorem**

For any function \( f. x. y \) that is monotonic in both arguments, and for chain \( X \),

\[ \begin{align*}
\| \{ x, y : x \in X \land y \in X : f. x. y \} &= \| \{ x : x \in X : f. x. x \} & (137) \\
\| \{ x, y : x \in X \land y \in X : f. x. y \} &= \| \{ x : x \in X : f. x. x \} & (138)
\end{align*} \]

Proof
\[ \uparrow \{ x, y : f \cdot x \cdot y \} \]
\[ = \{ X \text{ is a chain } \} \]
\[ \uparrow \{ x, y : x \leq y \lor y \leq x : f \cdot x \cdot y \} \]
\[ = \{ (47) \} \]
\[ \uparrow \{ x, y : x \leq y : f \cdot x \cdot y \} \uparrow \{ x, x : y \leq x : f \cdot x \cdot y \} \]
\[ = \{ f \text{ is monotonic } \} \]
\[ \uparrow \{ y : f \cdot y \cdot y \} \uparrow \{ x : f \cdot x \cdot x \} \]
\[ = \{ x : f \cdot x \cdot x \} \]

and

\[ \downarrow \{ x, y : f \cdot x \cdot y \} \]
\[ = \{ f \text{ is monotonic } \} \]
\[ f \downarrow X \downarrow X \]
\[ = \{ f \text{ is monotonic } \} \]
\[ \downarrow \{ x : f \cdot x \cdot x \} \]

\[ \Box \]

**Theorem**

If \( f \) and \( g \) are functions of one argument on a universally distributive lattice and they \( \uparrow \)-distribute over chain \( V \) then

\[ f \downarrow g \uparrow \text{-distributes over } V \quad (139) \]

**Proof**

Let \( V_i \) be the lowest-but-\( i \) element of chain \( V \).

\[ (f \downarrow g) \uparrow V \]
\[ = \{ (69) \} \]
\[ (f \uparrow V) \downarrow (g \uparrow V) \]
\[ = \{ f \text{ and } g \uparrow \text{-distribute over } V \} \]
\[ \uparrow \{ i : f \cdot V_i \} \downarrow \{ j : g \cdot V_j \} \]
11 Closures

Function \( f \) is called an \( \uparrow \) -closure if it satisfies the following three conditions.

\[
\begin{align*}
(a) & \quad \forall x :: x \leq f.x \\
(b) & \quad \forall x :: f.x = f.(f.x) \\
(c) & \quad \forall x, y :: x \leq y \Rightarrow f.x \leq f.y
\end{align*}
\]

**Theorem**

\( f \) is an \( \uparrow \) -closure \( \Leftrightarrow \forall x, y :: y \leq f.x = f.y \leq f.x \) \quad (140)

**Proof**

\( \Rightarrow : \)

\[
\begin{align*}
y & \leq f.x \\
\Rightarrow & \quad \{ (c) \} \\
f.y & \leq f.(f.x) \\
= & \quad \{ (b) \} \\
f.y & \leq f.x
\end{align*}
\]
\[
\Rightarrow \{ (a) \} \,\, \,\, \,\, y \leq f . x
\]
\[
\Leftrightarrow : \, (a):
\]

(a):

\[
\forall (x, y :: y \leq f . x = f . y \leq f . x) \Rightarrow \{ \text{ instantiate with } y := x \} \,\,\, \,\,\, \forall (x :: x \leq f . x)
\]

(b):

\[
\forall (x, y :: y \leq f . x = f . y \leq f . x) \Rightarrow \{ \text{ instantiate with } x, y := f . x, f . x \text{ and } x, y := x, f . x \} \,\,\, \,\,\, \forall (x :: f . x \leq f . (f . x)) \land \forall (x :: f . (f . x) \leq f . x)
\]

\[
= \forall (x :: f . x = f . (f . x))
\]

(c):

\[
f . x \leq f . y = \{ (y \leq f . x = f . y \leq f . x)[x := y, y := x] \} \,\,\, \,\,\, z \leq f . y
\]
\[
\Leftrightarrow \{ (a)[x := y] \} \,\,\, \,\,\, z \leq y
\]

\[
\begin{align*}
\text{For monotonic function } f \text{ on a complete lattice, } f \uparrow \downarrow \text{ is the lowest } \uparrow \downarrow \text{-closure function above } f. \\
\text{Stated pointwise, it is a function that, when applied to argument } x, \text{ is the lowest value } y \text{ above } x \text{ and above } f . y; \text{ that is}
\end{align*}
\]
\[
f \downarrow \uparrow . x = \{ y : x \uparrow f . y \leq y \} \tag{141}
\]

Since \( f \) and \( \uparrow \) are monotonic, \( x \uparrow f . y \) is a monotonic function of \( y \). Hence, according to Knaster-Tarski, \( f \downarrow \uparrow . x \) exists and equals \( \{ y : x \uparrow f . y = y \} \). Similarly, the highest \( \downarrow \downarrow \) -closure below \( f \) is

\[
f \downarrow \downarrow . x = \{ y : y \leq x \downarrow f . y \} \tag{142}
\]
From the definition of $f \uparrow$ we conclude

$$\forall(x, y :: x \leq y \land f \downarrow y \leq y) \Rightarrow f \uparrow . x \leq y$$

(143)

and

$$f \circ f \uparrow \leq f \uparrow \quad .$$

(144)

**Theorem** $f \uparrow$ is an $\uparrow$-closure

$$f \uparrow \text{ is an } \uparrow-\text{closure}$$

(145)

**Proof**

We check the three requirements (a) through (c).

(a): From the definition of $f \uparrow$ we conclude $x \leq f \uparrow . x$.

(b):

$$f \uparrow . (f \uparrow . x) = f \uparrow . x$$

$$= \{ (x) \}$$

$$f \uparrow . (f \uparrow . x) \leq f \uparrow . x$$

$$= \{ (x \uparrow f \downarrow y \leq y \Rightarrow f \downarrow . x \leq y)[x := f \downarrow . x, y := f \downarrow . x] \}$$

$$f \downarrow . x \uparrow f . (f \uparrow . x) \leq f \uparrow . x$$

$$= f . (f \uparrow . x) \leq f \uparrow . x$$

$$= \{ (144) \}$$

true

(c): From (130) we conclude that $f \uparrow$ is a monotonic function.

\[ \square \]

**Theorem**

$$\forall(x, y :: y \leq f \downarrow . x = f \downarrow . y \leq f \downarrow . x)$$

(146)

**Proof**

Immediate from (145) and (140).

\[ \square \]

**Theorem** no overshoot

$$\forall(y :: f . y \leq y = \forall(x :: x \leq y = f \downarrow . x \leq y))$$

(147)
Proof
⇒:
Let \( f.y \leq y \).

\[
\begin{align*}
x & \leq y \\
\Rightarrow & \{ f.y \leq y : (143) \} \\
f \leq^* x & \leq y \\
\Rightarrow & \{ x \leq f \leq^* x \} \\
z & \leq y
\end{align*}
\]

⇐:

\[
\begin{align*}
\forall (x :: x \leq y = f \leq^* x \leq y) \\
\Rightarrow & \{ \text{ instantiate with } x := y \} \\
f \leq^* y & \leq y \\
= & \{ (141) \} \\
f \leq^* y = y \\
\Rightarrow & \{ (141) \} \\
y & \leq y \land f.y \leq y \\
= & f.y \leq y
\end{align*}
\]

\[\square\]

**Theorem**

\[
\forall (x :: f.x \leq x = f \leq^* x = x)
\]

(148)

Proof
Immediate from (147) and \( f \leq^* x \geq x \).

\[\square\]

**Theorem**

\[
\forall (x, i :: i \geq 0 : f^i.x \leq f \leq^* x)
\]

(149)

Proof
by induction;
\[i = 0:\]
\[ f^i \cdot x \leq f \downarrow \downarrow \cdot x \]

\[ = \]

\[ x \leq f \downarrow \downarrow \cdot x \]

\[ \leftarrow \quad \{ \text{(141)} \} \]

\[ \text{true} \]

\[ i \geq 0 : \]

\[ f^{i+1} \cdot x \leq f \downarrow \downarrow \cdot x \]

\[ \leftarrow \quad \{ \text{(144)} \} \]

\[ f^{i+1} \cdot x \leq f . (f \downarrow \downarrow \cdot x) \]

\[ \leftarrow \quad \{ \text{f is monotonic} \} \]

\[ f^i \cdot x \leq f \downarrow \downarrow \cdot x \]

**Theorem** \(\downarrow\) is an \(\uparrow\) -closure

\(\downarrow\) is an \(\uparrow\) -closure \hfill (150)

**Proof**

We verify (a) through (c).

(a):

\[ f \leq f \downarrow \downarrow \]

\[ \leftarrow \quad \{ f \circ f \downarrow \downarrow \leq f \downarrow \downarrow \} \]

\[ f \leq f \circ f \downarrow \downarrow \]

\[ = \quad \{ \text{(144)} \} \]

\[ \text{true} \]

(b):

\[ f \downarrow \downarrow \cdot x = f \downarrow \downarrow \cdot x \]

\[ = \quad \{ \text{(149)} [i := 1, f := f \downarrow \downarrow ] \} \]

\[ f \downarrow \downarrow \cdot x \leq f \downarrow \downarrow \cdot x \]

\[ \leftarrow \quad \{ \text{(143)} [f := f \downarrow \downarrow, y := f \downarrow \downarrow \cdot x] \} \]

\[ x \leq f \downarrow \downarrow \cdot x \land f \downarrow \downarrow \cdot (f \downarrow \downarrow \cdot x) \leq f \downarrow \downarrow \cdot x \]

\[ = \quad \{ \text{(145)} \} \]

\[ \text{true} \]
(c):

\[
\begin{align*}
f \downarrow x \leq g \downarrow x \\
\iff \quad \{ \text{(143)} \mid y := g \downarrow x \} \\
z \leq g \downarrow x \land f . (g \downarrow x) \leq g \downarrow x \\
\iff \quad \{ \text{(143)} \} \\
f . (g \downarrow x) \leq g \downarrow x \\
\iff \quad \{ \text{(144)} \} \\
f . (g \downarrow x) \leq g . (g \downarrow x) \\
\iff \\
f \leq g
\end{align*}
\]

\[\blacksquare\]

**Theorem** $\downarrow$-decomposition

For monotonic $f$ and $g$,

\[
(f \uparrow g) \downarrow = f \downarrow c(g \circ f) \downarrow
\]  

(151)

**Proof**

Let $r = f \downarrow c$, $s = t \downarrow$, and $t = g \circ f \downarrow$.

Ping:

\[
\begin{align*}
(f \uparrow g) \downarrow . x \leq r . x \\
\iff \quad \{ \text{(143)} \mid f := f \uparrow g, y := r . x \} \\
z \leq r . x \land (f \uparrow g) . (r . x) \leq r . x \\
\iff \quad \{ \text{(60)} \} \\
z \leq r . x \land f . (r . x) \leq r . x \land g . (r . x) \leq r . x \\
\iff \quad \{ \text{(146)} \mid x := r . x \} \\
z \leq r . x \land f \downarrow . (r . x) = r . x \land g . (r . x) \leq r . x \\
\iff \quad \{ \text{r = f \downarrow c; hence f \downarrow . (r . x) = r . x } \} \\
z \leq f \downarrow . (s . x) \land g . (f \downarrow . (s . x)) \leq f \downarrow . (s . x) \\
\iff \quad \{ \text{y \leq f \downarrow y; use with y := s . x } \} \\
z \leq s . x \land g . (f \downarrow . (s . x)) \leq s . x \\
\iff \quad \{ \text{s = t \downarrow hence x \leq s . x } \}
\end{align*}
\]
\[ g.f\downarrow,(s.x))\leq s.x \]
\[
= \\
(t\circ t\downarrow).x\leq t\downarrow.x \\
= \\
\{\text{ (144) }\} \]

true

Pong:

\[
f\downarrow\circ(g\circ f\downarrow)\downarrow \leq \{ \text{ monotonicity } \} \\
(f\uparrow g)\downarrow.(f\uparrow g)\circ(f\-uparrow g)\downarrow \] \[
= \{ f\downarrow = f\circ f\downarrow \} \]
\[
(f\uparrow g)\downarrow.(f\uparrow g)\downarrow \downarrow \]
\[
= \{ f\downarrow \downarrow = f\downarrow \} \\
(f\uparrow g)\downarrow \]

\[\square\]

Theorem

\[
\mu f = f\downarrow.\bot \\
\nu f = f\downarrow.\top 
\]

\[\text{(152)}\]
\[\text{(153)}\]

Proof

From Knaster-Tarski we know that \(\mu f\) is the unique solution of

\[x : x = f.x \land \forall(y :: f.y \leq y \Rightarrow x \leq y)\] .

\[
\mu f = f\downarrow.\bot \]
\[
= \{ \mu f \text{ solves the equation } \} \\
\mu f = f.(f\downarrow.\bot) \land \forall(y :: f.y \leq y \Rightarrow f\downarrow.\bot \leq y) \]
\[
= \{ f\downarrow.\bot = x \uparrow f.(f\downarrow.\bot) \} \\
\forall(y :: f.y \leq y \Rightarrow f\downarrow.\bot \leq y) \]
\[
= \{ \bot \leq y \} \]
\[ \forall (y :: f.y \leq y \Rightarrow (\bot \leq y = f \downarrow \bot \leq y)) \]
\[ \{ (147) \text{: no overshoot} \} \]
\[ true \]

\[ \square \]

From (149) and (152) we conclude

\[ \mu f \geq \uparrow \{ i : i \geq 0 : f^i.\bot \} \]  \hspace{1cm} (154)

and, by duality

\[ \nu f \leq \downarrow \{ i : i \geq 0 : f^i.\top \} \]  \hspace{1cm} (155)

Let us now see under which condition equality holds.

**Theorem**

\[ \mu f = \uparrow \{ i : i \geq 0 : f^i.\bot \} \quad \text{if } f \text{ is positively } \uparrow \text{-continuous} \]  \hspace{1cm} (156)

\[ \nu f = \downarrow \{ i : i \geq 0 : f^i.\top \} \quad \text{if } f \text{ is positively } \downarrow \text{-continuous} \]  \hspace{1cm} (157)

**Proof**

First, we show that \( \{ i : i \geq 0 : f^i.\bot \} \) is a chain if \( f \) is monotonic, a condition which is implied by \( f \) being continuous. We prove by induction \( \forall (i : i \geq 0 : f^i.\bot \leq f^{i+1}.\bot) \).

\( i = 0 : \)

\[ f^0.\bot \leq f^1.\bot \]
\[ = \]
\[ \bot \leq f^1.\bot \]
\[ = \]
\[ true \]

\( i \geq 0 : \)

\[ f^{i+1}.\bot \leq f^{i+2}.\bot \]
\[ = \]
\[ f.(f^i.\bot) \leq f.(f^{i+1}.\bot) \]
\[ \Leftarrow \]
\[ \{ f \text{ is monotonic} \} \]
\[ f^i.\bot \leq f^{i+1}.\bot \]
\[ = \]
\[ \{ \text{induction hypothesis} \} \]
\[ true \]
From

\[ \uparrow\{i : i \geq 0 : f^i.\perp\} = \{ \text{range split } \} \]
\[ \bot \uparrow\{i : i \geq 1 : f^i.\perp\} = \]
\[ \uparrow\{i : i \geq 0 : f.(f^i.\perp)\} = \{ \{i : i \geq 0 : f^i.\perp\} \text{ is a nonempty chain, and } f \text{ is positively } \uparrow\text{-continuous } \} \]
\[ f.\uparrow\{i : i \geq 0 : f^i.\perp\} \]

we conclude that \( \uparrow\{i : i \geq 0 : f^i.\perp\} \) is a fixpoint of \( f \), and from (154) we conclude that it is below \( \mu f \), the lowest fixpoint. Hence \( \uparrow\{i : i \geq 0 : f^i.\perp\} = \mu f \).

\[ \square \]

12 Galois Connections

Let \( X \) and \( Y \) be partially ordered sets, and let \( f : X \to Y \) and \( g : Y \to X \). Functions \( f \) and \( g \) form a Galois connection just when \( \downarrow \uparrow \text{-Galois}(f, g) \) holds.

\[ \downarrow \uparrow \text{-Galois}(f, g) = \forall(x, y : x \in X \land y \in Y : f. x \leq y = x \leq g.y) \] (158)

We are especially interested in Galois connections for the case where \( X \) and \( Y \) are complete lattices because the functions involved in Galois connections have strong distributivity properties. Throughout this section, \( x \) ranges over \( X \) and \( y \) ranges over \( Y \). We have the following theorem.

**Theorem**

**Properties** (159)

(i) \( \downarrow \uparrow \text{-Galois}(f, g) \)

(ii)a \( f \) is universally \( \uparrow\text{-distributive } \land \forall(y :: g.y = \uparrow\{x : f.x \leq y : x\}) \)

(ii)b \( g \) is universally \( \downarrow\text{-distributive } \land \forall(x :: f.x = \downarrow\{y : x \leq g.y : y\}) \)

(iii)a \( f \) is monotonic \( \land \forall(y :: g.y = [x : f.x \leq y]) \)

(iii)b \( g \) is monotonic \( \land \forall(x :: f.x = [y : x \leq g.y]) \)

(iv) \( f \) and \( g \) are monotonic \( \land \forall(x, y :: x \leq g.f.x) \land f.(g.y) \leq y \),

are equivalent.

**Proof**

(i) \( \Rightarrow \) (ii)a:

First, we calculate
\[
\begin{align*}
f \uparrow V \leq y \\
= & \{ \text{(i); (158)} \} \\
\uparrow V \leq g.y \\
= & \{ \text{(20)} \} \\
\forall (z : z \in V : z \leq g.y) \\
= & \{ \text{(i); (158)} \} \\
\forall (z : z \in V : f.z \leq y) \\
= & \{ \text{(20)} \} \\
\uparrow \{ z : z \in V : f.z \} \leq y \\
= & \\
\uparrow (f. V) \leq y
\end{align*}
\]

from which we conclude (using (8)) the first part of (ii)a. The second part follows from

\[
\begin{align*}
\uparrow \{ x : f.x \leq y : x \} \\
= & \{ \text{(i); (158)} \} \\
\uparrow \{ x : x \leq g.y : x \} \\
= & \{ \text{(28)} \} \\
g.y
\end{align*}
\]

(ii)a \Rightarrow (iii)a:
The monotonicity of \( f \) follows from the distributivity of \( f \). On account of (116), the monotonicity of \( f \) implies that \( \{ x : f.x \leq y \} \) exists, and on account of (112) it equals \( \uparrow \{ x : f.x \leq y : x \} \).

(iii)a \Rightarrow (iv):
Monotonicity of \( g \) follows from

\[
\begin{align*}
g.y \leq g.z \\
= & \{ \text{(iii)} \} \\
[x : f.x \leq y] \leq [x : f.x \leq x] \\
= \\
\uparrow \{ x : f.x \leq y : x \} \leq \uparrow \{ x : f.x \leq z : x \} \\
\Leftarrow & \{ \text{(39)} \} \\
\forall (x : f.x \leq y \Rightarrow f.x \leq z) \\
\Leftarrow \\
y \leq z
\end{align*}
\]
The second part of (iv) follows from

\[ \forall (y::g.y = \{ x : f.x \leq y \}) \]
\[ \Rightarrow \{ (114) \} \]
\[ \forall (y::f.(g.y) \leq y \land \forall (x::f.x \leq y \Rightarrow x \leq g.y)) \]
\[ \Rightarrow \{ \text{ instantiate second term with } y := f.x \} \]
\[ \forall (y::f.(g.y) \leq y) \land \forall (x::f.x \leq f.x \Rightarrow x \leq g.(f.x)) \]
\[ = \]
\[ \forall (y::f.(g.y) \leq y) \land \forall (x::x \leq g.(f.x)) \]

(iv) \Rightarrow (i):

\[ z \leq g.y \]
\[ \Rightarrow \{ f \text{ is monotonic } \} \]
\[ f.x \leq f.(g.y) \]
\[ \Rightarrow \{ f.(g.y) \leq y \} \]
\[ f.x \leq y \]
\[ \Rightarrow \{ g \text{ is monotonic } \} \]
\[ g.(f.x) \leq g.y \]
\[ \Rightarrow \{ x \leq g.(f.x) \} \]
\[ z \leq g.y \]

Conditions (ii)b and (iii)b are dual to (ii)a and (iii)a.

\[ \square \]

We write \( \downarrow \uparrow \) - Galois\( (f,g) \) to indicate the asymmetry in \( f \) and \( g \). For any \( x \), \( f.x \) is the highest lower bound \( \downarrow \) of set \( \{ y : x \leq g.y : y \} \), hence the \( \downarrow \) in the first position. Since \( g \) is a lowest upper bound, we have an \( \uparrow \) in the second position. The original definition of Galois connection (cf. [2]) is

\[ \forall (x,y::f.x \leq y = g.y \leq x) \]

which we would write as \( \downarrow \uparrow \) - Galois\( (f,g) \). It is symmetric in \( f \) and \( g \): both \( f \) and \( g \) are antimonotonic, and both \( f \circ g \) and \( g \circ f \) are contractions. For any set \( V \), \( \uparrow (f.V) = f.\downarrow V \). For any \( x \), \( f.x = \downarrow \{ y : g.x \leq y : y \} \). As an example of such a Galois connection, notice that in a boolean lattice we have

\[ \forall (x,y::\neg x \leq y = \neg y \leq x) \]

and, hence, \( \downarrow \uparrow \) - Galois\( (\neg,\neg) \). Property \( \uparrow (f.V) = f.\downarrow V \) immediately gives us De Morgan’s rules.
The choice between the symmetric and asymmetric Galois connections seems to be no big deal when \( X \) and \( Y \) are different lattices (just switch from \( \leq \) to \( \geq \) in one of them). However, our main interest will be the case where \( X = Y \), and in that case the choice does matter because it is inconvenient to having to work with two partial orders on the same set. In our application, we need the asymmetric Galois connection, and we will omit the arrows from now on.

For a Galois connection \((f, g)\) we also have the following result.

\[
\text{Galois}(f, g) \\
\Rightarrow \{ \text{(iv) above} \}
\]

\[
\forall (x, y :: f.(g.y) \leq y \land x \leq g.(f.x))
\]

\[
\Rightarrow \{ \text{ instantiate with } y := f.x \land x := g.y \}
\]

\[
\forall (x, y :: f.(g.y) \leq y \land x \leq g.(f.x) \land f.(g.(f.x)) \leq f.x \land g.y \leq g.(f.(g.y)))
\]

\[
\Rightarrow \{ \text{(iv): monotonicity of } f \text{ and } g \}
\]

\[
\forall (x, y :: g.(f.(g.y)) \leq g.y \land f.x \leq f.(g.(f.x)) \land f.(g.(f.x)) \leq f.x \land g.y \leq g.(f.(g.y)))
\]

\[
= \forall (x, y :: g.y = g.(f.(g.y)) \land f.(g.(f.x)) = f.x)
\]

\[
\Rightarrow \quad g \circ f = f \circ f \circ g \circ f \quad \land \quad f \circ g = f \circ g \circ f \circ g
\]

Hence \( f \circ g \) is a \( \bot \)-closure and \( g \circ f \) is an \( \top \)-closure. From the universal distributivity properties of \( f \) and \( g \) we conclude

\[
f.\bot = \bot \land g.\top = \top
\]

\[\text{(160)}\]

**Theorem**

If \( \text{Galois}(f, g) \) then conditions

\[
x \leq g.y
\]

\[
f.x \leq f.(g.y)
\]

\[
f.x \leq y
\]

\[
g.(f.x) \leq y
\]

are equivalent.

**Proof**

See proof of (iv) \( \Rightarrow \) (i) above.

**Theorem**

If \( \text{Galois}(f, g) \) and \( X = Y \) then conditions
\[
g \cdot x \leq g \cdot y \\
f \cdot (g \cdot x) \leq f \cdot (g \cdot y) \\
f \cdot (g \cdot x) \leq y
\]
are equivalent.

Proof
Immediate from previous result by instantiating with \( x := g \cdot x \).

\[\Box\]

**Theorem**

If \( \text{Galois}(f, g) \) and \( X = Y \) then conditions

\[
f \cdot x \leq f \cdot y \\
g \cdot (f \cdot x) \leq g \cdot (f \cdot y) \\
x \leq g \cdot (f \cdot y)
\]
are equivalent.

Let \( V \) be a set of pairs of functions where each pair is a Galois connection. If \( F \cdot x = f \{ f, g : (f, g) \in V : f \cdot x \} \) and \( G \cdot y = f \{ f, g : (f, g) \in V : g \cdot y \} \) then \( \text{Galois}(F, G) \).

\[
F \cdot x \leq y \\
= \quad \{ \text{definition of } F \} \\
\uparrow f, g : (f, g) \in V : f \cdot x \leq y \\
= \quad \{ (20) \} \\
\forall f, g : (f, g) \in V : f \cdot x \leq y \\
= \quad \{ \text{Galois}(f, g) \} \\
\forall f, g : (f, g) \in V : x \leq g \cdot y \\
= \quad \{ (21) \} \\
x \leq \downarrow f, g : (f, g) \in V : g \cdot y \\
= \quad \{ \text{definition of } G \} \\
z \leq G \cdot y
\]

**Theorem**

If \( \text{Galois}(f, g) \) then

\[
(g \cdot (f \cdot x) = x) = (x \in g \cdot Y) \quad (161) \\
(f \cdot (g \cdot y) = y) = (y \in f \cdot X) \quad (162)
\]

Proof

\[ g.(f.x) = x \]
\[ \Rightarrow \{ f.x \in Y \} \]
\[ x \in g.Y \]
\[ \Rightarrow \]
\[ \exists (y : y \in Y : x = g.y) \]
\[ = \{ g.y = g.(f.(g.y)) \} \]
\[ \exists (y : y \in Y : x = g.y \land x = g.(f.(g.y))) \]
\[ \Rightarrow \]
\[ x = g.(f.x) \]

\( \square \)

Hence, the set of fixpoints of \( g \circ f \) is \( g.Y \) and the set of fixpoints of \( f \circ g \) is \( f.X \).

**Theorem**

If \( Galois(f,g) \) then

\[ (g.(f.x) \leq x) = (x \in g.Y) \quad (163) \]

\[ (y \leq f.(g.y)) = (y \in f.X) \quad (164) \]

**Theorem** Composition of Galois connections

If \( f0 : Y \rightarrow Z, \ g0 : Z \rightarrow Y, \ f1 : X \rightarrow Y, \ g1 : Y \rightarrow X, \)

\[ Galois(f0,g0) \land Galois(f1,g1) \Rightarrow Galois(f0 \circ f1, g1 \circ g0) \quad (165) \]

Proof

\[ y \leq f0.(f1.x) \]
\[ = \{ Galois(f0,g0) \} \]
\[ g0.y \leq f1.x \]
\[ = \{ Galois(f1,g1) \} \]
\[ g1.(g0.y) \leq x \]

\( \square \)

**Theorem** C.S. Scholten

\[ Galois(f,g) \Rightarrow Galois(f \uparrow, g \downarrow) \quad (166) \]

Proof
\[ x \leq g \downarrow .y \]
\[ \iff \{ x \leq f \downarrow .x \} \]
\[ f \downarrow .x \leq g \downarrow .y \]
\[ \iff \{ z \leq y \mid g.z \Rightarrow z \leq g \downarrow .y \} \]
\[ f \downarrow .x \leq y \mid g.(f \downarrow .x) \]
\[ = \{ (61) \} \]
\[ f \downarrow .x \leq y \land f \downarrow .x \leq g.(f \downarrow .x) \]
\[ = \{ Galois(f, g) \} \]
\[ f \downarrow .x \leq y \land f.(f \downarrow .x) \leq f \downarrow .x \]
\[ = \{ (144) \} \]
\[ f \downarrow .x \leq y \]
\[ \iff \{ \text{similarly} \} \]
\[ x \leq g \downarrow .y \]

\[ \square \]

13 Lifting

**Theorem** \( \mu \) is monotonic

\[ f \leq g \Rightarrow \mu f \leq \mu g \] (167)

Proof

\[
\mu f
= \{ \text{definition of } \mu \} \\
\downarrow \{ x : x = f.x : x \}
= \{ (122) \} \\
\downarrow \{ x : x \geq f.x : x \}
\leq \{ (42) \text{ applies since } x = g.x \Rightarrow x \geq f.x \text{ because } f \leq g \} \\
\downarrow \{ x : x = g.x : x \}
= \{ \text{definition of } \mu \} \\
\mu g
\]
Theorem

\[ \downarrow (f \cdot X) \downarrow (g \cdot X) = \downarrow ((f \downarrow g) \cdot X) \]  
(168)

\[ \downarrow (f \cdot X) \downarrow (g \cdot X) = \downarrow ((f \downarrow g) \cdot X) \]  
(169)

The following theorem is from [3].

**Theorem**  Lifting Galois connections

Let \( f : X \to Y \) and \( g : Y \to X \). Define

\[
\begin{align*}
   f' \cdot a &= f \circ a \circ g \\
   g' \cdot b &= g \circ b \circ f
\end{align*}
\]

where \( a : X \to X \) and \( b : Y \to Y \). If \( a \) and \( b \) are monotonic, we have

\[
Galois(f, g) \Rightarrow Galois(f', g')
\]  
(170)

Proof

\[
\begin{align*}
   a &\leq g' \cdot b \\
   &= \{ \text{definition of } g' \} \\
   a &\leq g \circ b \circ f \\
   &= \{ \text{definition of } \leq \} \\
   \forall (x :: a \cdot x \leq g \cdot (b \cdot (f \cdot x))) \\
   &= \{ \text{Galois}(f, g) \} \\
   \forall (x :: f \cdot (a \cdot x) \leq b \cdot (f \cdot x)) \\
   \Rightarrow & \{ \text{instantiate } x := g \cdot y \} \\
   \forall (y :: f \cdot (a \cdot (g \cdot y)) \leq b \cdot (f \cdot (g \cdot y))) \\
   \Rightarrow & \{ \text{Galois}(f, g) \Rightarrow f \cdot (g \cdot y) \leq y : b \text{ is monotonic} \} \\
   \forall (y :: f \cdot (a \cdot (g \cdot y)) \leq b \cdot y) \\
   &= \{ \text{definition of } f' \text{ and } \leq \} \\
   f' \cdot a &\leq b \\
   \Rightarrow & \{ \text{similarly} \} \\
   a &\leq g' \cdot b
\end{align*}
\]
14 More on fixpoints

Let $f$ be a monotonic function (on a complete lattice), which implies that $\mu f$ exists. Obviously,
\[ \forall (y : y = f.y : y \geq \mu f) . \]

Let $h.x.y$ be a monotonic function of $y$. Define functions $l$, $r$, and $s$ as follows.
\[
\begin{align*}
l.x &= \{ y : y = h.x.y \} \\
r &= \{ f : \forall (x :: f.x = h.x.(f.x)) \} \\
s &= \{ f : \forall (x :: f.x = h.g.x.(f.x)) \}
\end{align*}
\]

Notice that $\lambda (x :: h.x.(f.x))$ is a monotonic function of $f$:
\[
\begin{align*}
\forall (x :: h.x.(f.x) \leq h.x.(g.x)) \\
\leftarrow \{ h \text{ is monotonic in its last argument} \} \\
\forall (x :: f.x \leq g.x) \\
\end{align*}
\]
\[ = f \leq g \]

**Theorem** range of fixpoint is pointwise
\[ l = r \quad . \]

Proof
\[
\begin{align*}
\text{true} \\
= \{ (120) \text{ for } l.x \} \\
\forall (x :: l.x = h.x.(l.x)) \\
\Rightarrow \{ \text{ definition of } r \} \\
r \leq l \\
\text{true} \\
= \{ (120) \text{ for } r \} \\
\forall (x :: r.x = h.x.(r.x)) \\
\Rightarrow \{ \text{ definition of } l.x \} \\
\forall (x :: l.x \leq r.x) \\
= l \leq r
\end{align*}
\]
Theorem

\[ r \circ g = s \quad \text{(172)} \]

Proof

\[ \text{true} \]
\[ = \{ \text{(120) for } r.x \} \]
\[ \forall (x :: r.x = h.x.(r.x)) \]
\[ \Rightarrow \]
\[ \forall (x :: r.(g.x) = h.(g.x).(r.(g.x))) \]
\[ = \]
\[ \forall (x :: (r \circ g).x = h.(g.x).((r \circ g).x)) \]
\[ \Rightarrow \{ \text{(11): definition of } \circ; \text{ definition of } s \} \]
\[ s \leq r \circ g \]

\[ \text{true} \]
\[ = \{ \text{(120) for } s \} \]
\[ \forall (x :: s.x = h.(g.x).(s.x)) \]
\[ \Rightarrow \{ s.x \text{ solves defining equation of } l.(g.x) \} \]
\[ \forall (x :: l.(g.x) \leq s.x) \]
\[ = \{ \text{(171): } l = r \} \]
\[ \forall (x :: r.(g.x) \leq s.x) \]
\[ = \{ \text{(11): definition of } \circ \} \]
\[ r \circ g \leq s \]

\[ \text{Theorem} \quad \text{fixpoint is monotonic} \]

if \( h \) is monotonic in its first argument, \( l \) is monotonic

(173)

Proof

\[ l.x \leq l.y \]
\begin{align*}
&= \{ \text{definition of } l \} \\
&\quad [z : z = h \cdot x \cdot z] \leq [z : z = h \cdot y \cdot z] \\
&= \{ \text{Knaster Tarski } \} \\
&\quad [\{ z : z \geq h \cdot x \cdot z \} : z \geq h \cdot y \cdot z] \\
&\quad \leq \{ (40) \} \\
&\quad \forall (z : z \geq h \cdot x \cdot z \iff z \geq h \cdot y \cdot z) \\
&\quad \leq \{ \text{predicate calculus } \} \\
&\quad \forall (z : h \cdot x \cdot z \leq h \cdot y \cdot z) \\
&\quad \leq \{ h \text{ is monotonic in its first argument } \} \\
&\quad x \leq y
\end{align*}

\section{15 Operational Semantics}

We define the semantics of a program to be the set of all traces (finite or infinite state sequences) that may result from executing the program.

\begin{itemize}
\item $X$ the state space; a cartesian product with one coordinate per program variable
\item the state space is nonempty
\item $T$ the set of all nonempty traces
\item $|t|$ the length of trace $t$
\item $t.i$ the first-but-$i$ element of trace $t$; $0 \leq i < |t|$
\end{itemize}

We write $\text{last}.t$ for the last element of nonempty trace $t$. We write juxtaposition for catenation of strings. We write $x^\infty$ for an infinite sequences of $x$'s. We write $x(v := e \cdot x)$ for the state which is a copy of $x$ except the $v$ coordinate is replaced with the value of $e$ computed in state $x$. Relation $\leq$ is a partial order on traces; $s \leq t$ holds just when $s$ is a prefix of $t$. For $V$ a set of tracess, $\text{fin}.V$ is the subset of $V$ consisting of its finite traces, and $\text{inf}.V$ consists of the infinite traces.

We define the three basic constructs of our programming language. In the definition of $\text{skip}$, observe that we identify states and traces of length one.

\begin{align*}
\text{abort} &= \{ x : x \in X : x^\infty \} \quad (174) \\
\text{skip} &= X \quad (175) \\
v := e &= \{ x : x \in X : x(v := e \cdot x) \} \quad (176)
\end{align*}
We do not bother here with definedness of \( e.x \). Next we define sequential composition for \( U, V \subseteq T \).

\[
U; \ V = \{ u, x, v : ux \in U \land xv \in V : uvr \} \cup \inf.U
\]

(177)

Observe that term \( \inf.U \) can be omitted if \( V \) is nonempty. If \( V \) is empty, we have \( U; \ V = \inf.U \).

When traces \( u \) and \( v \) satisfy \( \last.u = v.0 \), we write \( u; v \) for the trace obtained by catenating \( u \) and the trace obtained by removing the leading element of \( v \).

\[
\begin{align*}
\text{; is associative} \quad & (178) \\
\text{; is universally } \cup -\text{distributive in both arguments} \quad & (179) \\
\text{skip is a left and right unit element of ;} \quad & (180) \\
U; \ V = \{ u : u \in U : \{ u \}; \ V \} \quad & (181)
\end{align*}
\]

For predicate \( b : X \rightarrow \mathit{boolean} \), we define \( b? \) to be the construct that does not modify the state but restricts the state to those states for which \( b \) is \( \text{true} \).

\[
\begin{align*}
b? &= \{ x : x \in X \land b.x : x \} \\
\text{if } & \ll(i : b_i \rightarrow s_i) \gg = \cup(i : b_i)?; s_i) \cup \forall(i : \neg b_i)?; \text{abort} \quad & (183)
\end{align*}
\]

in which we assume that \( ; \) binds more strongly than \( \cup \) does. For \( V \subseteq T \), we define \( V^n \) as

\[
\begin{align*}
V^0 &= \text{skip} \\
V^{n+1} &= V; V^n \quad \text{for } n \geq 0
\end{align*}
\]

The prefix order \( \leq \) on \( T \) is a partial order, but \( T \) is not a complete lattice, yet we need the lowest upper bound of certain sets.

**Theorem**

Every nonempty chain has a lowest upper bound

(184)

**Proof**

Consider chain \( c \) and let \( c_i \) be the lowest-but-\( i \) trace of the chain, that is \( c_i \leq c_{i+1} \). If \( c \) is finite, the highest trace in \( c \) is \( \uparrow c \). If \( c \) is an infinite set, \( c_i \) is strictly rising with \( i \). Let \( d \) be an infinite trace such that \( d.i = c_i.i \) for all \( i \). We have

\[
\begin{align*}
\text{c is a chain} \quad & \Rightarrow \\
\forall(i, j : 0 \leq i < \mid c_j \mid : c_j.i = c_i.i)
\end{align*}
\]
and hence $d$ is an upper bound of $c$. Let $e$ be an infinite trace different from $d$. We have
\[
d \neq e
\]
\[
\Rightarrow \quad \{ \text{both } d \text{ and } e \text{ are infinite} \}
\]
\[
\exists (i :: d.i \neq e.i)
\]
\[
= \quad \exists (i :: c.i \neq e.i)
\]
\[
\Rightarrow \quad \exists (i :: -(c.i \leq e))
\]
which implies that $e$ is not an upper bound of $c$. No finite trace is an upper bound of infinite chain $c$. Hence, $d$ is the one and only upper bound of $c$. In particular $d = \uparrow c$.

Next, we do the "inverse": given an infinite trace $t$, we define a chain $c$ to be a characterizing chain of $t$ just when $t = \uparrow c$.

We say that trace $t$ is a loop trace of trace set $A$ if a characterizing chain $c$ of $t$ exists such that $c.i \in A^i$ for all $i \geq 0$. The set of all loop traces of $A$ is denoted by $\text{loop} \cdot A$.

**Theorem**

If $A \subseteq T$ then

\[\text{loop} \cdot A \cup \text{inf} \cdot A = A; \text{loop} \cdot A\]  
(185)

**Proof**

$\subseteq$:

Obviously, $\text{inf} \cdot A = \text{inf} \cdot A; \text{loop} \cdot A \subseteq A; \text{loop} \cdot A$. Let $t \in \text{loop} \cdot A$ and let $c$ be a characterizing chain of $t$. Since $c_i \leq c_i$ for $i \geq 1$, we can define $d$ as $c_i = c_1; d_{i-1}$ for $i \geq 1$. Since $c$ is a chain, $d$ is a chain. Because $c_1 \in A$ and $c_i; d_i \in A; A'$ we have $d_i \in A'$ for $i \geq 1$ and because $d_0 \in \text{skip}$ we have $\uparrow d \in \text{loop} \cdot A$. Since $t = c_1; \uparrow d$ we have $t \in A; \text{loop} \cdot A$.

$\supseteq$:

Let $a; t \in A; \text{loop} \cdot A$. Either $a \in \text{inf} \cdot A$ and we are done, or else $a \in \text{fin} \cdot A$, $t \in \text{loop} \cdot A$, and $c$ is a characteristic chain of $t$. Define chain $d$ as $d_0 = a.0$ and $d_{i+1} = a; c_i$. Since $c_i \in A^i$ we have $d_{i+1} \in A^{i+1}$ and since $d_0 \in \text{true?}$ we have $a; t = \uparrow d \in \text{loop} \cdot A$. 

$\square$
For $A \subseteq T$ we define $A^\omega$ as
\[ A^\omega = \text{loop}
A \cup \{ n : n \geq 0 : A^n \} \] (186)

Next, we define the semantics of a loop. We use the abbreviation $DO = \text{do } b \rightarrow s \text{ od}$.
\[ DO = (b \Rightarrow s)^\omega; \neg b? \] (187)

This definition of $DO$ is rather complicated. An alternative way of defining $DO$ starts with its first unfolding. A definition of $DO$ is then obtained by solving equation
\[ DO : \quad DO = \text{if } b \rightarrow s; DO|\neg b \rightarrow \text{skip } \text{fi} \]

but it may have multiple solutions. One then introduces a topology, based on a metric, and shows that only one nonempty closed solution exists. Often, this requires that the nondeterminism be bounded to make certain functions continuous. Nondeterminism plays no role in our definition (and hence can be unbounded) and no metric or topology is needed. With our definition, we can prove that $DO$ equals its first unfolding.

**Theorem**

\[ A^\omega = \text{skip } \cup A; A^\omega \] (188)

Proof

\[
A^\omega \\
= \{ (186) \} \\
= \text{loop}
A \cup \{ n : n \geq 0 : A^n \} \\
= \{ (185); \inf A \subseteq \{ n : n \geq 0 : A^n \} \} \\
= \{ \text{definition of } A^{n+1} \} \\
= \text{loop}
A \cup \text{skip } \cup \{ n : n \geq 1 : A^n \} \\
= \{ (179) \} \\
= \text{loop}
A \cup \text{skip } \cup \{ n : n \geq 0 : A^n \} \\
= \{ (179) \} \\
= \text{loop}
A \cup \{ n : n \geq 0 : A^n \} \\
= \{ (186) \} \\
= \text{skip } \cup A; A^\omega
\]
Theorem

\[ \text{DO} = \begin{cases} \text{if } b \rightarrow s; \text{DO}\neg b \rightarrow \text{skip} \end{cases} \]  \hspace{1cm} (189)

Proof

\[ \begin{align*}
\text{if } b \rightarrow s; \text{DO}\neg b \rightarrow \text{skip} \\
= & \quad \{ \text{definition} \} \\
= & \quad b?; s; \text{DO} \cup \neg b?; \text{skip} \cup \text{false}?; \text{abort} \\
= & \quad \{ \text{false}? = \emptyset; (180); \text{skip is right unit of ;} \} \\
= & \quad b?; s; \text{DO} \cup \neg b? \\
= & \quad \{ (187) \} \\
= & \quad b?; s; (b?; s)^\omega; \neg b? \cup \text{skip}; \neg b? \\
= & \quad \{ (180); \text{skip is left unit of ;} \} \\
= & \quad b?; s; (b?; s)^\omega; \neg b? \cup \text{skip}; \neg b? \\
= & \quad \{ (179) \} \\
= & \quad (b?; s)^\omega; \neg b? \\
= & \quad \{ (188) \} \\
= & \quad \text{DO} \\
\end{align*} \]

16 Properties of programs

Theorem

For every program \( S \) and every state \( x \), \( S \) contains a trace starting with \( x \). \hspace{1cm} (190)

Proof

The proof is by induction over the syntax of the program notation. The theorem is obviously true if \( S \) is one of \( \text{abort} \), \( \text{skip} \), or \( \text{assignment} \).

If the theorem holds for \( A \) and \( B \) then we show that it holds for \( A; B \) as well. To prove

\[ \forall(a : a \in A : \exists(t : t \in A; B : a \leq t)) \]
we observe $a \in A: B$ if $a$ is infinite. Also, if $a$ is finite, a trace $y$ exists such that $y \in B$ whose first element $y$ equals $a$’s last element since $a$ is nonempty and $B$ is a program for which the induction hypothesis holds. From the definition of $; a s$ it follows that $a \in A: B$ The result now follows from $a \leq a s$.

The theorem follows from this result plus the induction hypothesis that $A$ contains a trace starting with $x$, for any $x \in X$.

Let $IF = \text{if } b_i \rightarrow s_i, \text{fi}$. Notice that $b?: V$ is the subset of $V$ whose traces start with a state in which $b$ holds. Since for every state $x$ we have $\forall(i :: b, x) \vee \exists(i :: b, x)$, the fact that the theorem holds for $IF$ follows from the induction hypothesis for $s_i$ and the fact that the theorem holds for $\text{abort}$.

Let $DO = \text{do } b \rightarrow s \text{ od}$. Suppose that for some state $x \in X$ we have no trace in $DO$ starting with $x$. Since traces starting with $x$ are present in $(b?: s)^n$ for all $n$, provided the induction hypothesis holds for $s$, but apparently not in $(b?: s)^{n'}$, it follows that all those traces are finite and their last state satisfies $b$. Define a characterizing chain $c$ as $c_0 = x$ and choose $c_{i+1} \in \{c_i\}; s$ arbitrarily. $\uparrow c$ starts with $x$ and is a loop trace of $b?: s$ and hence it is in $DO$.

As a result, we have that programs are nonempty sets of traces.

Next, we look at properties of programs. A property may be viewed as a set of traces, and a program has a property if all its traces are in that set. We write $\text{Prop}$ for the set of all properties $P(T)$. We introduce the weakest precondition for a program and a property to be the condition on the initial state such that all traces with that initial state have the required property.

Function $w : \text{Prog} \times \text{Prop} \times X \rightarrow \text{boolean}$ is defined as

$$w.S.Q.x = \{x\}; S \subseteq Q$$

(191)

This allows us to define the classical weakest precondition and weakest liberal precondition as

$$\text{wlp}.S.Q = w.S.(\text{lt}.Q)$$

(192)

$$\text{wp}.S.Q = w.S.(\text{ct}.Q)$$

(193)

where the liberal and conservative termination functions are defined by

$$\text{lt}.Q = \{t : t \in T \land (|t| = \infty \lor Q.(\text{last}.t) : t)\}$$

(194)

$$\text{ct}.Q = \{t : t \in T \land (|t| < \infty \land Q.(\text{last}.t) : t)\}$$

(195)

Substitution yields

$$\text{wlp}.S.Q.x = \forall(t : t \in \{x\}; S : |t| = \infty \lor Q.(\text{last}.t)$$

(196)

$$\text{wp}.S.Q.x = \forall(t : t \in \{x\}; S : |t| < \infty \land Q.(\text{last}.t)$$

(197)

Both $\text{wlp}$ and $\text{wp}$ are functions that, for fixed command $S$, map a predicate to a predicate. Such a function is sometimes called a predicate transformer. Next, we derive some properties of $\text{wlp}$ and $\text{wp}$.
We use some of the results of lattice theory in this exploration. To that end, observe that the boolean form a complete boolean lattice with

\[
\begin{align*}
\bot &= \text{false} \\
\top &= \text{true} \\
\leq &= \Rightarrow \\
\uparrow &= \forall, \exists \\
\downarrow &= \land, \lor
\end{align*}
\]

**Theorem**

\[\text{wp} \text{ is universally } \land-\text{distributive.}\]

**Proof**

Let \( P \) be a set of predicates.

\[
\text{wp}\cdot S \forall(p : p \in P : \text{p}) \cdot x \\
= \quad \{ \text{196}: \text{definition wp} \} \\
\forall(t : t \in \{ x \}; S : |t|=\infty \lor \forall(p : p \in P : \text{p} \cdot (\text{last} \cdot t)) \\
= \quad \{ \text{67}, \bot = \forall \} \\
\forall(t : t \in \{ x \}; S : |t|=\infty \lor \forall(p : p \in P : \text{p} \cdot (\text{last} \cdot t)) \\
= \quad \{ \text{predicate calculus} \} \\
\forall(t : t \in \{ x \}; S : \forall(p : p \in P : |t|=\infty \lor p \cdot (\text{last} \cdot t)) \\
= \quad \{ \text{predicate calculus} \} \\
\forall(p : p \in P ; \forall(t : t \in \{ x \}; S : |t|=\infty \lor p \cdot (\text{last} \cdot t)) \\
= \quad \{ \text{196}: \text{definition wp} \} \\
\forall(p : p \in P ; \text{wp}\cdot S \cdot p \cdot x) \\
= \quad \{ \text{67}, \bot = \forall \} \\
\forall(p : p \in P ; \text{wp}\cdot S \cdot p \cdot x)
\]

\( \square \)

**Theorem**

For all \( S \) and \( Q \)

\[\text{wp}\cdot S \cdot Q = \text{wp}\cdot S \cdot Q \land \text{wp}\cdot S \cdot \text{true}\]

**Proof**

\[(\text{wp}\cdot S \cdot Q \land \text{wp}\cdot S \cdot \text{true} \cdot x)\]
\[
\begin{align*}
= & \{ (67), \ l = \wedge \} \\
wp.S.Q.x & \wedge wp.S.true.x \\
= & \{ (196) \text{ and } (197); \text{ definition of } wp \text{ and } wp \} \\
\forall(t : t \in \{ x \}; S : |t| = \infty \lor Q.(last.t)) & \wedge \forall(t : t \in \{ x \}; S : |t| < \infty \lor true.(last.t)) \\
= & \{ \text{ predicate calculus } \} \\
\forall(t : t \in \{ x \}; S : |t| < \infty \lor Q.(last.t)) & \wedge \forall(t : t \in \{ x \}; S : |t| = \infty \lor \text{false.(last.t))} \\
= & \{ (197); \text{ definition of } wp \} \\
wp.S.Q.x & \\
\end{align*}
\]

Here is another property of \( wp \).

**Theorem**  Law of the Excluded Miracle

\[
wp.S.false = false
\]  \( (200) \)

Proof

\[
\begin{align*}
wp.S.false.x \\
= & \{ (197); \text{ definition of } wp \} \\
\forall(t : t \in \{ x \}; S : |t| < \infty \lor \text{false.(last.t))} & \\
= & \{ \text{ predicate calculus } \} \\
false & \\
\end{align*}
\]

Next, we calculate the \( wp \) of some programs. The first one is \( skip \).

\[
\begin{align*}
wp.skip.Q.x \\
= & \{ (196); \text{ definition of } wp \} \\
\forall(t : t \in \{ x \}; skip : |t| = \infty \lor Q.(last.t)) & \\
= & \{ (180); \text{ skip is right unit of ; } \} \\
\forall(t : t \in \{ x \}; |t| = \infty \lor Q.(last.t)) & \\
= & Q.x \\
\end{align*}
\]

and hence (and by a similar calculation for \( wp \))

\[
wp.skip.Q = Q
\]  \( (201) \)
\[\text{wp}\ skip Q = Q\]  \hspace{1cm} (202)

The next command is \textit{abort}.

\[
\text{wp}\ abort Q x \\
= \{ (196): \text{definition of wp } \} \\
\forall(t : t \in \{x\}; \text{abort : } |t| = \infty \lor Q.(last.t)) \\
= \{ \text{definition of ; and abort } \} \\
\forall(t : t \in \{x^\infty\} : |t| = \infty \lor Q.(last.t)) \\
= \text{true}
\]

and hence

\[
\text{wp}\ abort Q = \text{true} \hspace{1cm} (203)\]

\[
\text{wp}\ abort Q = \text{false} \hspace{1cm} (204)\]

The third command is the assignment \(v := e\).

\[
\text{wp}(v := e). Q x \\
= \{ (196): \text{definition of wp } \} \\
\forall(t : t \in \{x\}; v := e : |t| = \infty \lor Q.(last.t)) \\
= \{ \text{definition of ; and } v := e \} \\
\forall(t : t \in \{x x(v := e,x)\} : |t| = \infty \lor Q.(last.t)) \\
= Q.(x(v := e,x))
\]

and hence

\[
\text{wp}(v := e). Q = Q_e^v \hspace{1cm} (205)\]

\[
\text{wp}(v := e). Q = Q_e^v \hspace{1cm} (206)\]

Next we look at the command constructors.

\[
\text{wp}(S; U). Q x \\
= \{ (196): \text{definition of wp } \} \]
\[ \forall (t : t \in \{ x \}; S) : U : |t| = \infty \lor Q.(last.t) \]

\[ = \{ \text{change dummy } t := s; u \} \]

\[ \forall (s, u : s \in \{ x \}; S \land u \in \{ \text{last.s} \}; U : |s; u| = \infty \lor Q.(last.(s; u))) \]

\[ = \]

\[ \forall (s : s \in \{ x \}; S : |s| = \infty \lor \forall (u : u \in \{ \text{last.s} \}; U : |u| = \infty \lor Q.(last.u))) \]

\[ = \]

\[ \forall (s : s \in \{ x \}; S : |s| = \infty \lor \text{wp}.U.Q.(last.s) \]

\[ = \{ (196): \text{definition of } \text{wp} \} \]

\[ \text{wp}.S.(\text{wp}.U.Q).x \]

and hence

\[ \text{wp}.(S; U) = (\text{wp}.S) \circ (\text{wp}.U) \]

\[ \text{wp}.(S; U) = (\text{wp}.S) \circ (\text{wp}.U) \quad (207) \]

\[ \text{wp}.\text{if}\ [(i :: b_i \rightarrow s_i) \text{ fl.Q}.x \]

\[ = \{ \text{definitions of IF and wp} \} \]

\[ \forall (t : t \in \{ x \}; (\cup (i :: b_i; s_i) \cup \forall (i :: \neg b_i; \text{abort}) : |t| = \infty \lor Q.(last.t)) \]

\[ = \{ \text{all traces in abort are infinite} \} \]

\[ \forall (t : t \in \{ x \}; \cup (i :: b_i; s_i) : |t| = \infty \lor Q.(last.t)) \]

\[ = \{ (179) \} \]

\[ \forall (t : t \in \cup (i :: \{ x \}; b_i; s_i) : |t| = \infty \lor Q.(last.t)) \]

\[ = \]

\[ \forall (t : t \in \cup (i :: b_i; x : \{ x \}; s_i) : |t| = \infty \lor Q.(last.t)) \]

\[ = \]

\[ \forall (i :: b_i; x : \forall (t : t \in \{ x \}; s_i : |t| = \infty \lor Q.(last.t))) \]

\[ = \{ (196): \text{definition of wp} \} \]

\[ \forall (i :: b_i; x : \text{wp}.s_i; Q).x \]

and hence

\[ \text{wp}.\text{if}\ [(i :: b_i \rightarrow s_i) \text{ fl.Q} = \forall (i :: b_i ; \text{wp}.s_i, Q) \]

\[ \text{wp}.\text{if}\ [(i :: b_i \rightarrow s_i) \text{ fl.Q} = \exists (i :: b_i) \land \forall (i :: b_i ; \text{wp}.s_i, Q) \]

\[ \text{wp}.\text{if}\ [(i :: b_i \rightarrow s_i) \text{ fl.Q} = \forall (i :: b_i ; \text{wp}.s_i, Q) \]

\[ \text{wp}.\text{if}\ [(i :: b_i \rightarrow s_i) \text{ fl.Q} = \exists (i :: b_i) \land \forall (i :: b_i ; \text{wp}.s_i, Q) \]

\[ \text{wp}.\text{if}\ [(i :: b_i \rightarrow s_i) \text{ fl.Q} = \forall (i :: b_i ; \text{wp}.s_i, Q) \]

\[ \text{wp}.\text{if}\ [(i :: b_i \rightarrow s_i) \text{ fl.Q} = \exists (i :: b_i) \land \forall (i :: b_i ; \text{wp}.s_i, Q) \]
\[ wlp.DO.Q \]
\[ = \{ (189) \} \]
\[ wlp.if b \rightarrow s; DO|\neg b \rightarrow \text{skip fl}.Q \]
\[ = \{ \text{see result for } IF \} \]
\[ (b \Rightarrow wlp.(s; DO).Q) \land (\neg b \Rightarrow wlp.skip.Q) \]
\[ = \{ (94) \} \]
\[ (b \land wlp.(s; DO).Q) \lor (\neg b \land wlp.skip.Q) \]
\[ = \{ \text{see results for } ; \text{ and } \text{skip} \} \]
\[ (b \land wlp.s.(wp.DO.Q)) \lor (\neg b \land Q) \]

and hence \( wlp.DO.Q \) is a solution of equation ( \( Y \) is a predicate)

\[ Y : Y = (b \land wlp.s.Y) \lor (\neg b \land Q) \quad (211) \]

\( wp.DO.Q \) is a solution of

\[ Y : Y = (b \land wp.s.Y) \lor (\neg b \land Q) \quad (212) \]

**Theorem**

\( wlp.DO.Q \) is the highest solution of (211).

**Proof**

Let \( Y \) be any solution of (211). We have to show \( Y \Rightarrow wlp.DO.Q \), that is,

\[ Y \Rightarrow wlp.DO.Q \]
\[ = \]
\[ \forall(x :: Y.x \Rightarrow wlp.DO.Q.x) \]
\[ = \]
\[ \forall(x :: Y.x \Rightarrow \forall(t : t \in \{x\}; DO : |t| = \infty \lor Q.(last.t))) \]
\[ = \]
\[ \forall(x : t ; Y.x \land t \in \{x\}; DO \land |t| < \infty : Q.(last.t)) \]

First, we prove by induction,

\[ t \in \{x\}; (b?; s)^n \Rightarrow Y.(last.t) \]

while assuming \( Y.x \land |t| < \infty \).

\( n = 0 \):
\[ t \in \{ x \}; (b?; s)^0 \]
\[ = \]
\[ t = x \]
\[ \Rightarrow \quad \{ \ Y.x \ \} \]
\[ Y.(\text{last}.t) \]

\( n \geq 0: \)

\[ t \in \{ x \}; (b?; s)^{n+1} \]
\[ \Rightarrow \]
\[ \exists (u, v :: |u| < \infty \land u \in \{ x \}; (b?; s)^n \land v \in \{ \text{last}.u \}; b?; s \land t = u; v) \]
\[ \Rightarrow \quad \{ \ \text{induction hypothesis} \ \} \]
\[ \exists (u, v :: Y.(\text{last}.u) \land v \in \{ \text{last}.u \}; b?; s \land t = u; v) \]
\[ \Rightarrow \quad \{ \ Y \text{ solves (211)}; v \in \{ \text{last}.u \}; b?; s \Rightarrow b.(\text{last}.u) \ \} \]
\[ \exists (u, v :: \text{wp}s.Y.(\text{last}.u) \land v \in \{ \text{last}.u \}; b?; s \land t = u; v) \]
\[ \Rightarrow \quad \{ \ \text{last}.u = v.0 \ \} \]
\[ \exists (v :: \text{wp}s.Y.(v.0) \land v \in \{ v.0 \}; s \land \text{last}.t = \text{last}.v) \]
\[ \Rightarrow \quad \{ \ |v| < \infty \ \} \]
\[ \exists (v :: Y.(\text{last}.v) \land \text{last}.t = \text{last}.v) \]
\[ \Rightarrow \]
\[ Y.(\text{last}.t) \]

Next we prove the theorem.

\[ Y.x \land |t| < \infty \land t \in \{ x \}; DO \]
\[ \Rightarrow \quad \{ \ \text{definition DO} \ \} \]
\[ \exists (n :: t \in \{ x \}; (b?; s)^n \land \neg b?) \]
\[ = \]
\[ \exists (n, u, y :: t = uy \land uy \in \{ x \}; (b?; s)^n \land \neg b.y) \]
\[ \Rightarrow \quad \{ \ \text{see result above} \ \} \]
\[ \exists (u, y :: t = uy \land Y.y \land \neg b.y) \]
\[ \Rightarrow \quad \{ \ Y \text{ solves (211)} \ \} \]
\[ \exists (u, y :: t = uy \land Q.y) \]
\[ \Rightarrow \]

\[ Q.(\text{last}.t) \]

\[ \square \]

**Theorem**

\[ \text{wp}.DO.Q \text{ is the lowest solution of (212).} \] \hspace{1cm} (214)

**Proof**

We need to show \( \text{wp}.DO.Q \Rightarrow Y \) for any solution \( Y \) of (212), that is,

\[ t \in \{ x \}; DO \land |t| < \infty \land Q.(\text{last}.t) \Rightarrow Y.x \]

for every trace \( t \) and state \( x \). We have

\[ t \in \{ x \}; DO \land |t| < \infty \]

\[ = \]

\[ t \in \{ x \}; (b?; s)^{\infty} \land \neg b? \land |t| < \infty \]

\[ = \]

\[ t \in \{ x \}; (\text{loop}(b?; s) \cup \cup (n : n \geq 0 : (b?; s)^n)); \neg b? \land |t| < \infty \]

\[ = \]

\[ t \in \{ x \}; \cup (n : n \geq 0 : (b?; s)^n)) \land \neg b? \land |t| < \infty \]

and we prove by induction on \( n \) that the conjunction of \( Q.(\text{last}.t) \) and the last line implies \( Y.x \).

\( n = 0 : \)

\[ t \in \{ x \}; \neg b?; \text{skip} \land |t| < \infty \land Q.(\text{last}.t) \]

\[ = \]

\[ t = xx \land \neg b.x \land Q.x \]

\[ \Rightarrow \]

\[ \{ Y \text{ solves (212) } \} \]

\[ Y.x \]

\( n \geq 0 : \)

\[ t \in \{ x \}; (b?; s)^{n+1} \land \neg b? \land |t| < \infty \land Q.(\text{last}.t) \]

\[ = \]

\[ \forall (y, u : t \in \{ x \}; b?; s; \{ y \}; u \land u \in \{ y \}; (b?; s)^n); \neg b? \land Q.(\text{last}.t) \land |t| < \infty) \]
\[ \Rightarrow \{ \text{induction hypothesis} \} \]
\[ \forall (y, u :: t \in \{ x \}; b :: \{ y \}; u \land Y.y \land |t| < \infty) \]
\[ \Rightarrow \{ \text{definition } wp \} \]
\[ b.x \land \text{wp.s.Y.x} \]
\[ \Rightarrow \{ Y \text{ solves (212)} \} \]
\[ Y.x \]

17 Nondeterminism

We say that a program is deterministic if every possible outcome of the program is unavoidable. With \( wp \) and \( wlp \) we express properties about the final state only, so “outcomes” are final states.

Program \( S \) is deterministic \( \Leftrightarrow \forall (Q :: wp.S.\neg Q = \neg wlp.S.Q) \) (215)

A rewrite of this expression reveals

\[ wp.S.\neg Q.x = \neg wlp.S.Q.x \]
\[ = \{ (197) \text{ and (196); definition of } wp \text{ and } wlp \} \]
\[ \forall (t : t \in \{ x \}; S :: |t| < \infty \land \neg Q.(last.t)) = \neg \forall (t : t \in \{ x \}; S :: |t| = \infty \lor Q.(last.t)) \]
\[ = \{ \text{predicate calculus} \} \]
\[ \forall (t : t \in \{ x \}; S :: |t| < \infty \land \neg Q.(last.t)) = \exists (t : t \in \{ x \}; S :: |t| < \infty \land \neg Q.(last.t)) \]

and the latter line corresponds to our informal definition.

Theorem

If \( S \) is deterministic

\( wp.S \) is universally \( \lor \neg \)-distributive

\( wlp.S \) is positively \( \lor \neg \)-distributive

Proof

\[ wp.S.\exists (Q :: Q \in V : Q) \]
\[ = \{ (215) \} \]
\[ \neg wlp.S.\neg \exists (Q :: Q \in V : Q) \]
It is easily checked that skip, abort, and $v := e$ are deterministic. Also, sequential composition of deterministic commands is deterministic. However, the if-command need not be deterministic. The nondeterminism may even be unbounded, as shown by program \( UN \), which assigns an arbitrary natural value to variable \( v \).

\[
UN = \text{if } \{(i : i \geq 0 : \text{true } \rightarrow v := i)\} \text{fi}
\]

We have

\[
wp.\ UN.\ Q \ = \ wp.\ UN.\ Q \ = \ \forall(i : i \geq 0 : Q^v_i)
\]

and we show that \( wp.\ UN \) is not even \( \forall \)-continuous. In particular, we show that \( wp.\ UN \) does not \( \forall \)-distribute over set \( \{j : j \geq v\} \) which is a chain. We have

\[
wp.\ UN.\ \exists(j : j \geq v)
\]

\[
= \{ \text{take } j := v \} \quad \text{wp.\ UN.\ true}
\]

\[
= \quad \text{true}
\]

and

\[
\exists(j : wp.\ UN.(j \geq v))
\]

\[
= \quad \exists(j : \forall(i : (j \geq v)_i^v))
\]

\[
= \quad \{ \text{substitution} \}
\]

\[
\exists(j : \forall(i : j \geq i))
\]
\[ \exists (j :: \text{false}) \]
\[ = \text{false} \]

However, we have the following result.

**Theorem**

If all \( wlp.s_i \) are positively \( \lor \)-continuous and the range of \( i \) is finite,

\[ wlp.if ((i :: b_i \rightarrow s_i) \text{ if } i is positively \lor\text{-continuous.} \tag{218} \]

**Proof**

We have \( wlp.if ((i :: b_i \rightarrow s_i) \text{ if } i is positively \lor\text{-continuous.} \). Since every constant function is positively \( \lor \)-continuous, and because of (103), we have that \( -b_i \lor wlp.s_i \) is positively \( \lor \)-continuous. Because of (139) and because the range of \( i \) is finite, we have that \( \forall (i :: -b_i \lor wlp.s_i) \) and hence also that \( wlp.if ((i :: b_i \rightarrow s_i) \text{ if } i is positively \lor\text{-continuous.} \)

\[ \square \]

**Theorem**

If \( wlp.S \) is positively \( \lor \)-continuous, then so is \( wp.S \).

**Proof**

Immediate from (199) and (105).

\[ \square \]

Weakest preconditions were originally studied in the context of programs whose nondeterminism is bounded. As a result, their \( wp \) is positively \( \lor \)-continuous and hence, according to (152), (212), and (214)

\[ wp.do b \rightarrow s od.Q = \exists (i :: i \geq 0 : f^i \text{false}) \]

where

\[ f.X = (-b \lor wp.s.X) \land (b \lor Q) \]

and this is then used as the definition of \( wp.DO \). (Notice that \( f \) depends on \( b, s, \) and \( Q \).) We prefer characterization (214). Programs whose nondeterminism is unbounded may not be easy to implement, but they are often interesting stepping stones in the development of programs from their specifications.

**Theorem**

If \( s \) is deterministic, then so is \( do b \rightarrow s od \).

**Proof**
true

$$\begin{align*}
wlp.DO.\neg X &= \{ \ (211) \ \} \\
wp.DO.\neg X &= \{ \ Y: Y = (b \land \ wp.s.Y) \lor (-b \land \neg X) \} \\
wp.DO.\neg X &= \{ \ (91): \ De\ Morgan \ \} \\
\neg wp.DO.\neg X &= \{ \ Y: Y = (b \lor \neg wp.s.\neg Y) \land (b \lor X) \} \\
\neg wp.DO.\neg X &= \{ \ (s\ is\ deterministic) \} \\
\neg wp.DO.\neg X &= \{ \ Y: Y = (b \lor \ wp.s.Y) \land (b \lor X) \} \\
\neg wp.DO.\neg X &= \{ \ (212) \ \} \\
wp.DO.X &= \\\nwp.DO.X &= \\\nDO \ is \ deterministic
\end{align*}$$

\[\Box\]

18 Axiomatic semantics

In the preceding sections, we have identified a program with its set of traces, and we have derived its \(wp\) and \(wlp\) from this operational semantics. We might have followed an alternative path in which we state the \(wp\) and \(wlp\) of every program and never mention the operational semantics. This is referred to as axiomatic semantics. If one postulates, for example, each \(wlp\) then (198) can be proved by induction: first it is shown for the basic commands (\(skip\), \(abort\), and assignment) and then inductively for sequential, alternative, and repetitive compositions.

In the present section we forget about the operational semantics. Because a program’s \(wp\) is of more practical importance than its \(wlp\), we focus on the former. To emphasize this focus, we identify a program with its \(wp\), so that we have (\(v := e\)). \(Q = Q_v\) and \(skip.Q = Q\), or, if we want to write it without \(Q\),

$$\begin{align*}
skip &= id \\
abort &= false
\end{align*}$$

The definition of \(abort\) shows that it is the lifting of the constant predicate \(false\) (or \(\bot\)) to the level of predicate transformers. It suggests that we also introduce

$$\begin{align*}
magic &= true
\end{align*}$$
but, as its name suggests, its implementation may not be obvious. In fact, *magic* violates the Law of the Excluded Miracle, which implies that no operational semantics in the sense of the previous section exists for *magic*. A command that violates the Law of the Excluded Miracle is called a partial command. In our discussion of an axiomatic semantics, we do not assume that commands satisfy the Law of the Excluded Miracle, in fact we do not even assume that they are positively \(\land\)-distributive. Our only restriction is that every command be monotonic. In the sequel, a command can be any monotonic predicate transformer, and not only one for which we have given the syntactic representation as a program text. From (65) we know that these commands form a complete lattice, with bottom *abort* and top *magic*. Whenever we introduce a construct that composes commands, we have the proof obligation to show that the composite is monotonic given that the components are. For example,

\[ S; T = S \circ T \]

and the monotonicity of \( S; T \) follows from (13).

The positively \(\land\)-distributive predicate transformers do not form a complete lattice. However, from (73) we know that the highest lower bound of any set of positively \(\land\)-distributive predicate transformers is a positively \(\land\)-distributive itself. Most of our programs are positively \(\land\)-distributive, and some of them even universally \(\land\)-distributive, as noted when they are introduced. Both *skip* and *magic* are universally \(\land\)-distributive; *abort* is not universally but positively \(\land\)-distributive. From (102) we infer that \( S; T \) has every distributivity property shared by \( S \) and \( T \).

As a property of \( ; \) we have

\[ \text{skip}; S = S = S; \text{skip} \]

since \( \text{id} \) is the left and right identity element of function composition. Also, because function composition is associative, we have

\[ ; \text{ is associative} \]

(222)

Combinations of \( ; \) with *abort* and *miracle* give

\[ \text{miracle}; S = \text{miracle} \]

(223)

\[ (S; \text{miracle} = \text{miracle}) = (S.\text{true} = \text{true}) \]

(224)

\[ \text{abort}; S = \text{abort} \]

(225)

\[ (S; \text{abort} = \text{abort}) = (S.\text{false} = \text{false}) \]

(226)

Before we proceed with the if- and do-commands, we see how the lifting of the ordering on predicates to predicate transformers can be interpreted.

\[ S \leq T \]
\[
\forall (Q:: S.Q \Rightarrow T.Q)
\]

\[
\forall (P, Q:: (P \Rightarrow S.Q) \Rightarrow (P \Rightarrow T.Q))
\]

\[
\forall (P, Q:: \{ P \} S\{ Q \} \Rightarrow \{ P \} T\{ Q \})
\]

The last line can be read as: \( T \) satisfies every specification that \( S \) satisfies, and hence, \( S \leq T \) expresses that \( S \) can be refined by \( T \). From (16) and (17) we conclude

\[
S \leq S' \land T \leq T' \Rightarrow S; T \leq S'; T'
\]

which shows that a sequential composition of commands can be refined by refining its components. Given that we have a partial order \( \leq \) on programs, we can derive the \( \uparrow \) and \( \rhd \) on programs. They are usually written as \( \vee \) and \( \mid \).

\[
(S \vee T).Q = S.Q \vee T.Q
\]

\[
(S \mid T).Q = S.Q \land T.Q
\]

If \( S \) and \( T \) are monotonic, then so are \( S \vee T \) and \( S \mid T \). If \( S \) and \( T \) are positively \( \land \)-distributive, then so is \( S \mid T \) on account of (73), but \( S \vee T \) need not be positively \( \land \)-distributive. Command \textit{abort} is the unit element of \( \lor \) and the zero element of \( \mid \), whereas \textit{magic} is the unit element of \( \mid \) and the zero element of \( \lor \). On account of (63), both \( \lor \) and \( \mid \) are monotonic in \( S \) and \( T \). They have a lower binding power than \( ; \) has. Of course, we have

\[
(S \leq T) = (S \mid T = S) = (S \vee T = T)
\]

We have seen the usefulness of \( b? \) in the section on operational semantics. We introduce a similar construct in our program notation, and then add another one. They are called the assert and guard command respectively.

\[
\{ P \}.Q = P \land Q
\]

\[
[P,Q = P \Rightarrow Q
\]

Both are positively \( \land \)-distributive, and hence, also monotonic functions of \( Q \). Both \( \{ P \} \) and \( [P] \) act as \textit{skip} if \( P \) holds. If \( P \) does not hold, then \( \{ P \} \) acts as \textit{abort} whereas \( [P] \) acts as \textit{magic}. Notice

\[
\{ \text{false} \} = \text{abort}
\]

\[
[\text{false}] = \text{magic}
\]

\[
\{ \text{true} \} = [\text{true}] = \text{skip}
\]
Sequences of guard and assert statements can be combined.

\[ \{P\}; \{Q\} = \{P \land Q\} \]  \hspace{1cm} (227)

\[ [P]; [Q] = [P \land Q] \]  \hspace{1cm} (228)

We can now define the guarded command \( P \rightarrow S \) as an abbreviation for \([P]; S\). Since sequential composition is monotonic, \( P \rightarrow S \) is monotonic with respect to \( S \). The binding power of \( \rightarrow \) is lower than that of \( ; \) and higher than that of \( \lbrack \rbrack \). We have

\[ [P]; Q \rightarrow S = P \land Q \rightarrow S \]  \hspace{1cm} (229)

and from the associativity of \( ; \) we have

\[ (b \rightarrow S); T = b \rightarrow S; T \]  \hspace{1cm} (230)

The if-command can be defined as

\[
\text{if } \lbrack(i :: b_i \rightarrow S_i) \cdot \rbrack = \{\exists(i :: b_i)\}; \lbrack(i :: b_i \rightarrow S_i) \cdot \rbrack
\]

Since \( \lbrack \rbrack \) and \( \rightarrow \) are monotonic, the if-command is monotonic with respect to any of the commands \( S_i \).

The shape of the formula suggests that we might define the if-command for any (partial) command \( S \) as

\[
\text{if } S \cdot \rbrack = \{\neg S. \text{false}\}; S
\]

but we refrain from doing so because this construct is not monotonic with respect to \( S \). For example, we have \( \text{skip} \leq \text{magic} \) but not \( \text{skip} \leq \text{abort} \), and yet \( \text{if skip \cdot \rbrack = \text{skip} \) and \( \text{if magic \cdot \rbrack = \text{abort} \)

We can rewrite the if-command as

\[
\text{if } \lbrack(i :: b_i \rightarrow S_i) \cdot \rbrack = \{\exists(i :: b_i)\}; \lbrack(i :: b_i); S_i \rbrack
\]

which suggest that we might also have another if-command, such as

\[
\text{if } \circ(i :: b_i \rightarrow S_i) \cdot \rbrack = [\exists(i :: b_i)]; \lor(i :: \{b_i\}; S_i)
\]

or maybe

\[
\text{if } \circ(i :: b_i \rightarrow S_i) \cdot \rbrack = \lor(i :: \{b_i\}; S_i)
\]

The latter corresponds to angelic choice whereas the original one corresponds to demonic choice. When the \( b_i \) are mutually exclusive, the two commands coincide.

According to (214), \( DO \) is the lowest solution of

\[
Y : Y = (b \land wp.s.Y) \lor (\neg b \land Q)
\]
and hence we propose

\[ \texttt{do } b \to s \texttt{ od } = \langle Y : Y = \texttt{if } b \to s ; Y \mid \neg b \to \texttt{skip} \rangle \]

which is equivalent to

\[ \texttt{do } b \to s \texttt{ od } = \langle Y : Y = [b] ; s ; Y \mid \neg b \rangle \]

and to

\[ \texttt{do } b \to s \texttt{ od } = \langle Y : Y = \{b\} ; s ; Y \lor \{\neg b\} \rangle \]

Similar to the rejected attempt at defining \texttt{if } \texttt{S fl} one might try to introduce \texttt{do S od}. In fact, this is done in [11] as follows.

\[ \texttt{do } S \texttt{ od } = \langle Y : Y = S ; Y \mid [S, \texttt{false}] \rangle \]

We refrain from doing so because this construct is not monotonic in \texttt{S} as shown by the following example. We have \texttt{do } \texttt{x} \neq 0 \to \texttt{x} := 0 \texttt{ od } = \texttt{x} := 0 \texttt{ and } \texttt{do magic od } = \texttt{skip}, and \texttt{x} \neq 0 \to \texttt{x} := 0 \leq \texttt{magic} but not \texttt{x} := 0 \leq \texttt{skip}. We stick to our earlier definition of the loop \texttt{do } \texttt{b} \to \texttt{s od}. According to (130), it is a monotonic function of \texttt{s}.

We now investigate some distribution properties of programs. From (71) we get

\[ \{i : S_i\}; T = \{i : S_i; T\} \]

but for

\[ T; \{i : S_i\} = \{i : T; S_i\} \]

we need \texttt{and distributivity of } \texttt{T}. If \texttt{T} is universally \texttt{and}-distributive, we have (232). If \texttt{T} is positively \texttt{and}-distributive, we have (232) for nonempty set \{i : S_i\}. If \texttt{T} is finitely \texttt{and}-distributive, we have (232) for nonempty, finite set \{i : S_i\}. We give the proof of the latter.

\[ (T; (U \mid V)).Q \]

\[ = \{ \text{ definition of } \mid \text{ and } \} \]

\[ T.(U.Q) \land V.Q \]

\[ = \{ T \text{ is finitely } \land \text{-distributive } \} \]

\[ T.(U.Q) \land T.(V.Q) \]

\[ = \{ \text{ definition of } \mid \text{ and } \} \]

\[ (T; U \mid T; V).Q \]

From
\[
\text{if } \{(i :: b_i \rightarrow S_i) \text{ fi}; T \\
= \{ \exists (i :: b_i); \{(i :: b_i \rightarrow S_i); T \text{ fi} \}
= \{ \exists (i :: b_i); \{(i :: (b_i \rightarrow S_i); T \text{ fi} \}
= \{ \exists (i :: b_i); \{(i :: b_i \rightarrow S_i); T \text{ fi} \}
= \text{if } \{(i :: b_i \rightarrow S_i) \text{ fi}; T
\]
we have
\[
\text{if } \{(i :: b_i \rightarrow S_i) \text{ fi}; T = \text{if } \{(i :: b_i \rightarrow S_i); T \text{ fi}
\]
In general, we do not have
\[
T; \text{if } \{(i :: b_i \rightarrow S_i) \text{ fi} = \text{if } \{(i :: b_i \rightarrow T; S_i) \text{ fi}
\]
not even when T is \wedge\text{-distributivity.}
Let \( DO = \text{do } b \rightarrow S \text{ od } \) and \( DO' = \text{do } b' \rightarrow S' \text{ od } \). From (212) we conclude
\[
DO. Q = DO.(Q \wedge \neg b) \tag{234}
\]
and
\[
(Q \Rightarrow \neg b) \Rightarrow (Q \Rightarrow (DO = \text{skip})) \tag{235}
\]
Given \( (Q \Rightarrow \neg b) \) and \( Q \), we have
\[
D. X
= \{ DO \text{ solves (212) } \}
(b \wedge S.X) \lor (\neg b \land X)
= \{ \text{from } (Q \Rightarrow \neg b) \text{ and } Q, \text{ we have } \neg b \}
X
\]
If \( b' \Rightarrow b \) then
\[
DO; DO' = DO
\]
\[(DO; DO').Q\]
\[=\]
\[DO.(DO'.Q)\]
\[=\]
\[\{ (234) \}\]
\[DO.(DO'.Q \land \neg b)\]
\[=\]
\[\{ (235) \}\]
\[DO.(Q \land \neg b)\]
\[=\]
\[\{ (234) \}\]
\[DO.Q\]

As a result, we have

\[DO; DO = DO\]

Also, we have

\[DO = \text{do } b \rightarrow DO \text{ od}\]

A slightly simpler looping construct is sometimes written as

\[S^* = [Y : Y = S; Y \parallel \text{skip}]\]

and from

\[(b \rightarrow s)^*; [-b]\]

\[=\]
\[[Y : Y = b \rightarrow s; Y \parallel \text{skip}]; [-b]\]
\[=\]
\[\{ (172) [h.x.y := b \rightarrow s; y | x, g := [-b]] \}\]
\[[Y : Y = b \rightarrow s; Y \parallel [-b]]\]
\[=\]
\[\text{do } b \rightarrow s \text{ od}\]

we have

\[\text{do } b \rightarrow s \text{ od } = (b \rightarrow s)^*; [-b]\]

Observe that \(S^*\) is identical to the lowest \(\downarrow\)-closure \(S \downarrow\) of \(S\). Hence, we have from (151),

\[(A \parallel B)^* = A^*; (B; A^*)^*\]
Another interesting property of this loop is the so-called leapfrog rule. It holds for all \( B \) and for all finitely \( \land \)-distributive \( A \).

\[
A; (B; A)^* = (A; B)^*; A
\]

\[
A; (B; A)^*
= A; [Y : Y = B; A; Y \mid skip]
= A; \{ Y : Y = B; A; Y \mid skip : Y \}
= \{ (46) \}
\{ Y : Y = B; A; Y \mid skip : A; Y \}
= \{ introduce X \}
\{ X, Y : X = A; Y \land Y = B; X \parallel skip : X \}
= \{ eliminate Y \}
\{ X : X = A; (B; X \parallel skip) : X \}
= \{ A is finitely \( \land \)-distributive \}
\{ X : X = A; B; X \mid A : X \}
= \{ (172) \mid h, x, y := A; B; y[x, g := A] \}
\{ X : X = A; B; X \mid skip : X \}; A
= (A; B)^*; A
\]

From these two, we derive Greg Nelson’s theorem that

\[
\text{do } b \rightarrow s \mid c \rightarrow t \text{ od } = \text{ do } b \rightarrow s \text{ od; do } c \rightarrow t; \text{ do } b \rightarrow s \text{ od od}
\]

provided \( c \Rightarrow \neg b \).

\[
\text{do } b \rightarrow s \parallel c \rightarrow t \text{ od }
= (b \rightarrow s \mid c \rightarrow t)^*; \lnot (b \lor c)
= (b \rightarrow s \mid c \rightarrow t)^*; \lnot b; \lnot c
= \]
\[(b \to s)^*; (c \to t; (b \to s)^*)^*; [-b]; [-c] = \begin{cases} c = -b \land c \end{cases}
\]
\[(b \to s)^*; (-b \land c \to t; (b \to s)^*)^*; [-b]; [-c] = \begin{cases} \text{leapfrog; } [-b] \text{ is } \land\text{-distributive } \end{cases}
\]
\[(b \to s)^*; [-b]; (c \to t; (b \to s)^*)^*; [-b]^*; [-c] = \begin{cases} \text{do } b \to s \text{ od; } (c \to t; \text{ do } b \to s \text{ od})^*; [-c] \end{cases}
\]
\[= \begin{cases} \text{do } b \to s \text{ od; } \text{ do } c \to t; \text{ do } b \to s \text{ od od} \end{cases}
\]

19 Needs work

Invariance theorem(s).
Strongest postcondition.
Program inversion.

20 Appendix

Table of binding powers:
\[
\begin{array}{c c c c c}
\circlearrowleft & \circlearrowright \\
\uparrow & \uparrow \\
\leq & \geq & \subseteq & \supseteq \\
\& & \vee \\
\Rightarrow & \Leftarrow
\end{array}
\]

References


