WEAKEST PREEXPECTATION REASONING 4

Lifting weakest precondition reasoning and anticipated value reasoning for (non)deterministic programs (see Chapter 2) to probabilistic programs *with nondeterminism* (see Chapter 3) is the subject matter of this chapter. We will present so–called *expectation transformer* calculi for quantitative reasoning about partial and total correctness of such programs. Furthermore, we study some basic properties of these transformers.

The idea of expectation transformers goes back to Kozen's seminal work on probabilistic propositional dynamic logic (PPDL) [Koz85]. PPDL is a modal logic for reasoning about (in our terminology) tame probabilistic programs, i.e. probabilistic programs *without* nondeterminism. Later, McIver & Morgan (re)incorporated nondeterministic choice and developed the weakest preexpectation calculus [MMS96; MM05]. We will present the calculus in the style of McIver & Morgan here, although it should be noted that — on fully probabilistic programs — their calculus is very closely related to PPDL (basically, the two formalisms coincide).

4.1 REASONING ABOUT EXPECTED VALUES

In an effort to enable formal reasoning about probabilistic programs, we now lift the notion of anticipated value reasoning (which subsumes reasoning about predicates; see Section 2.3.2) to *weakest preexpectation reasoning* (which will subsume reasoning about probabilities of events). As a first example, consider the program

$$\{x := 5\}\,[4/5]\,\{x := 10\}\,.$$

In contrast to a deterministic program, the variable x may have value 5 *or* 10 after termination of the program. Hence, there is *no single anticipated value of* x. That fact renders the whole concept of an anticipated value useless *as is* in the context of the above probabilistic program.

The situation is similar to a nondeterministic choice, were we also did not necessarily have a single anticipated value available. More detrimentally even, we had no information whatsoever on what branch is going to be executed. We therefore chose the minimal (demonic nondeterminism) or maximal (angelic nondeterminism) anticipated value of x (see Section 2.3.3). For the program above, this would be 5 or 10, respectively.

For probabilistic choices, the situation is in some sense better: We do have some information on the further course of the execution, namely the

probability with which each of the two branches is executed. While this still does not tell us *for sure* what is going to happen, it does provide us with *quantitative information* that we can sensibly incorporate into reasoning about probabilistic programs.

Imagine in the program above that x is some sort of payoff, penalty, or something alike. Instead of determining a minimal or maximal anticipated value of x, a generally accepted and arguably very important concept in the realm of probability theory is the notion of the *expected value* of x. For the case of payoffs, we would then obtain a mean or average payoff.

In order to determine the expected value of x in the program above, we need to average the anticipated value of x from the left and the right branch, which is $\mathsf{wp}\,[\![x := 5]\!]\,(x) = 5$ and $\mathsf{wp}\,[\![x := 10]\!]\,(x) = 10$, respectively. Weighting those anticipated values with the probabilities with which the branches are executed then gives us the expected value

$$\tfrac{4}{5} \cdot \mathsf{wp}\,[\![x := 5]\!]\,(x) + \tfrac{1}{5} \cdot \mathsf{wp}\,[\![x := 10]\!]\,(x) \;=\; \tfrac{4}{5} \cdot 5 + \tfrac{1}{5} \cdot 10 \;=\; 6\,.$$

The explanations above already provide most of the intuition we need for extending the anticipated value calculus to reasoning about expected values for probabilistic programs. In the following, we gradually develop the weakest preexpectation calculus which allows us to conduct this sort of reasoning in a systematic way.

4.1.1 *Weakest Preexpectations*

Similarly to anticipated value reasoning (see Section 2.3), we are given a program C, an initial state σ, and a function f mapping (final) program states to positive reals or infinity. Let us first — for now — leave nondeterminism out of the picture and consider only tame programs C.

For the function f we would like to know its *expected value* with respect to the distribution over final states obtained by executing C on σ (i.e. the distribution $[\![C]\!]_\sigma$, see Section 3.3.2). Such a function f can thus simply be thought of as a random variable. We follow a widespread terminology here[1] and refer to the class of random variables we use in our setting as to *expectations*:

Definition 4.1 (Expectations):

A. *The set of expectations, denoted* $\mathbb{E}$*, is defined to coincide with the set of anticipations (see* Definition 2.10*), i.e.*

$$\mathbb{E} \;=\; \left\{ f \;\middle|\; f \colon \Sigma \to \mathbb{R}^{\infty}_{\geq 0} \right\} \quad = \mathbb{A}\,.$$

Consequently, the complete lattice $(\mathbb{E}, \preceq)$*, its least element, and the construction of suprema is defined exactly as for anticipations,*

1 See e.g. [CNZ17], [Fen+17], [Cha+16], [CS14], [Coc14], [GKM14], [CS13], [Kat+10], [APM09], [MM05], [Mon05], or [MM99].

i.e. the order relation is given by

$$f_1 \preceq f_2 \qquad \text{iff} \qquad \forall \sigma \in \Sigma\colon \quad f_1(\sigma) \leq f_2(\sigma)\,;$$

the least element is

$$\lambda\sigma.\ 0\,,$$

which we overloadingly denote by 0*; and the supremum of a subset* $S \subseteq \mathbb{E}$ *is constructed pointwise by*

$$\sup S \;=\; \lambda\sigma.\ \sup_{f \in S} f(\sigma)\,.$$

We write $f \ll g$ *to indicate that* f *is everywhere smaller than* g*, i.e.*

$$f \ll g \qquad \text{iff} \qquad \forall \sigma \in \Sigma\colon \quad f(\sigma) < g(\sigma)\,.$$

B. *The set of bounded expectations, denoted* $\mathbb{E}_{\leq \exists b}$ *is defined as*[2]

$$\mathbb{E}_{\leq \exists b} \;=\; \{f \in \mathbb{E} \mid \exists\, b \in \mathbb{R}_{\geq 0}\colon\ f \preceq b\}\,.$$

$(\mathbb{E}_{\leq \exists b}, \preceq)$ *is a lattice with least element 0 (as above) but it is* not complete *since suprema are not guaranteed to exist.*[3]

C. *The set of one–bounded expectations, denoted* $\mathbb{E}_{\leq 1}$ *is defined as*

$$\mathbb{E}_{\leq 1} \;=\; \{f \in \mathbb{E} \mid f \preceq 1\}\,.$$

$(\mathbb{E}_{\leq 1}, \preceq)$ *is a complete lattice with least element 0 (as in* A. *above) and greatest element* $\lambda\sigma.\ 1$ *which we overloadingly denote by* 1*. Suprema are constructed as in* $\mathbb{E}$*.*

We remark that McIver & Morgan's oeuvre on weakest preexpectation reasoning relies in almost its entirety on *bounded* expectations [MM05; KM17] (in particular, see [MM05, p. 25 (especially Footnote 39) and Section 2.11]). Bounded expectations do *not* form a complete lattice and existence of least fixed points is not a consequence of the Kleene fixed point theorem but has to be proven by different means [MM05, Lemma 5.6.8].

We, on the other hand, take a more general view in which expectations may generally be *unbounded* and even evaluate to infinity. Indeed, we depend on these more general unbounded expectations because we will later use expectations to reason about expected runtimes, which in general cannot bounded by a constant but depend on the input.

2 For $b \in \mathbb{R}_{\geq 0}$, we write $f \preceq b$ to mean $f \preceq \lambda\sigma.\ b$.
3 E.g., the set $\{\lambda\sigma.\ 1, \lambda\sigma.\ 2, \lambda\sigma.\ 3, \ldots\} \subset \mathbb{E}_{\leq \exists b}$ has supremum $\lambda\sigma.\ \infty \in \mathbb{E}$ but $\lambda\sigma.\ \infty \notin \mathbb{E}_{\leq \exists b}$.

Reasoning about expected values. Analogously to anticipated value reasoning, we will refer to the expectation whose expected value we want to know as to a *postexpectation*. Given a postexpectation $f \in \mathbb{E}$ and a probabilistic program C, we would like to know a function $g \in \mathbb{E}$ that maps each (initial) state σ to the expected value of f after execution of C on input σ. We call this function g the

weakest preexpectation of C with respect to postexpectation f

and denote it by $\mathsf{wp}\,[\![C]\!]\,(f)$. The characterizing equation of a weakest preexpectation is given by

$$\mathsf{wp}\,[\![C]\!]\,(f) \;=\; \lambda\sigma.\; \int_{\Sigma} f \; d\,[\![C]\!]_\sigma\,, \tag{4.1}$$

where we denote by $\int_\Sigma h\, d\,\mu$ the expected value of expectation (read: random variable) h with respect to a distribution μ over the set of program states Σ.

Example 4.2 (Weakest Preexpectations):
Consider the program

$$\{x := x+5\}\,[4/5]\,\{x := 10\}\,.$$

Suppose we want to know the expected value of x, i.e. the weakest preexpectation of the above program with respect to postexpectation x. This weakest preexpectation is given by

$$\tfrac{4}{5}\cdot(x+5) + \tfrac{1}{5}\cdot 10 \;=\; \frac{4x}{5} + 6\,.$$

When executing the above program on initial state σ, the expected value of x is hence $4\sigma(x)/5 + 6$.

Reasoning about probabilities. An important special case is when the postexpectation given as $[F]$, where F is a predicate. In that case, we can think of F as an event and $\mathsf{wp}\,[\![C]\!]\,([F])(\sigma)$ is then the probability that executing C on input σ will terminate in a final state $\tau \models F$, or in other words: $\mathsf{wp}\,[\![C]\!]\,([F])(\sigma)$ is the probability that event F occurs after termination of C on input σ.

Example 4.3 (Probabilities of Events as Weakest Preexpectations):
Reconsider the program

$$\{x := x+5\}\,[4/5]\,\{x := 10\}$$

from Example 4.2 and take predicate (i.e. event) $F = (x{=}10)$. Then the weakest preexpectation of this program with respect to postexpectation $[F]$ is

$$\tfrac{4}{5} \cdot [x+5=10] + \tfrac{1}{5} \cdot [\mathsf{true}] \;=\; \frac{4 \cdot [x=5]+1}{5}\,.$$

Thus, for any initial state σ, the probability that x is 10 after executing the above program on σ is $(4\cdot 1+1)/5 = 1$ if $\sigma(x) = 5$, and $(4\cdot 0+1)/5 = 1/5$ otherwise.

Reasoning about Nondeterminism. We now bring nondeterminism back into the picture. As we have seen in Section 3.3.2, we have to resolve all nondeterminism occurring along the computation of a pGCL program in order to sensibly obtain a probability distribution over final states. Resolving nondeterminism was achieved by so–called schedulers that resolve all nondeterminism occurring in the computation tree. We thereby obtained a distribution $[\![C]\!]^{\mathfrak{s}}_{\sigma}$ relative to some scheduler $\mathfrak{s}$.

As we can see in Equation 4.1, for characterizing weakest preexpectations we need a probability distribution. For programs with nondeterminism, we will have to choose a scheduler in order to obtain a probability distribution. *A* sensible choice is a scheduler that *minimizes* the preexpectation. The characterizing equation of weakest preexpectations for full pGCL is thus given by

$$\mathsf{wp}\,[\![C]\!]\,(f) \quad = \quad \lambda\sigma.\;\; \inf_{\mathfrak{s}\in\mathsf{Scheds}} \int_{\Sigma} f \; d\,[\![C]\!]^{\mathfrak{s}}_{\sigma}\,. \tag{4.2}$$

The above minimizing exegesis of weakest preexpectations agrees with Dijkstra's original notion of weakest preconditions of (nonprobabilistic) nondeterministic programs. For anticipated value reasoning, we called this interpretation *demonic* nondeterminism.

Angelic nondeterminism, i.e. a *maximizing* exegesis, is in some cases an equally sensible choice: If we think about expected runtimes for instance, a maximizing scheduler cannot even be conceived as very *angelic* in the truest sense of the word, but indeed as a *demonic* worst–case. The characterizing equation of angelic weakest preexpectations for full pGCL is given by

$$\mathsf{awp}\,[\![C]\!]\,(f) \quad = \quad \lambda\sigma.\;\; \sup_{\mathfrak{s}\in\mathsf{Scheds}} \int_{\Sigma} f \; d\,[\![C]\!]^{\mathfrak{s}}_{\sigma}\,. \tag{4.3}$$

4.1.2 *Weakest Liberal Preexpectations*

For deterministic programs, the weakest liberal pre*condition* of a program C with respect to a postcondition (i.e. a predicate) F is a predicate G such that the execution of C on an initial state $\sigma \models G$ will either diverge or terminate in a state $\tau \models F$ (cf. Section 2.2.5). Weakest liberal pre*expectations* are the probabilistic analog to this concept:

For a predicate F, the weakest liberal pre*expectation* of C with respect to postexpectation $[F]$ is an expectation $g \in \mathbb{E}_{\leq 1}$ such that $g(\sigma)$ equals the *probability* that executing C on input σ will either diverge or terminate in a state $\tau \models F$. In other words: $g(\sigma)$ is the probability that event F occurs *if* C terminates on σ. More generally, for any $f \in \mathbb{E}_{\leq 1}$, the

weakest liberal preexpectation of C with respect to postexpectation f,

denoted by $\mathsf{wlp}\,[\![C]\!]\,(f)$, is an expectation in $\mathbb{E}_{\leq 1}$ such that the expected value of f after execution of C on an initial state σ plus the probability that C does not terminate on σ equals $\mathsf{wlp}\,[\![C]\!]\,(f)(\sigma)$, formally

$$\mathsf{wlp}\,[\![C]\!]\,(f) \quad = \quad \lambda\sigma.\ \inf_{\mathfrak{s}\in\mathsf{Scheds}} \int_{\Sigma} f \; d\,[\![C]\!]^{\mathfrak{s}}_{\sigma} \;+\; \left(1 - \int_{\Sigma} 1 \; d\,[\![C]\!]^{\mathfrak{s}}_{\sigma}\right), \tag{4.4}$$

where the infimum on the right–hand–side accounts for possible demonic nondeterminism occurring in the program.

EXAMPLE 4.4 (Weakest Liberal Preexpectations):

A. Consider the program

$$\{\texttt{diverge}\}\,[1/3]\,\{x := 10\}$$

and take predicate (read: event) $F = (x{=}10)$. Then the weakest liberal preexpectation of this program with respect to postexpectation $[F]$ is

$$\tfrac{1}{3}\cdot 1 + \tfrac{2}{3}\cdot 1 \;=\; 1.$$

B. Consider the program

$$\{\texttt{diverge}\}\,[1/3]\,\{x := x+5\}\,.$$

Then the weakest liberal preexpectation of the above program with respect to postexpectation $[F]$ from A. is

$$\tfrac{1}{3}\cdot 1 + \tfrac{2}{3}\cdot [x+5=10] \;=\; \frac{1+2\cdot[x=5]}{3}.$$

C. Consider the program

$$\begin{array}{l} c := 1 \\ \texttt{while}(c=1)\{ \\ \quad \{\texttt{diverge}\}\,[1/2]\,\{x := x+1\}\,; \\ \quad \{\texttt{skip}\}\,[1/2]\,\{c := 0\} \\ \} \end{array}$$

and take the event that x is even. Then the weakest liberal preexpectation of this program with respect to postexpectation $[x \text{ even}]$ is

$$\frac{2}{3} + \frac{4 \cdot [x \text{ odd}]}{15} + \frac{[x \text{ even}]}{15} .$$

4.1.3 *The Weakest Preexpectation Calculus*

So far, we have seen characterizations for (angelic) weakest (liberal) preexpectations, but we have not seen how to systematically determine them given a concrete program and postexpectation. It turns out that for (angelic) weakest preexpectations this can be done analogously to anticipated value reasoning (see Section 2.3) and for weakest liberal preexpectations this can be done analogously to weakest liberal precondition reasoning (see Section 2.2.5). Thus, we define continuation–passing style, backwards–moving expectation transformers as follows:

DEFINITION 4.5 (Expectation Transformers [Koz85; MM05]):
For $C \in$ pGCL we define the following expectation transformers:

A. *The weakest preexpectation transformer*

$$\mathsf{wp}\,[\![C]\!] \colon \quad \mathbb{E} \to \mathbb{E}$$

is defined according to the rules in Table 4.1.

B. *The weakest liberal preexpectation transformer*

$$\mathsf{wlp}\,[\![C]\!] \colon \quad \mathbb{E}_{\leq 1} \to \mathbb{E}_{\leq 1}$$

is defined according to the rules in Table 4.2.

C. *The angelic weakest preexpectation transformer*

$$\mathsf{awp}\,[\![C]\!] \colon \quad \mathbb{E} \to \mathbb{E}$$

is defined according to the rules obtained from Table 4.1 *by replacing every occurrence of* wp *by* awp *and the* min *by a* max.

D. *The angelic weakest liberal preexpectation transformer*

$$\mathsf{awlp}\,[\![C]\!] \colon \quad \mathbb{E}_{\leq 1} \to \mathbb{E}_{\leq 1}$$

is defined according to the rules obtained from Table 4.2 *by replacing every occurrence of* wlp *by* awlp *and the* min *by a* max.

C	$\mathsf{wp}[\![C]\!](f)$
`skip`	f
`diverge`	0
$x := E$	$f[x/E]$
$x :\approx \mu$	$\lambda\sigma.\ \int_{\mathsf{Vals}} \Big(\lambda v.\ f\big(\sigma[x \mapsto v]\big)\Big)\, d\,\mu_\sigma$
$C_1 \fatsemi C_2$	$\mathsf{wp}[\![C_1]\!]\big(\mathsf{wp}[\![C_2]\!](f)\big)$
$\texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\}$	$[\varphi]\cdot \mathsf{wp}[\![C_1]\!](f) \;+\; [\neg\varphi]\cdot \mathsf{wp}[\![C_2]\!](f)$
$\{C_1\} \,\square\, \{C_2\}$	$\min\Big\{\mathsf{wp}[\![C_1]\!](f), \mathsf{wp}[\![C_2]\!](f)\Big\}$
$\{C_1\}\,[p]\,\{C_2\}$	$p\cdot \mathsf{wp}[\![C_1]\!](f) \;+\; (1-p)\cdot \mathsf{wp}[\![C_2]\!](f)$
$\texttt{while}(\varphi)\{C'\}$	$\mathsf{lfp}\, X.\ [\neg\varphi]\cdot f \;+\; [\varphi]\cdot \mathsf{wp}[\![C']\!](X)$

Table 4.1: The weakest preexpectation transformer.

E. *For* wp, *we call the function*

$$ {}^{\mathsf{wp}}_{\langle\varphi,C\rangle}\Phi_f(X) \;=\; [\neg\varphi]\cdot f \;+\; [\varphi]\cdot \mathsf{wp}[\![C]\!](X) $$

the wp*–characteristic function of* $\texttt{while}(\varphi)\{C\}$ *with respect to postexpectation* f*. We define the* wlp*–,* awp*–, and* awlp*–characteristic functions* ${}^{\mathsf{wlp}}_{\langle\varphi,C\rangle}\Phi_f$, ${}^{\mathsf{awp}}_{\langle\varphi,C\rangle}\Phi_f$, *and* ${}^{\mathsf{awlp}}_{\langle\varphi,C\rangle}\Phi_f$ *analogously. If either of* wp, wlp, awp, awlp, φ, C, *or* f *are clear from the context, we omit them from* Φ*.*

Notice that by the Kleene Fixed Point Theorem (Theorem A.5) we have

$$ \mathsf{wp}[\![\texttt{while}(\varphi)\{C\}]\!](f) \;=\; \mathsf{lfp}\, {}^{\mathsf{wp}}_{\langle\varphi,C\rangle}\Phi_f \;=\; \sup_{n\in\mathbb{N}} {}^{\mathsf{wp}}_{\langle\varphi,C\rangle}\Phi^n_f(0)\,, $$

where ${}^{\mathsf{wp}}_{\langle\varphi,C\rangle}\Phi^n_f$ denotes n-fold application of ${}^{\mathsf{wp}}_{\langle\varphi,C\rangle}\Phi_f$ to its argument. The analogous statement holds for awp. For wlp, since this is defined via a greatest fixed point, we have a dual statement, namely

$$ \mathsf{wlp}[\![\texttt{while}(\varphi)\{C\}]\!](f) \;=\; \mathsf{gfp}\, {}^{\mathsf{wlp}}_{\langle\varphi,C\rangle}\Phi_f \;=\; \inf_{n\in\mathbb{N}} {}^{\mathsf{wlp}}_{\langle\varphi,C\rangle}\Phi^n_f(1)\,. $$

Analogous statements hold for the angelic expectation transformers.

An immediate corollary about preexpectations of loops which can be derived by a close inspection of the characteristic function is the following:

C	$\mathsf{wlp}\,[\![C]\!]\,(f)$
`skip`	f
`diverge`	1
$x := E$	$f\,[x/E]$
$x :\approx \mu$	$\lambda\sigma.\ \int_{\mathsf{Vals}} \Big(\lambda v.\ f\big(\sigma\,[x \mapsto v]\big)\Big)\, d\,\mu_\sigma$
$C_1 \fatsemi C_2$	$\mathsf{wlp}\,[\![C_1]\!]\,\big(\mathsf{wlp}\,[\![C_2]\!]\,(f)\big)$
$\texttt{if}\,(\varphi)\,\{C_1\}\,\texttt{else}\,\{C_2\}$	$[\varphi]\cdot \mathsf{wlp}\,[\![C_1]\!]\,(f) \;+\; [\neg\varphi]\cdot \mathsf{wlp}\,[\![C_2]\!]\,(f)$
$\{C_1\}\,\square\,\{C_2\}$	$\min\big\{\mathsf{wlp}\,[\![C_1]\!]\,(f),\, \mathsf{wlp}\,[\![C_2]\!]\,(f)\big\}$
$\{C_1\}\,[p]\,\{C_2\}$	$p\cdot \mathsf{wlp}\,[\![C_1]\!]\,(f) \;+\; (1-p)\cdot \mathsf{wlp}\,[\![C_2]\!]\,(f)$
$\texttt{while}(\varphi)\{C'\}$	$\mathsf{gfp}\ X.\ [\neg\varphi]\cdot f \;+\; [\varphi]\cdot \mathsf{wlp}\,[\![C']\!]\,(X)$

Table 4.2: The weakest liberal preexpectation transformer.

COROLLARY 4.6 (Postexpectation Strengthening for Loops): *Let $C \in$ pGCL and $\mathcal{T} \in \{\mathsf{wp}, \mathsf{wlp}, \mathsf{awp}, \mathsf{awlp}\}$. Then*

$$\mathcal{T}\,[\![\texttt{while}(\varphi)\{C\}]\!]\,(f) \;=\; \mathcal{T}\,[\![\texttt{while}(\varphi)\{C\}]\!]\,([\neg\varphi]\cdot f)\ ,$$

for an appropriate choice of $f \in \mathbb{E}$ or $f \in \mathbb{E}_{\leq 1}$ (depending on $\mathcal{T}$).

Proof. The $\mathcal{T}$–characteristic function of $\texttt{while}(\varphi)\{C\}$ with respect to post-expectation f is given by

$$\begin{aligned}&\lambda X.\ [\neg\varphi]\cdot f \;+\; [\varphi]\cdot \mathcal{T}\,[\![C]\!]\,(X)\\ &\;=\; \lambda X.\ [\neg\varphi]\cdot[\neg\varphi]\cdot f \;+\; [\varphi]\cdot \mathcal{T}\,[\![C]\!]\,(X)\ ,\end{aligned}$$

which is the $\mathcal{T}$–characteristic function with respect to $[\neg\varphi]\cdot f$. Since the characteristic functions coincide, so do their fixed points. Q.E.D.

Intuitively, the above corollary can be interpreted as the fact that a loop can only ever terminate in a state which satisfies the negation of the loop guard.

Notice that all rules for the anticipated value transformer in Table 2.3 are also found in Table 4.1. For the weakest preexpectation transformer, we have merely added rules for the probabilistic constructs. The conceptually easier one, namely the rule for probabilistic choice, reads

$$\mathsf{wp}\,[\![\{C_1\}\,[p]\,\{C_2\}]\!]\,(f) \;=\; p\cdot \mathsf{wp}\,[\![C_1]\!]\,(f) \;+\; (1-p)\cdot \mathsf{wp}\,[\![C_2]\!]\,(f)$$

The intuition behind this definition is straightforward: Since we cannot single out a value of f which is established by either C_1 or C_2, we simply average

these two values according to the probabilities with which C_1 and C_2 are executed, respectively, thus obtaining the *expected value* of f after executing C_1 with probability p and C_2 with probability $1-p$.

The rule for random assignments is technically more involved and reads

$$\mathsf{wp}\,[\![x :\approx \mu]\!]\,(f) \;=\; \lambda\sigma.\ \int_{\mathsf{Vals}} \Big(\lambda v.\ f\big(\sigma\,[x \mapsto v]\big)\Big)\, d\,\mu_\sigma\ .$$

The mechanics of the right–hand–side are as follows: Instead of averaging only over two options, we average now over updated versions of f according to a probability distribution. In more detail, the right–hand–side takes as input a state σ and averages the values of $f(\sigma\,[x \mapsto v])$ (i.e. f updated according to assignment $x := v$), where the values v are distributed according to probability distribution μ_σ.

Example 4.7 (Reasoning about Expected Values):
Reconsider the program

$$\{x := x+5\}\,[4/5]\,\{x := 10\}\ .$$

Suppose we want to know the expected value of x, i.e. the weakest preexpectation of the above program with respect to postexpectation x. We will reuse our annotation style from earlier (see Example 2.11), i.e.

$$\begin{array}{l} /\!/\!/\ g' \\ /\!/\!/\ g \\ C \\ /\!/\!/\ f \end{array}$$

expresses the fact that $g = \mathsf{wp}\,[\![C]\!]\,(f)$ and moreover that $g' = g$. We can then annotate the above program as shown in Figure 4.1 (read from bottom to top). When executing the above program on initial state σ, the expected value of x is hence $4\sigma(x)/5 + 6$.

Example 4.8 (Reasoning about Probabilities):
Reconsider the program from Example 4.7. Suppose we want to know the probability that x has value 10 after execution of that program. Then we can annotate this program as shown in Figure 4.2. When executing the above program on initial state σ with $\sigma(x) = 5$, then the probability that x equals 10 is $4/5 + 1/5 = 1$. Otherwise, it is $1/5$.

```
/// 4x/5 + 6
/// 4/5 · (x + 5) + 1/5 · 10
{
    /// x + 5
    x := x + 5
    /// x
} [4/5] {
    /// 10
    x := 10
    /// x
}
/// x
```

Figure 4.1: Weakest preexpectation annotations for Example 4.7.

```
/// 4/5 · [x = 5] + 1/5
/// 4/5 · [x = 5] + 1/5 · 1
{
    /// [x = 5]
    /// [x + 5 = 10]
    x := x + 5
    /// [x = 10]
} [4/5] {
    /// 1
    /// [true]
    /// [10 = 10]
    x := 10
    /// [x = 10]
}
/// [x = 10]
```

Figure 4.2: Weakest preexpectation annotations for Example 4.8.

EXAMPLE 4.9 (Weakest Liberal Preexpectations):
Reconsider the program

$$\{\texttt{diverge}\}\,[1/3]\,\{x := 10\}\,.$$

Suppose we want to know the probability that either the program terminates in a state where x has value 10 after execution of that program or the program diverges, i.e. the weakest *liberal* preexpectation of that program with respect to postexpectation $[x = 10]$. Then we can annotate this program as shown in Anticipated Values and Nontermination 4.9. The sought–after probability is thus 1.

EXAMPLE 4.10 (Weakest Liberal Preexpectations of While Loops):
Reconsider the program

$$\begin{array}{l} c := 1\,; \\ \texttt{while}(c = 1)\{ \\ \quad \{\texttt{diverge}\}\,[1/2]\,\{x := x+1\}\,; \\ \quad \{\texttt{skip}\}\,[1/2]\,\{c := 0\} \\ \}\,. \end{array}$$

Suppose we want to reason about the weakest liberal preexpectation of event „x is even“. The characteristic function of the loop with respect to postexpectation $[x \text{ even}]$ is given by

$$\begin{aligned} \Phi(X) \;=\; & [c \neq 1] \cdot [x \text{ even}] \\ & + \; [c = 1] \cdot \left(\frac{1}{2} + \frac{X[x/x+1] + X[c,x/0,x+1]}{4} \right). \end{aligned}$$

The first four iterations of the fixed point iteration for Φ are:

$$\begin{aligned} \Phi(1) \;&=\; [c \neq 1] \cdot [x \text{ even}] + [c = 1] \\ \Phi^2(1) \;&=\; [c \neq 1] \cdot [x \text{ even}] \;+\; [c = 1] \cdot \left(\frac{3}{4} + \frac{[x \text{ odd}]}{4} \right) \\ \Phi^3(1) \;&=\; [c \neq 1] \cdot [x \text{ even}] \;+\; [c = 1] \cdot \left(\frac{11}{16} + \frac{[x \text{ even}]}{16} + \frac{[x \text{ odd}]}{4} \right) \\ \Phi^4(1) \;&=\; [c \neq 1] \cdot [x \text{ even}] \\ &\qquad + \; [c = 1] \cdot \left(\frac{43}{64} + \frac{[x \text{ odd}]}{64} + \frac{[x \text{ even}]}{16} + \frac{[x \text{ odd}]}{4} \right) \end{aligned}$$

```
/// 1
/// 4/5 · 1 + 1/5 · 1
{
    /// 1
    diverge
    /// [x = 10]
} [4/5] {
    /// 1
    /// [true]
    /// [10 = 10]
    x := 10
    /// [x = 10]
}
/// [x = 10]
```

Figure 4.3: Weakest liberal preexpectation annotations for Example 4.9.

More detailed calculations are left as an exercise. After four iterations, we can slowly start seeing a somewhat complicated pattern for $n > 2$:

$$\Phi^n(1) \;=\; [c \neq 1]\cdot[x \text{ even}] \;+\; [c=1]\cdot\left(\frac{2^{n-1}+1}{4^{n-1}} + \sum_{i=0}^{\lfloor\frac{n-3}{2}\rfloor}\frac{[x \text{ even}]}{4^{2(i+1)}} + \sum_{i=0}^{\lfloor\frac{n-2}{2}\rfloor}\frac{[x \text{ odd}]}{4^{2i+1}}\right)$$

We omit proving the above pattern correct. By taking the limit (i.e. $n \to \omega$), we see that the sought–after weakest liberal preexpectation converges to

$$\begin{aligned}
&\mathsf{wp}\,[\![\texttt{while}(x>0)\{\ldots\}]\!]\,(z)\\
&= \mathsf{lfp}\,\Phi\\
&= \sup_{n\in\mathbb{N}} \Phi^n(0) \qquad \text{(by Theorem A.5)}\\
&= \sup_{n\in\mathbb{N}} [c\neq 1]\cdot[x \text{ even}] \;+\; [c=1]\cdot\left(\frac{2^{n-1}+1}{4^{n-1}} + \sum_{i=0}^{\lfloor\frac{n-2}{2}\rfloor}\frac{[x \text{ odd}]}{4^{2i+1}} + \sum_{i=0}^{\lfloor\frac{n-3}{2}\rfloor}\frac{[x \text{ even}]}{4^{2(i+1)}}\right)\\
&= [c\neq 1]\cdot[x \text{ even}] \;+\; [c=1]\cdot\left(\frac{2}{3} + \frac{4\cdot[x \text{ odd}]}{15} + \frac{[x \text{ even}]}{15}\right).
\end{aligned}$$

For the whole program, we can finally make these annotations:

$$\begin{aligned}
&/\!/\!/\ \tfrac{2}{3} + \tfrac{4\cdot[x \text{ odd}]}{15} + \tfrac{[x \text{ even}]}{15}\\
&c := 1\,;\\
&/\!/\!/\ [c\neq 1]\cdot[x \text{ even}] \;+\; [c=1]\cdot\left(\tfrac{2}{3} + \tfrac{4\cdot[x \text{ odd}]}{15} + \tfrac{[x \text{ even}]}{15}\right)\\
&\texttt{while}(c=1)\{\\
&\quad \{\texttt{diverge}\}\,[1/2]\,\{x := x+1\}\,;\\
&\quad \{\texttt{skip}\}\,[1/2]\,\{c := 0\}\\
&\}\\
&/\!/\!/\ [x \text{ even}]
\end{aligned}$$

We have thus proven

$$\mathsf{wlp}\,[\![\ldots]\!]\,([x \text{ even}]) \;=\; \frac{2}{3} + \frac{4\cdot[x \text{ odd}]}{15} + \frac{[x \text{ even}]}{15}.$$

This means that if we start the program in a state where x is odd, then there is a probability of $2/3 + 4/15 = 14/15$ that the program either not terminates or

terminates in a state where x is even. If we start the program in a state where x is even, this probability is $11/15$. Notice in particular that $14/15 + 11/15 = 5/3 > 1$.

4.1.4 *Connection to Operational Semantics*

Recall the characterizing equations of wp (Equation 4.2, p. 81), awp (Equation 4.3, p. 81), and wlp (Equation 4.4, p. 82). The next theorem states formally that those expectation transformers actually satisfy those equations:

Theorem 4.11 (Operational vs. Expectation Transformer Semantics): *Let $C \in \mathsf{pGCL}$, $f \in \mathbb{E}$, and $g \in \mathbb{E}_{\leq 1}$. Then:*

A. $$\mathsf{wp}\,[\![C]\!]\,(f) \;=\; \lambda\sigma.\ \inf_{\mathfrak{s}\in\mathsf{Scheds}} \int_{\Sigma} f\, d\,[\![C]\!]^{\mathfrak{s}}_{\sigma}$$

B. $$\mathsf{awp}\,[\![C]\!]\,(f) \;=\; \lambda\sigma.\ \sup_{\mathfrak{s}\in\mathsf{Scheds}} \int_{\Sigma} f\, d\,[\![C]\!]^{\mathfrak{s}}_{\sigma}$$

C. $$\mathsf{wlp}\,[\![C]\!]\,(f) \;=\; \lambda\sigma.\ \inf_{\mathfrak{s}\in\mathsf{Scheds}} \int_{\Sigma} f\, d\,[\![C]\!]^{\mathfrak{s}}_{\sigma} \;+\; \left(1 - \int_{\Sigma} 1\, d\,[\![C]\!]^{\mathfrak{s}}_{\sigma}\right)$$

D. $$\mathsf{awlp}\,[\![C]\!]\,(f) \;=\; \lambda\sigma.\ \sup_{\mathfrak{s}\in\mathsf{Scheds}} \int_{\Sigma} f\, d\,[\![C]\!]^{\mathfrak{s}}_{\sigma} \;+\; \left(1 - \int_{\Sigma} 1\, d\,[\![C]\!]^{\mathfrak{s}}_{\sigma}\right)$$

Proof. By induction on the structure of C.

Theorem 4.11 establishes a connection between the expectation transformers and the probability distribution over final states obtained by the operational semantics $[\![C]\!]^{\mathfrak{s}}_{\sigma}$ under some scheduler (see Definition 3.8). A connection between the distribution transformer semantics of Kozen [Koz79; Koz81] and a Markov process semantics was shown earlier by Sharir, Pnueli, and Hart [SPH84]. A formal connection between Markov chains and weakest preexpectations with an emphasis on invariants was studied more recently by Gretz, Katoen, and McIver [GKM12; GKM14].

4.2 HEALTHINESS CONDITIONS

Expectation transformers enjoy several properties like continuity, monotonicity, etc. In the literature, some of these properties are called *healthiness conditions* [MM05; Kei15; Hin+16] or *homomorphism properties* [BW98]. Informally speaking, healthiness conditions are a collection of properties that characterize those backward–moving predicate (or expectation) transformers that are the dual of a forward–moving state (or distribution) transformer that arises from an actual (probabilistic) program.

Many of the properties we present here will be used in our proofs. In their own right, they aid in concrete reasoning about probabilistic programs, for instance by forming a foundation for compositional reasoning.

4.2.1 *Continuity*

Continuity is perhaps one of the most fundamental properties that expectation transformers enjoy because it ensures for instance well–definedness of expectation transformer semantics of while loops. An expectation transformer $\mathcal{T} : \mathbb{E} \to \mathbb{E}$ is continuous iff for any chain of expectations $S = \{s_0 \preceq s_1 \preceq s_2 \preceq \ldots\} \subseteq \mathbb{E}$ we have

$$\mathcal{T}(\sup S) \;=\; \sup\, \mathcal{T}(S)\,;$$

see Definition A.2 for more details. All expectation transformers we have presented in this thesis are continuous:

> **Theorem 4.12 (Continuity of Expectation Transformers):**
> *Let C be a* pGCL *program. Then the associated expectation transformers* wp $[\![C]\!]$, wlp $[\![C]\!]$, awp $[\![C]\!]$, *and* awlp $[\![C]\!]$ *are continuous.*

Proof. By induction on the structure of C. Q.E.D.

The importance of continuity for well–defined semantics of loops can be sketched as follows: For any loop–free program C, continuity of wp $[\![C]\!]$ ensures that the characteristic function of the loop $\texttt{while}(\varphi)\{C\}$ (that has C as its loop body) is also continuous. This ensures by the Kleene fixed point theorem (Theorem A.5) that the characteristic function has a least fixed point, which in turn ensures that wp $[\![\texttt{while}(\varphi)\{C\}]\!]$ is well–defined. The fact that the transformer wp $[\![\texttt{while}(\varphi)\{C\}]\!]$ itself is also continuous ensures well–defined expectation transformer semantics of nested loops.

4.2.2 *Strictness*

Strictness is a healthiness condition that Dijkstra calls „Law of the Excluded Miracle“ [Dij75]. For his weakest precondition calculus, this law states that there exists no initial state from which the execution of a program C can terminate in a state satisfying the predicate false. In terms of wp, this law reads

$$\text{wp}\,[\![C]\!]\,(\text{false}) \;=\; \text{false}\,.$$

For weakest liberal preconditions a dual law (that we call *costrictness*) would state that for all initial states the execution of a program C either terminates (in some state state satisfying true) or does not terminate. In terms of wlp,

this can be expressed as

$$\mathsf{wlp}\,[\![C]\!]\,(\mathsf{true}) \;=\; \mathsf{true}\,.$$

In our quantitative setting, strictness and costrictness are defined as follows:

DEFINITION 4.13 (Strictness and Costrictness):
Let $C \in \mathsf{pGCL}$ and let $\mathcal{T} : \mathbb{E} \to \mathbb{E}$ be an expectation transformer. Then:

A. *$\mathcal{T}$ is called strict, iff*

$$\mathcal{T}(0) \;=\; 0\,.$$

B. *$\mathcal{T}$ is called costrict, iff*

$$\mathcal{T}(1) \;=\; 1\,.$$

Analogously to Dijkstra's predicate transformers, liberal expectation transformers are costrict and their nonliberal versions are strict:

THEOREM 4.14 (Strictness of Expectation Transformers):
Let $C \in \mathsf{pGCL}$. Then:

A. *$\mathsf{wp}\,[\![C]\!]$ and $\mathsf{awp}\,[\![C]\!]$ are strict.*

B. *$\mathsf{wlp}\,[\![C]\!]$ and $\mathsf{awlp}\,[\![C]\!]$ are costrict.*[4]

Proof. Strictness of wp and awp follows from feasibility of wp and awp, respectively; see Section 4.2.3. Q.E.D.

The quantitative version of strictness tells us that the expected value of the constantly 0 random variable after executing a program C is 0. Alternatively stated: the probability that C terminates in a state satisfying false is 0. Costrictness tells us that the probability to either terminate or not is 1.

4.2.3 *Feasibility*

The property that McIver & Morgan call *feasibility* states that preexpectations cannot become „too large" [MM05]. The notion of feasibility makes sense for bounded expectations $f \in \mathbb{E}_{\leq \exists b}$ only. Formally, it is stated as follows:

THEOREM 4.15 (Feasibility of Expectation Transformers[5]):
Let $C \in \mathsf{pGCL}$. Moreover, let $f \in \mathbb{E}_{\leq \exists b}$ be an expectation bounded by $b \in \mathbb{R}_{\geq 0}$, i.e. $f \preceq b$. Then

$$\mathsf{wp}\,[\![C]\!]\,(f) \;\preceq\; b \quad \text{and} \quad \mathsf{awp}\,[\![C]\!]\,(f) \;\preceq\; b\,.$$

4 For costrictness of wlp, see [MM05, Fact B.3.4 on p. 331].
5 See [MM05, Lemma 5.6.4 *and* p. 228].

Feasibility of wp implies its strictness and can thus be seen as a quantitative generalization of strictness. To see that feasibility implies strictness observe that 0 is a 0–bounded expectation and feasibility of wp states that

$$\mathsf{wp}\,[\![C]\!]\,(0) \;\preceq\; 0\,,$$

which implies $\mathsf{wp}\,[\![C]\!]\,(0) = 0$ by expectations being non–negative. An analogous argument applies to awp.

4.2.4 *Monotonicity*

The distinct feature of everything extant is its monotony.

— Vladimir Nabokov

Monotonicity is another fundamental property of expectation transformers. According to Back and von Wright, monotonicity is „the only healthiness criteria [sic] that has gone unquestioned“ [BW89]. An expectation transformer $\mathcal{T}$ is monotonic iff for any two expectations $f, g \in \mathbb{E}$, we have that

$$f \preceq g \quad \text{implies} \quad \mathcal{T}(f) \preceq \mathcal{T}(g)\,;$$

see Definition A.3 for more details. All expectation transformers we have presented so far are monotonic:

Theorem 4.16 (Monotonicity of Expectation Transformers):
Let $C \in$ pGCL. Then the associated expectation transformers $\mathsf{wp}\,[\![C]\!]$, $\mathsf{wlp}\,[\![C]\!]$, $\mathsf{awp}\,[\![C]\!]$, *and* $\mathsf{awlp}\,[\![C]\!]$ *are monotonic. Furthermore, all* wp–, wlp–, awp–, *and* awlp–*characteristic functions are monotonic.*

Proof. Every continuous function is monotonic, see Theorem A.4. Q.E.D.

Monotonicity is not just a healthiness condition but plays an important role in reasoning about programs. In the following we present two implications of the monotonicity property.

Compositionality. Monotonicity is useful for compositional reasoning in the following sense: Imagine two programs C_1 and C_2 and a postexpectation f such that

$$\mathsf{wp}\,[\![C_1]\!]\,(f) \;\preceq\; \mathsf{wp}\,[\![C_2]\!]\,(f)\,.$$

Then monotonicity ensures that if we put the components C_1 and C_2 into some context $C\,\fatsemi\,\ldots$, then we can be certain that

$$\mathsf{wp}\,[\![C \fatsemi C_1]\!]\,(f) \;\preceq\; \mathsf{wp}\,[\![C \fatsemi C_2]\!]\,(f)\,,$$

since $\mathsf{wp}\,[\![C \fatsemi C_i]\!]\,(f) = \mathsf{wp}\,[\![C]\!]\,\big(\mathsf{wp}\,[\![C_i]\!]\,(f)\big)$, for $i \in \{1, 2\}$.

Relation to the consequence rule. Monotonicity is closely related to the *consequence rule* of Hoare logic. This rule reads

$$\frac{G \implies G' \qquad \langle G' \rangle\, C\, \langle F' \rangle \qquad F' \implies F}{\langle G \rangle\, C\, \langle F \rangle}\ \text{(cons)}\,.$$

It weakens precondition G to G' and strengthens postcondition F to F' in order to prove validity of $\langle G \rangle\, C\, \langle F \rangle$ by proving validity of $\langle G' \rangle\, C\, \langle F' \rangle$.

The analogy to weakest preexpectation reasoning is as follows: In order to prove $g \preceq \mathsf{wp}\,[\![C]\!]\,(f)$ it suffices to

1. choose $g' \succeq g$,
2. choose $f' \preceq f$, and
3. prove $g' \preceq \mathsf{wp}\,[\![C]\!]\,(f')$,

since this gives

$$\begin{aligned} g &\preceq g' && \text{(by 1. above)} \\ &\preceq \mathsf{wp}\,[\![C]\!]\,(f') && \text{(by 3. above)} \\ &\preceq \mathsf{wp}\,[\![C]\!]\,(f)\,, && \text{(by 2. above and monotonicity, Theorem 4.16)} \end{aligned}$$

which in turn implies $g \preceq \mathsf{wp}\,[\![C]\!]\,(f)$. The *"consequence rule of weakest precondition reasoning"* thus reads

$$\frac{g \preceq g' \qquad g' \preceq \mathsf{wp}\,[\![C]\!]\,(f') \qquad f' \preceq f}{g \preceq \mathsf{wp}\,[\![C]\!]\,(f)}\ \text{(wp–cons)}\,.$$

4.2.5 *Linearity*

Linearity of expectation transformers plays a prominent role in the development of McIver & Morgan as they show that superlinearity[6] alone already characterizes their wp and thus implies monotonicity, strictness, continuity, and so on. However, as mentioned before, McIver & Morgan heavily rely on the fact that their expectations are bounded (see Definition 4.1 B.). For showing that superlinearity implies continuity they even need to restrict to a finite state space Σ [MM05, p. 148].

While this allows McIver & Morgan to nicely characterize all „healthy" expectation transformers by means of just a single healthiness condition, we cannot make the restriction of boundedness and do not wish to restrict to a finite state space. Another point is that we make use of the angelic awp

6 Note that McIver & Morgan use the term *sublinear* for *superlinear* transformers [MM05]. However, the super–/sub–nomenclature we use here is more in accordance with the standard mathematics terminology [Wikk; KM17].

transformer which is sub- instead of superlinear and while we saw that awp is monotonic, this fact does not follow from sublinearity.

Since our setting differs from that of McIver & Morgan, we will conduct our own linearity studies here. Linearity is made up of two properties: homogeneity and additivity. We study the former first:

Definition 4.17 (Positive Homogeneity):
Let $f \in \mathbb{E}$, $r \in \mathbb{R}_{\geq 0}$, and let $\mathcal{T} : \mathbb{E} \to \mathbb{E}$ be an expectation transformer. Then $\mathcal{T}$ is called positively homogeneous[7] iff

$$\mathcal{T}(r \cdot f) = r \cdot \mathcal{T}(f) .$$

We consider *positive* homogeneity (i.e. our scaling factor r is positive) instead of general homogeneity since we need to stay within the realm of expectations which are non–negative. Both nonliberal expectation transformers we have presented are positively homogeneous:

Theorem 4.18 (Positive Homogeneity of Expectation Transformers):
For any $C \in \mathsf{pGCL}$, $\mathsf{wp}\,[\![C]\!]$ and $\mathsf{awp}\,[\![C]\!]$ are positively homogeneous.

Proof. By induction on the structure of C.

Positive homogeneity of wp implies its strictness by the following reasoning:

$$\begin{aligned} \mathsf{wp}\,[\![C]\!]\,(0) &= \mathsf{wp}\,[\![C]\!]\,(2 \cdot 0) \\ &= 2 \cdot \mathsf{wp}\,[\![C]\!]\,(0) \qquad \text{(by positive homogeneity, Theorem 4.18)} \end{aligned}$$

which implies that $\mathsf{wp}\,[\![C]\!]\,(0) = 0$. The proof for awp is analogous.

Positive homogeneity and monotonicity together also suffice to prove for expectation transformers Markov's well–known inequality:

Theorem 4.19 (Markov's Inequality):
Let $C \in \mathsf{pGCL}$, $f \in \mathbb{E}$, and $a \in \mathbb{R}_{\geq 0}$ with $a > 0$. Then:

A. $\mathsf{wp}\,[\![C]\!]\,([f \geq a]) \preceq \dfrac{\mathsf{wp}\,[\![C]\!]\,(f)}{a}$

B. $\mathsf{awp}\,[\![C]\!]\,([f \geq a]) \preceq \dfrac{\mathsf{awp}\,[\![C]\!]\,(f)}{a}$

7 This property is called "scaling" by McIver and Morgan [MM05].

Proof. For A., consider the following:

$$
\begin{aligned}
& a \cdot [f \geq a] \preceq f \\
\text{implies} \quad & \mathsf{wp}\,[\![C]\!]\,(a \cdot [f \geq a]) \preceq \mathsf{wp}\,[\![C]\!]\,(f) \\
& \qquad\qquad \text{(by monotonicity, Theorem 4.16)} \\
\text{implies} \quad & a \cdot \mathsf{wp}\,[\![C]\!]\,([f \geq a]) \preceq \mathsf{wp}\,[\![C]\!]\,(f) \\
& \qquad\qquad \text{(by positive homogeneity, Theorem 4.18)} \\
\text{iff} \quad & \mathsf{wp}\,[\![C]\!]\,([f \geq a]) \preceq \frac{\mathsf{wp}\,[\![C]\!]\,(f)}{a} \qquad \text{(by } a > 0\text{)}
\end{aligned}
$$

The proof for awp is analogous. Q.E.D.

wlp and awlp are *not* positively homogenous as the following example shows:

$$\mathsf{wlp}\,[\![\texttt{diverge}]\!]\left(\tfrac{1}{2} \cdot 0\right) = 1 > \tfrac{1}{2} = \tfrac{1}{2} \cdot \mathsf{wlp}\,[\![\texttt{diverge}]\!]\,(0)$$

Neither wlp nor awlp satisfy Markov's inequality.

In order to study linearity, we now study additivity of our expectation transformers. Together with homogeneity this yields the notion of linearity.

DEFINITION 4.20 (Linearity of Expectation Transformers):
Let $f, g \in \mathbb{E}$, $r \in \mathbb{R}_{\geq 0}$, and $\mathcal{T} \colon \mathbb{E} \to \mathbb{E}$ be an expectation transformer. Then:

A. *$\mathcal{T}$ is called sublinear[8] iff*

$$\mathcal{T}(r \cdot f + g) \preceq r \cdot \mathcal{T}(f) + \mathcal{T}(g)\,.$$

B. *$\mathcal{T}$ is called superlinear iff*

$$r \cdot \mathcal{T}(f) + \mathcal{T}(g) \preceq \mathcal{T}(r \cdot f + g)\,.$$

C. *$\mathcal{T}$ is called linear iff*

$$\mathcal{T}(r \cdot f + g) = r \cdot \mathcal{T}(f) + \mathcal{T}(g)\,.$$

All our expectation transformers satisfy one of the above notions of linearity:

THEOREM 4.21 (Linearity of Expectation Transformers[9]):
Let $C \in$ pGCL. Then:

A. *$\mathsf{wp}\,[\![C]\!]$ is superlinear.*[10]

B. *$\mathsf{awp}\,[\![C]\!]$ is sublinear.*

8 Recall Footnote 6 on page 95.
9 See [Koz83; MM05].
10 Recall Footnote 6 on page 95.

Suppose moreover that C is tame. Then angelic and demonic expectation transformers coincide, and

C. $\mathsf{wp}\,[\![C]\!]$ *and* $\mathsf{awp}\,[\![C]\!]$ *are linear.*

D. $\mathsf{wlp}\,[\![C]\!]$ *and* $\mathsf{awlp}\,[\![C]\!]$ *are superlinear.*

Proof. As for A. and B., the proof is by induction on the structure of C.

As for C., linearity of wlp follows from the fact that wp and awp obviously coincide on tame programs. But since wp is superlinear and awp is sublinear, wp and awp have to be linear.

As for D., superlinearity of wlp follows from the connection of wlp and wp (see Corollary 4.26) as discussed in Section 4.3. Q.E.D.

Superlinearity of wp implies monotonicity of wp by the following reasoning: Let f and g be two expectations such that $f \preceq g$. Then there exists an expectation $\epsilon \in \mathbb{E}$ such that $g = f + \epsilon$. By superlinearity of wp we then have

$$\begin{aligned} \mathsf{wp}\,[\![C]\!]\,(f) &\preceq \mathsf{wp}\,[\![C]\!]\,(f) + \mathsf{wp}\,[\![C]\!]\,(\epsilon) \\ &\preceq \mathsf{wp}\,[\![C]\!]\,(f+\epsilon) \qquad \text{(by superlinearity, Theorem 4.21 A.)} \\ &= \mathsf{wp}\,[\![C]\!]\,(g) \end{aligned}$$

The above reasoning fails for awp as it is sublinear instead of superlinear and the inequality is hence in the wrong direction. Still, awp is monotonic.

A useful corollary for expectation *subtraction* is the following:

Corollary 4.22 (Linearity and Subtractions):
Let $C \in \mathsf{pGCL}$ and let $f, g \in \mathbb{E}$ such that $f \preceq g$, thus $g - f$ is a well–defined (non–negative) expectation. Then:

A. $\mathsf{wp}\,[\![C]\!]\,(g-f) \preceq \mathsf{wp}\,[\![C]\!]\,(g) - \mathsf{wp}\,[\![C]\!]\,(f)$.

B. $\mathsf{awp}\,[\![C]\!]\,(g-f) \succeq \mathsf{awp}\,[\![C]\!]\,(g) - \mathsf{awp}\,[\![C]\!]\,(f)$.

Suppose moreover that C is tame. Then:

C. $\mathsf{wp}\,[\![C]\!] = \mathsf{awp}\,[\![C]\!]$ *and* $\mathsf{wp}\,[\![C]\!]\,(g-f) = \mathsf{wp}\,[\![C]\!]\,(g) - \mathsf{wp}\,[\![C]\!]\,(f)$.

Proof. For wp consider the following:

$$\begin{aligned} \mathsf{wp}\,[\![C]\!]\,(g-f) &= \mathsf{wp}\,[\![C]\!]\,(g-f) + \mathsf{wp}\,[\![C]\!]\,(f) - \mathsf{wp}\,[\![C]\!]\,(f) \\ &\preceq \mathsf{wp}\,[\![C]\!]\,(g-f+f) - \mathsf{wp}\,[\![C]\!]\,(f) \\ &\qquad \text{(by superlinearity, Theorem 4.21 A.)} \\ &= \mathsf{wp}\,[\![C]\!]\,(g) - \mathsf{wp}\,[\![C]\!]\,(f) \end{aligned}$$

The reasoning for awp and for the case of tame programs is analogous. Q.E.D.

Thus, we see that for subtractions wp behaves sublinearly instead of superlinearly whereas awp behaves superlinearly instead of sublinearly.

We can make use of Corollary 4.22 to show that — in addition to monotonicity — superlinearity of wp also implies feasibility of wp: For showing this, let $f \in \mathbb{E}_{\leq \exists b}$ be an expectation bounded by $b \in \mathbb{R}_{\geq 0}$. Then $f \preceq b$ and

$$0 \preceq \mathsf{wp}[\![C]\!](b-f) \preceq \mathsf{wp}[\![C]\!](b) - \mathsf{wp}[\![C]\!](f) \qquad \text{(by Corollary 4.22 A.)}$$

$$\text{implies} \quad 0 \preceq \mathsf{wp}[\![C]\!](b) - \mathsf{wp}[\![C]\!](f)$$

$$\text{implies} \quad \mathsf{wp}[\![C]\!](f) \preceq \mathsf{wp}[\![C]\!](b)$$

Again, the above reasoning fails for awp as awp is sublinear instead of superlinear and the inequality is therefore in the wrong direction. Nevertheless, awp is feasible.

4.3 RELATING EXPECTATION TRANSFORMERS

The definitions of the different expectation transformers we studied in this chapter are quite similar and it is not surprising that the transformers are closely related. The most elementary and most obvious relationship between angelic (awp and awlp) and demonic preexpectations (wp and wlp) is that demonic preexpectations are never greater than angelic preexpectations:

Corollary 4.23 (Angelic Bound Demonic Preexpectations): *Let $C \in \mathsf{pGCL}$, $f \in \mathbb{E}$, and $g \in \mathbb{E}_{\leq 1}$. Then*

A. $\mathsf{wp}[\![C]\!](f) \preceq \mathsf{awp}[\![C]\!](f)$, *and*

B. $\mathsf{wlp}[\![C]\!](g) \preceq \mathsf{awlp}[\![C]\!](g)$.

The most elementary relationship between liberal (wlp and awlp) and nonliberal preexpectations (wp and awp) is that weakest preexpectations are never greater than weakest liberal preexpectations:

Corollary 4.24 (Liberal Bound Nonliberal Preexpectations): *Let $C \in \mathsf{pGCL}$ and $f \in \mathbb{E}_{\leq 1}$. Then*

A. $\mathsf{wp}[\![C]\!](f) \preceq \mathsf{wlp}[\![C]\!](f)$, *and*

B. $\mathsf{awp}[\![C]\!](f) \preceq \mathsf{awlp}[\![C]\!](f)$.

Proof. Follows immediately from the fact that nonliberal preexpectations are defined as a *least* fixed point whereas liberal preexpectations are defined as a *greatest* fixed point. Q.E.D.

Intuitively, we can understand Corollary 4.24 as the fact that a program is more likely to be partially correct as it is likely to be totally correct.

In addition to the above, we can make a more precise statement relating liberal and nonliberal preexpectations. In there, some care regarding nondeterminism must be taken — liberal transformers become nonliberal and angelic ones become demonic; and vice versa:

THEOREM 4.25 (Relationship between Expectation Transformers): *Let $C \in \mathsf{pGCL}$ be tame and let $f \in \mathbb{E}_{\leq 1}$. Then*

A. $\quad \mathsf{wp}\,[\![C]\!]\,(f) \;=\; 1 - \mathsf{awlp}\,[\![C]\!]\,(1-f)$

B. $\quad \mathsf{awp}\,[\![C]\!]\,(f) \;=\; 1 - \;\mathsf{wlp}\,[\![C]\!]\,(1-f)$

C. $\quad \mathsf{wlp}\,[\![C]\!]\,(f) \;=\; 1 - \;\mathsf{awp}\,[\![C]\!]\,(1-f)$

D. $\quad \mathsf{awlp}\,[\![C]\!]\,(f) \;=\; 1 - \;\mathsf{wp}\,[\![C]\!]\,(1-f)$

Proof. For proving A., consider the following:

$$
\begin{aligned}
&1 - \mathsf{awlp}\,[\![C]\!]\,(1-f) \\
&= 1 - \lambda\sigma.\ \sup_{\mathfrak{s}\in\mathsf{Scheds}} \int_\Sigma 1 - f \; d\,[\![C]\!]^{\mathfrak{s}}_\sigma \;+\; 1 - \int_\Sigma 1 \; d\,[\![C]\!]^{\mathfrak{s}}_\sigma \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by Theorem 4.11 D.)} \\
&= 1 - \lambda\sigma.\ \sup_{\mathfrak{s}\in\mathsf{Scheds}} \int_\Sigma 1 \; d\,[\![C]\!]^{\mathfrak{s}}_\sigma - \int_\Sigma f \; d\,[\![C]\!]^{\mathfrak{s}}_\sigma \;+\; 1 - \int_\Sigma 1 \; d\,[\![C]\!]^{\mathfrak{s}}_\sigma \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Big(\text{by linearity of } \textstyle\int\Big) \\
&= 0 - \lambda\sigma.\ \sup_{\mathfrak{s}\in\mathsf{Scheds}} - \int_\Sigma f \; d\,[\![C]\!]^{\mathfrak{s}}_\sigma \\
&= \lambda\sigma.\ \inf_{\mathfrak{s}\in\mathsf{Scheds}} \int_\Sigma f \; d\,[\![C]\!]^{\mathfrak{s}}_\sigma \\
&= \mathsf{wp}\,[\![C]\!]\,(f) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by Theorem 4.11 A.)}
\end{aligned}
$$

The proofs for B., C., and D. are analogous. **Q.E.D.**

Let us gain some intuition on the above by considering Theorem 4.25 A. and choosing as postexpectation a predicate $[\varphi]$. Then we get

$$\mathsf{wp}\,[\![C]\!]\,([\varphi]) \;=\; 1 - \mathsf{awlp}\,[\![C]\!]\,(1-[\varphi]) \;=\; 1 - \mathsf{awlp}\,[\![C]\!]\,([\neg\varphi])\ .$$

$\mathsf{wp}\,[\![C]\!]\,([\varphi])$ tries to minimize the probability of C terminating in a state satisfying φ. How can wp achieve that? It can either drive C towards diverging or terminating but satisfying $\neg\varphi$, i.e. the opposite of φ. So wp will maximize the probability of either of these two events. But this is precisely what $\mathsf{awlp}\,[\![C]\!]\,([\neg\varphi])$ does.

For tame programs, the angelic and demonic transformers coincide and the nonliberal transformers are linear. We hence get from linearity (Theorem 4.21 c.) and Theorem 4.25 c. the following corollary:

COROLLARY 4.26 ([Koz83]):
Let $C \in \mathsf{pGCL}$ be tame and let $f \in \mathbb{E}_{\leq 1}$. Then

$$\mathsf{wlp}\,[\![C]\!]\,(f) \;=\; \mathsf{wp}\,[\![C]\!]\,(f) + 1 - \mathsf{wp}\,[\![C]\!]\,(1)\ .$$

Thus, $\mathsf{wlp}\,[\![C]\!]\,(f)$ is expressible as the sum of $\mathsf{wp}\,[\![C]\!]\,(f)$ and the probability that C does *not* terminate, i.e. $1 - \mathsf{wp}\,[\![C]\!]\,(1)$. This connection between wp and wlp allows us to study linearity of wlp: We have

$$\begin{aligned}
&\mathsf{wlp}\,[\![C]\!]\,(r \cdot f + g) \\
&= 1 - \mathsf{wp}\,[\![C]\!]\,(1) + \mathsf{wp}\,[\![C]\!]\,(r \cdot f + g) && \text{(by Corollary 4.26)} \\
&= 1 - \mathsf{wp}\,[\![C]\!]\,(1) + r \cdot \mathsf{wp}\,[\![C]\!]\,(f) + \mathsf{wp}\,[\![C]\!]\,(g) \\
& && \text{(by linearity, Theorem 4.21 c.)} \\
&\geq\ r \cdot \mathsf{wp}\,[\![C]\!]\,(f) + \mathsf{wp}\,[\![C]\!]\,(g)
\end{aligned}$$

for any tame $C \in \mathsf{pGCL}$, $f, g \in \mathbb{E}_{\leq 1}$, and $r \in \mathbb{R}_{\geq 0}$, such that $r \cdot f + g \in \mathbb{E}_{\leq 1}$. Thus, wlp is in general not linear for tame programs but superlinear instead.

An even closer connection between $\mathsf{wp}\,[\![C]\!]$ and $\mathsf{wlp}\,[\![C]\!]$ can be established in case that C terminates almost–surely. We first note that

$$\mathsf{wp}\,[\![C]\!]\,(1)$$

is an expectation, such that $\mathsf{wp}\,[\![C]\!]\,(1)(\sigma)$ gives the *(minimal) probability that C terminates on input σ*. Dually, $\mathsf{awp}\,[\![C]\!]\,(1)(\sigma)$ gives the maximal probability of C terminating on σ.

Yet dually,

$$\mathsf{awlp}\,[\![C]\!]\,(0)$$

is an expectation such that $\mathsf{awlp}\,[\![C]\!]\,(0)(\sigma)$ gives the maximal probability that C diverges on σ, whereas $\mathsf{wlp}\,[\![C]\!]\,(0)(\sigma)$ gives the minimal probability of C terminating on σ.

If we know a predicate T, such that C terminates almost–surely (i.e. with probability 1) from every initial state satisfying T, we can express this by

$$[T] \;\preceq\; \mathsf{wp}\,[\![C]\!]\,(1)\ .$$

In fact, if we know such a predicate T, we can „sandwich“ weakest preexpectations by means of weakest liberal preexpectations:

THEOREM 4.27 ((Non)liberal Preexpectations under Termination):
Let $C \in \mathsf{pGCL}$, let $f \in \mathbb{E}_{\leq 1}$, let T be a predicate, and let C terminate from any state satisfying T, i.e. let $[T] \preceq \mathsf{wp}\,[\![C]\!]\,(1)$. Then

A. $[T] \cdot \mathsf{wlp}\,[\![C]\!]\,(f) \;\preceq\; \mathsf{wp}\,[\![C]\!]\,(f) \;\preceq\; \mathsf{wlp}\,[\![C]\!]\,(f)$, *and*

B. $[T] \cdot \mathsf{awlp}\,[\![C]\!]\,(f) \;\preceq\; \mathsf{awp}\,[\![C]\!]\,(f) \;\preceq\; \mathsf{awlp}\,[\![C]\!]\,(f)$

Proof. For A., consider the following: Let $f \ominus g$ be defined as $\max\{f-g, 0\}$ and consider the following:

$$\Big(\mathsf{wlp}\,[\![C]\!]\,(f) + \mathsf{wp}\,[\![C]\!]\,(1)\Big) \ominus 1 \;\preceq\; \mathsf{wp}\,[\![C]\!]\,((f+1)\ominus 1)$$
(by [MM05, Fact B.3.2 on p.331])

$$\text{implies}\quad \Big([T]\cdot\mathsf{wlp}\,[\![C]\!]\,(f) + [T]\cdot\mathsf{wp}\,[\![C]\!]\,(1)\Big) \ominus [T] \;\preceq\; \mathsf{wp}\,[\![C]\!]\,((f+1)\ominus 1)$$
(Multiply left–hand side by $[T]$)

$$\text{implies}\quad \Big([T]\cdot\mathsf{wlp}\,[\![C]\!]\,(f) + [T]\cdot[T]\Big) \ominus [T] \;\preceq\; \mathsf{wp}\,[\![C]\!]\,((f+1)\ominus 1)$$
(by assumption $[T] \preceq \mathsf{wp}\,[\![C]\!]\,(1)$)

$$\text{iff}\quad \Big([T]\cdot\mathsf{wlp}\,[\![C]\!]\,(f) + [T]\Big) \ominus [T] \;\preceq\; \mathsf{wp}\,[\![C]\!]\,((f+1)\ominus 1)$$

$$\text{iff}\quad [T]\cdot\mathsf{wlp}\,[\![C]\!]\,(f) \;\preceq\; \mathsf{wp}\,[\![C]\!]\,(f)$$

For B., we exploit Theorem 4.27 A. and Theorem 4.25 A. and B. as follows:

$$\begin{aligned}
&\mathsf{awp}\,[\![C]\!]\,(f)\\
&= 1 - \mathsf{wlp}\,[\![C]\!]\,(1-f) && \text{(by Theorem 4.25 B.)}\\
&\geq [T] - [T]\cdot\mathsf{wlp}\,[\![C]\!]\,(1-f)\\
&\geq [T]\cdot\Big(1 - [T]\cdot\mathsf{wlp}\,[\![C]\!]\,(1-f)\Big)\\
&\geq [T]\cdot\Big(1 - \mathsf{wp}\,[\![C]\!]\,(1-f)\Big) && \text{(by assumption and Theorem 4.27 A.)}\\
&= [T]\cdot\Big(1 - \big(1 - \mathsf{awlp}\,[\![C]\!]\,(1-(1-f))\big)\Big) && \text{(by Theorem 4.25 A.)}\\
&= [T]\cdot\mathsf{awlp}\,[\![C]\!]\,(f) && \text{Q.E.D.}
\end{aligned}$$

In the case of universally almost–surely terminating programs (i.e. programs that terminate almost–surely on *every* input), we even get that liberal and nonliberal expectation transformers coincide entirely, which implies that (on $\mathbb{E}_{\leq 1}$) least and greatest fixed points coincide:

COROLLARY 4.28:
Let $C \in \mathsf{pGCL}$, let $f \in \mathbb{E}_{\leq 1}$, and let C terminate universally almost–surely, i.e. let $\mathsf{wp}\,[\![C]\!]\,(1) = 1$. Then

A. $\mathsf{wlp}\,[\![C]\!]\,(f) \;=\; \mathsf{wp}\,[\![C]\!]\,(f)$, *and*

B. $\mathsf{awlp}\,[\![C]\!]\,(f) \;=\; \mathsf{awp}\,[\![C]\!]\,(f)$

Proof. For A., we have by Corollary 4.24 that $\mathsf{wp}\,[\![C]\!]\,(f) \preceq \mathsf{wlp}\,[\![C]\!]\,(f)$. It is left to show that $\mathsf{wlp}\,[\![C]\!]\,(f) \preceq \mathsf{wp}\,[\![C]\!]\,(f)$ holds which is an immediate

consequence from Theorem 4.27 A. by choosing $T = \text{true}$. The proof for the angelic transformers is analogous. Q.E.D.

We will see later in Chapter 5 that Theorem 4.27 and Corollary 4.28 have important consequences for reasoning about loops, as they makes reasoning about lower bounds of expected values considerably easier. Reasoning about the precondition of Corollary 4.28 however, namely almost–sure termination, is often difficult, as we will discuss in Chapter 6.