Let $\boldsymbol{P}$ be the set of propositions.

Definition 1. *$E \cup \{\lightning\}$ is the set of atomic propositions hereby called events or actions. The internal action is ε. The action $\lightning$ is program termination.*

Definition 2. *A programming language is a tuple $\left(\mathcal{P}, \vdash_W, \overset{e}{\hookrightarrow}, \bowtie\right)$ s.t.:*

- $\mathcal{P}$ *: Set - is a set of admissible, partial programs.*
- $\vdash_W$ *: $\mathcal{P} \to \boldsymbol{P}$ - a judgement that holds iff a program is not partial.*
- $\overset{e}{\hookrightarrow}$ *: $\mathcal{W} \to E \cup \{\lightning\} \to \mathcal{W}$ - a step relation, where $\mathcal{W} = \{w \in \mathcal{P} \mid \ \vdash_W w\}$. For $e \in E \cup \{\lightning\}$ and $p, p' \in \mathcal{W}$ we say for $p \overset{e}{\hookrightarrow} p'$ that program p performs a step with action e to program p'. If $e = \varepsilon$, we write $p \hookrightarrow p'$. In case $e = \lightning$, we write $p \Downarrow$.*
- $\bowtie$ *: $\mathcal{P} \to \mathcal{P} \to \mathcal{P}$ - links two partial programs together in some way, resulting in a new partial program.*

Let S, I, and $\mathbf{T}$ be any programming language.

Definition 3. *A trace $\overline{\tau}$ is an infinite sequence of events that results from the relation $\overset{e}{\hookrightarrow}$. That is, we obtain the trace $\overline{\tau} = e_0 e_1 \ldots$ for the execution sequence $p \overset{e_0}{\hookrightarrow} p' \overset{e_1}{\hookrightarrow} \ldots$ and write $p \rightsquigarrow \overline{\tau}$. The set of all traces is Traces.*

We assume $\lightning$ to occur at most once in the trace and if it does occur, an infinite sequence of ε follows.

Definition 4. *A finite sequence of events m is a finite trace prefix of $\overline{\tau}$ iff it satisfies the following judgement. We write the empty, finite trace prefix as $\cdot$.*

$$\frac{}{\cdot \leq \overline{\tau}} \qquad \frac{m \leq \overline{\tau}}{e :: m \leq e :: \overline{\tau}}$$

Definition 5. *The behavior of a whole program p is a set of all traces it produces, i.e. $Behav(p) = \{\overline{\tau} \mid p \rightsquigarrow \overline{\tau}\}$.*

Definition 6. *A property π is a set of admissible traces. Thus, if p satisfies π (written $p \vDash \pi$), then $Behav(p) \subseteq \pi$.*

Definition 7. *A hyperproperty H is a set of sets of admissible traces. Thus, if p satisfies π (also written $p \vDash H$), then $Behav(p) \in H$.*

Note th

Definition 8. *For every property π, there is a unique hyperproperty that expresses the same property, namely $\mathscr{P}(\pi)$, where $\mathscr{P}(\bullet)$ is the powerset. We write $[]$.*

Definition 9. *A program p robustly satisfies a property π, written $p \vDash_R \pi$, iff $\forall C \in \mathcal{P}, C \bowtie p \vDash \pi$. The same notation is used for robust hyperproperty satisfaction.*

Definition 10. *A (hyper-)property class $\mathcal{C}$ is a set of (hyper-)properties.*

Definition 11. *The class of safety properties contains all properties that can be refuted with a finite trace prefix:*

$$Safety = \{\pi \mid \forall \overline{\tau} \in Traces, t \notin \pi \text{ iff } \exists m \geq \overline{\tau}, \forall \overline{\tau}' \in Traces, m \leq \overline{\tau}' \implies \overline{\tau}' \notin \pi\}$$

Definition 12. *A compiler between languages* S *and* $\mathbf{T}$ *is a partial function* $[\![\bullet]\!]^{\mathsf{S}\to\mathbf{T}}$ *from* $\mathcal{P}$ *to* $\mathcal{P}$*.*

Definition 13. *For a given class* $\mathbb{C}$*, a compiler from language* S *to* $\mathbf{T}$ *robustly preserves* $\mathbb{C}$ *iff*

$$\forall \pi \in \mathbb{C}, \forall \mathsf{p} \in \mathcal{P}, \mathsf{p} \vDash \pi \implies [\![\mathsf{p}]\!]^{\mathsf{S}\to\mathbf{T}} \vDash \pi$$

We write $\vdash [\![\bullet]\!]^{\mathsf{S}\to\mathbf{T}} : \mathbb{C}$*.*

Definition 14. *Given two compilers* $[\![\bullet]\!]^{\mathsf{S}\to I}$ *and* $[\![\bullet]\!]^{I\to\mathbf{T}}$*, their sequential composition is* $[\![\bullet]\!]^{\mathsf{S}\to I\to\mathbf{T}} = [\![[\![\bullet]\!]^{\mathsf{S}\to I}]\!]^{I\to\mathbf{T}}$*.*

Definition 15. *The conjunctive composition of two properties* π_1, π_2 *is the set-intersection* $\pi_1 \cap \pi_2$

Do the lifting

Definition 16. *The conjunctive composition of two classes of properties* $\mathbb{C}_1, \mathbb{C}_2$ *is the set-intersection* $\mathbb{C}_1 \cap \mathbb{C}_2$*.*

Lemma 1. *Given* $\vdash [\![\bullet]\!]^{\mathsf{S}\to I} : \mathbb{C}_1$ *and* $\vdash [\![\bullet]\!]^{I\to\mathbf{T}} : \mathbb{C}_2$*, then* $\vdash [\![\bullet]\!]^{\mathsf{S}\to I\to\mathbf{T}} : \mathbb{C}_1 \cap \mathbb{C}_2$*.*