Intuitionistic LTL and a New Characterization of Safety and Liveness

Patrick Maier

MPI–I–2004–2–002 July 2004

Author's Address

Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
Phone: +49 681 9325-218
Fax: +49 681 9325-299
Email: maier@mpi-sb.mpg.de

Publication Notes

This report is the extended version of a paper presented at the *Annual Conference of the European Association for Computer Science Logic* (CSL '04), Karpacz, Poland, September 20 to 24, 2004.

Acknowledgements

The author thanks Viorica Sofronie-Stokkermans for the many intensive discussions that greatly helped to develop and clarify the ideas of this report. Thanks to an anonymous reviewer for directing our attention to the connection between intuitionistic safety/liveness and the Scott topology on strings.

Abstract

Classical linear-time temporal logic (LTL) is capable of specifying of and reasoning about infinite behaviors only. While this is appropriate for specifying non-terminating reactive systems, there are situations (e. g., assume-guarantee reasoning, run-time verification) when it is desirable to be able to reason about finite and infinite behaviors. We propose an interpretation of the operators of LTL on finite and infinite behaviors, which defines an intuitionistic temporal logic (ILTL). We compare the expressive power of LTL and ILTL. We demonstrate that ILTL is suitable for assume-guarantee reasoning and for expressing properties that relate finite and infinite behaviors. In particular, ILTL admits an elegant logical characterization of safety and liveness properties.

Keywords

Linear-time temporal logic, intuitionistic logic, safety and liveness properties.

1 Introduction

Linear-time temporal logic (LTL) [17] is a convenient specification language for reactive systems. The underlying computational model is that of an infinite behavior, i. e., a non-terminating sequence of interactions between the system and its environment, which makes LTL a specification language for infinite behaviors only. In theory, this is not a problem because every reactive system with finite (and infinite) behaviors can be transformed into one which exhibits only infinite behaviors. In practice, however, it is sometimes essential to reason about finite and infinite behaviors simultaneously and, perhaps, to distinguish finite from infinite behaviors. For example, in run-time verification one needs to relate observed (real) finite behaviors to specified (ideal) infinite behaviors in order to determine whether the observations violate the specification or not. Or, in modular verification, one has to check that a component satisfies an assume-guarantee specification, which amounts to checking that the component keeps satisfying the guarantee at least as long an arbitrary environment satisfies the assumption. Here again, assumption and guarantee are specified as sets of infinite behaviors whereas it is natural to view the component as a prefix-closed set of finite (and possibly infinite) behaviors.

There are various suggestions as how to extend LTL to finite behaviors. For instance, [12] extends the logic with weak and strong next operators whose interpretations differ at the end of finite behaviors. Likewise, [7] interprets LTL formulas by weak and strong semantics, which also differ on finite behaviors. In contrast, we propose a semantics for LTL that treats finite and infinite behaviors uniformly. Inspired from the above view of reactive systems as prefix-closed sets of finite and infinite behaviors, our semantics is based on prefix-closed sets. This gives rise to a Heyting algebra of prefix-closed sets rather than a Boolean algebra (because the complement of a prefix-closed set need not be prefix-closed), so we end up with ILTL, an intuitionistic variant of LTL. The idea of using the Heyting algebra of prefix-closed sets of behaviors as the semantic basis for an intuitionistic logic can also be found in [3], [2] and [13]. However, the interpretation of the temporal operators of LTL in this Heyting algebra seems novel to this paper. Departing from the semantic approach to temporal logic, [6] studies a fragment of ILTL, namely the one generated by the temporal next-operator, using proof-theoretic methods.

In temporal verification, the classification of safety and liveness properties, informally introduced by Lamport [11] and made precise by Alpern and Schneider [4], plays an important role because many (deductive) verification methods are applicable only to safety or liveness properties. Still, these methods are universal thanks to the decomposition theorem [4] (and its effective version for ω-regular properties [5]) stating that every linear-time temporal property can be expressed as a conjunction of a safety and a liveness property. Clearly, a similar classification of safety and liveness properties and a decomposition theorem for our intuitionistic logic ILTL would be desirable. We present a novel abstract classification of safety and liveness properties in a Heyting algebra, which is immediately applicable to all intuitionistic linear-time temporal logics including ILTL, and we prove a decomposition theorem. As the classification only uses the operators of the Heyting algebras, we obtain a simple logical characterization of safety and liveness and an effective decomposition theorem for free.

Over the years, there has been a body of work about safety and liveness. In the direction of generalizing the topology-based results of Alpern and Schneider, [9] proves a decomposition theorem for disjunctively complete Boolean algebras, which [15] generalizes to modular complemented lattices. Our results subsume [9] because every Boolean algebra is a Heyting algebra. However, a modular complemented lattice need not be a Heyting algebra, and vice versa, so [15] is neither subsumed nor does it subsume our results. Beyond linear-time, [14] proposes a classification of safety and liveness for branching time. Concerning effective reasoning with safety and liveness properties, [12] gives syntactic characterizations of safety and liveness properties in LTL with past operators; [18] does the same without using past operators. Interestingly, in the introduction to [16], Plotkin and Stirling shortly put forward some ideas about an intuitionistic linear-time temporal logic and a corresponding classification of safety and liveness properties. We consider it likely that their ideas give rise to the same classification of safety and liveness as ours.

Plan. Section 2 introduces some notation. Section 3 defines the intuitionistic temporal logic ILTL, compares it to its classical companion LTL and illustrates the use ILTL as a semantic basis for assume-guarantee specifications. Section 4 introduces intuitionistic safety and liveness and compares these notions to the classical ones proposed by Alpern and Schneider [4], and Section 5 presents a more abstract algebraic view on intuitionistic safety and liveness. Section 6 concludes.

2 Preliminaries

Behaviors. We fix a non-empty set AP of atomic propositions. By Σ, we denote the power set of AP. Given $p \in AP$, we abbreviate the set of sets containing p by Σ_p, i. e., $\Sigma_p = \{a \in \Sigma \mid p \in a\}$. By Σ^∞, we denote the set of non-empty words over the alphabet Σ. Words can be of finite or infinite length, so Σ^∞ is partitioned into Σ^+ and Σ^ω, the sets of finite and infinite words, respectively. Here in the context of discrete linear-time, a behavior is just a word in Σ^∞.

Power set lattice of behaviors. By $\boldsymbol{\mathcal{P}(\Sigma^\infty)} = \langle \mathcal{P}(\Sigma^\infty), \cap, \cup \rangle$, we denote the power set lattice of Σ^∞, ordered by $\subseteq$. Frequently, we will refer to the elements of this lattice as languages or properties.

We call a function $C : \mathcal{P}(\Sigma^\infty) \to \mathcal{P}(\Sigma^\infty)$ a closure operator on Σ^∞ if C is inflationary, idempotent and monotone, i. e., for all $L, L' \subseteq \Sigma^\infty$, $L \subseteq C(L)$ and $C(C(L)) = C(L)$ and $L \subseteq L'$ implies $C(L) \subseteq C(L')$. We call C a topological closure operator on Σ^∞ if C is a closure operator which distributes over finite joins, i. e., $C(\emptyset) = \emptyset$ and for all $L_1, L_2 \subseteq \Sigma^\infty$, $C(L_1 \cup L_2) = C(L_1) \cup C(L_2)$.

Boolean algebra of sets of infinite behaviors. Let $\inf : \mathcal{P}(\Sigma^\infty) \to \mathcal{P}(\Sigma^\infty)$ be defined by mapping a language L to $\inf(L) = L \cap \Sigma^\omega$, the set of infinite behaviors in L. Note that $\inf$ is an endomorphism of the complete lattice $\boldsymbol{\mathcal{P}(\Sigma^\infty)}$, in particular $\inf$ preserves infinite joins and meets. By INF, we denote the range of $\inf$, i. e., $\mathit{INF} = \{\inf(L) \mid L \subseteq \Sigma^\infty\} = \mathcal{P}(\Sigma^\omega)$. Due to $\inf$ being an endomorphism, INF induces a sublattice of $\boldsymbol{\mathcal{P}(\Sigma^\infty)}$, which turns out to be a complete lattice of sets. In fact, $\boldsymbol{\mathit{INF}} = \langle \mathit{INF}, \cap, \cup, -, \Sigma^\omega, \emptyset \rangle$ is a complete Boolean algebra, where the unary operator $-$ denotes complementation, i. e., $-L = \{w \in \Sigma^\omega \mid w \notin L\}$.

Heyting algebra of prefix-closed sets of behaviors. Let $\preceq$ be the prefix order on Σ^∞, and let $\mathrm{pref}(w) = \{u \in \Sigma^\infty \mid u \preceq w\}$ denote the set of all prefixes of a behavior $w \in \Sigma^\infty$. Thus, $\mathrm{pref} : \Sigma^\infty \to \mathcal{P}(\Sigma^\infty)$ is a function from behaviors to languages. We extend the domain of pref to languages in the usual way, i. e., we define $\mathrm{pref} : \mathcal{P}(\Sigma^\infty) \to \mathcal{P}(\Sigma^\infty)$ by $\mathrm{pref}(L) = \bigcup_{w \in L} \mathrm{pref}(w)$. Note that pref is a closure operator on Σ^∞, which is why we call a language in the range of pref prefix-closed. Moreover, pref preserves infinite joins, yet in general, it does not preserve meets, not even finite ones. By PREF, we denote the range of pref, i. e., $\mathit{PREF} = \{\mathrm{pref}(L) \mid L \subseteq \Sigma^\infty\}$ is the set of prefix-closed languages. Despite pref not preserving all meets, PREF induces a complete sublattice of $\boldsymbol{\mathcal{P}(\Sigma^\infty)}$, which turns out to be a complete lattice of sets. In fact, $\boldsymbol{\mathit{PREF}} = \langle \mathit{PREF}, \cap, \cup, \Rightarrow, \Sigma^\infty, \emptyset \rangle$ is a complete Heyting algebra, i. e., for all languages $L_1, L_2 \in \mathit{PREF}$ there is a greatest language $L \in \mathit{PREF}$, namely $L = \{w \in \Sigma^\infty \mid \mathrm{pref}(w) \cap L_1 \subseteq L_2\}$, such that $L_1 \cap L \subseteq L_2$. We call L the relative pseudo-complement of L_1 and L_2 and denote it by $L_1 \Rightarrow L_2$.

3 Linear-Time Temporal Logics

The set of formulas Form of the linear-time temporal logics considered in this paper is defined by the following grammar, where p ranges over the atomic propositions AP, and φ and ψ range over Form.

$$\mathit{Form} ::= \top \mid \bot \mid p \mid \varphi{\wedge}\psi \mid \varphi{\vee}\psi \mid \varphi{\to}\psi \mid \neg\varphi \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \mid \varphi\mathbf{U}\psi \mid \varphi\mathbf{W}\psi$$

For $\varphi, \psi \in \mathit{Form}$, we treat $\varphi \leftrightarrow \psi$ as a shorthand for $(\varphi \to \psi) \wedge (\psi \to \varphi)$. To save on parenthesis, we adopt the convention that the unary operators $\neg$ (negation), $\mathbf{X}$ (next), $\mathbf{F}$ (eventually) and $\mathbf{G}$ (always) have the highest binding power, followed by the binary operators $\mathbf{U}$ (until) and $\mathbf{W}$ (weak until). The remaining binary operators follow with binding power decreasing in the usual order from $\wedge$ (conjunction) to $\vee$ (disjunction) to $\to$ (implication) to $\leftrightarrow$ (equivalence).

We say that a formula is in negation normal form (NNF) if it does not contain implication nor equivalence and negation is applied only to atomic propositions.

$$
\begin{array}{rlrl}
\mathrm{Mod_c}(\top) = \Sigma^\omega & & \mathrm{Mod_c}(\bot) = \emptyset & \\
\mathrm{Mod_c}(\varphi \wedge \psi) = \mathrm{Mod_c}(\varphi) \cap \mathrm{Mod_c}(\psi) & & \mathrm{Mod_c}(\neg\varphi) = -\mathrm{Mod_c}(\varphi) & \\
\mathrm{Mod_c}(\varphi \vee \psi) = \mathrm{Mod_c}(\varphi) \cup \mathrm{Mod_c}(\psi) & & \mathrm{Mod_c}(\varphi \to \psi) = \mathrm{Mod_c}(\neg\varphi \vee \psi) &
\end{array}
$$

$$
\begin{aligned}
\mathrm{Mod_c}(p) &= \Sigma_p\Sigma^\omega = \{w \in \Sigma^\omega \mid \exists a \in \Sigma_p \exists u \in \Sigma^\omega : w = au\} \\
\mathrm{Mod_c}(\mathbf{X}\varphi) &= \mathrm{next_c}(\mathrm{Mod_c}(\varphi)) \\
\mathrm{Mod_c}(\varphi\, \mathbf{U}\, \psi) &= \textstyle\bigcup_{n<\omega} \mathrm{untilnext}[\mathrm{Mod_c}(\varphi), \mathrm{Mod_c}(\psi)]_\mathrm{c}^n(\emptyset) \\
\mathrm{Mod_c}(\varphi\, \mathbf{W}\, \psi) &= \textstyle\bigcap_{n<\omega} \mathrm{untilnext}[\mathrm{Mod_c}(\varphi), \mathrm{Mod_c}(\psi)]_\mathrm{c}^n(\Sigma^\omega) \\
\mathrm{Mod_c}(\mathbf{F}\varphi) &= \textstyle\bigcup_{n<\omega} \mathrm{next}_\mathrm{c}^n(\mathrm{Mod_c}(\varphi)) = \mathrm{Mod_c}(\top\, \mathbf{U}\, \varphi) \\
\mathrm{Mod_c}(\mathbf{G}\varphi) &= \textstyle\bigcap_{n<\omega} \mathrm{next}_\mathrm{c}^n(\mathrm{Mod_c}(\varphi)) = \mathrm{Mod_c}(\varphi\, \mathbf{W}\, \bot)
\end{aligned}
$$

Figure 1: Classical interpretation of formulas.

3.1 Classical Semantics

By interpreting formulas over the Boolean algebra $\boldsymbol{INF}$, we provide a semantical definition of the classical linear-time temporal logic LTL[1], where the classical interpretation function $\mathrm{Mod_c} : \mathit{Form} \to \mathit{INF}$ is defined recursively in figure 1. This definition makes use of the monotone functions $\mathrm{next_c}$ and $\mathrm{untilnext}[L_1, L_2]_\mathrm{c}$ (with parameters $L_1, L_2 \in \mathit{INF}$) on INF, which map a language L to $\mathrm{next_c}(L) = \Sigma L$ and $\mathrm{untilnext}[L_1, L_2]_\mathrm{c}(L) = L_2 \cup (L_1 \cap \mathrm{next_c}(L))$, respectively.

Given sets of formulas Φ and Ψ, we say that Φ classically entails Ψ, denoted by $\Phi \models_\mathrm{c} \Psi$, if $\bigcap_{\varphi\in\Phi} \mathrm{Mod_c}(\varphi) \subseteq \bigcap_{\psi\in\Psi} \mathrm{Mod_c}(\psi)$. If Φ is a singleton set $\{\varphi\}$, we may omit set braces and write $\varphi \models_\mathrm{c} \Psi$ in place of $\{\varphi\} \models_\mathrm{c} \Psi$; similarly for $\Psi = \{\psi\}$. If Φ is the empty set, we may write $\models_\mathrm{c} \Psi$ in place of $\emptyset \models_\mathrm{c} \Psi$. We call ψ a classical tautology if $\models_\mathrm{c} \psi$.

3.2 Intuitionistic Semantics

Similar to the classical logic LTL above, we define an intuitionistic variant called $ILTL$ by interpreting formulas over the Heyting algebra $\boldsymbol{PREF}$, where the intuitionistic interpretation function $\mathrm{Mod_i} : \mathit{Form} \to \mathit{PREF}$ is defined recursively in figure 2. This definition uses the monotone functions $\mathrm{next_i}$ and $\mathrm{untilnext}[L_1, L_2]_\mathrm{i}$ (with parameters $L_1, L_2 \in \mathit{PREF}$) on PREF, which map a language $L \in \mathit{PREF}$ to $\mathrm{next_i}(L) = \Sigma \cup \Sigma L$ and $\mathrm{untilnext}[L_1, L_2]_\mathrm{i}(L) = L_2 \cup (L_1 \cap \mathrm{next_i}(L))$, respectively.

Given sets of formulas Φ and Ψ, we say that Φ intuitionistically entails Ψ, denoted by $\Phi \models_\mathrm{i} \Psi$, if $\bigcap_{\varphi\in\Phi} \mathrm{Mod_i}(\varphi) \subseteq \bigcap_{\psi\in\Psi} \mathrm{Mod_i}(\psi)$. As in the classical case, we may omit set braces around single formulas, and we may omit the empty set on the left-hand side. We call ψ an intuitionistic tautology if $\models_\mathrm{i} \psi$.

Proposition 1. *For all formulas φ and ψ, $\varphi \models_\mathrm{i} \psi$ if and only if $\models_\mathrm{i} \varphi \to \psi$.*

[1] Although presented differently, this semantics agrees with the standard semantical definition of LTL, cf. [17] or [8].

$$
\begin{array}{rlrl}
\mathrm{Mod_i}(\top) = \Sigma^\infty & & \mathrm{Mod_i}(\bot) = \emptyset & \\
\mathrm{Mod_i}(\varphi \wedge \psi) = \mathrm{Mod_i}(\varphi) \cap \mathrm{Mod_i}(\psi) & & \mathrm{Mod_i}(\varphi \to \psi) = \mathrm{Mod_i}(\varphi) \Rightarrow \mathrm{Mod_i}(\psi) & \\
\mathrm{Mod_i}(\varphi \vee \psi) = \mathrm{Mod_i}(\varphi) \cup \mathrm{Mod_i}(\psi) & & \mathrm{Mod_i}(\neg\varphi) = \mathrm{Mod_i}(\varphi \to \bot) &
\end{array}
$$

$$
\begin{aligned}
\mathrm{Mod_i}(p) &= \Sigma_p \cup \Sigma_p\Sigma^\infty = \{w \in \Sigma^\infty \mid \exists a \in \Sigma_p \exists u \in \Sigma^\infty : w = a \text{ or } w = au\} \\
\mathrm{Mod_i}(\mathbf{X}\varphi) &= \mathrm{next_i}(\mathrm{Mod_i}(\varphi)) \\
\mathrm{Mod_i}(\varphi \,\mathbf{U}\, \psi) &= \textstyle\bigcup_{n<\omega} \mathrm{untilnext}[\mathrm{Mod_i}(\varphi), \mathrm{Mod_i}(\psi)]_{\mathrm{i}}^{n}(\emptyset) \\
\mathrm{Mod_i}(\varphi \,\mathbf{W}\, \psi) &= \textstyle\bigcap_{n<\omega} \mathrm{untilnext}[\mathrm{Mod_i}(\varphi), \mathrm{Mod_i}(\psi)]_{\mathrm{i}}^{n}(\Sigma^\infty) \\
\mathrm{Mod_i}(\mathbf{F}\varphi) &= \textstyle\bigcup_{n<\omega} \mathrm{next}_{\mathrm{i}}^{n}(\mathrm{Mod_i}(\varphi)) = \mathrm{Mod_i}(\top \,\mathbf{U}\, \varphi) \\
\mathrm{Mod_i}(\mathbf{G}\varphi) &= \textstyle\bigcap_{n<\omega} \mathrm{next}_{\mathrm{i}}^{n}(\mathrm{Mod_i}(\varphi)) = \mathrm{Mod_i}(\varphi \,\mathbf{W}\, \bot)
\end{aligned}
$$

Figure 2: Intuitionistic interpretation of formulas.

Proof. Let $\varphi, \psi \in \mathit{Form}$. Then $\varphi \models_{\mathrm{i}} \psi$ if and only if $\mathrm{Mod_i}(\varphi) \subseteq \mathrm{Mod_i}(\psi)$ if and only if $\mathrm{Mod_i}(\varphi) \Rightarrow \mathrm{Mod_i}(\psi) = \Sigma^\infty$ if and only if $\models_{\mathrm{i}} \varphi \to \psi$ □

In summary, the definition of the intuitionistic semantics is largely analogous to the definition of the classical semantics, except for the intuitionistic interpretation of implication and negation and a slight difference in the treatment of the next operator. Note that these differences are forced by the carrier PREF of the Heyting algebra, as the classical interpretations do not result in prefix-closed sets.

3.3 Expressive Power

Comparing the expressive power of LTL and ILTL amounts to comparing the sets of behaviors that can be specified by formulas in these logics. Unfortunately, LTL and ILTL interpret formulas over the two different algebras $\boldsymbol{\mathit{INF}}$ and $\boldsymbol{\mathit{PREF}}$, so we cannot directly compare their interpretations. However, using the defining mappings $\mathrm{inf} : \mathcal{P}(\Sigma^\infty) \to \mathit{INF}$ and $\mathrm{pref} : \mathcal{P}(\Sigma^\infty) \to \mathit{PREF}$ of these algebras, we can map the carrier of each algebra to (a subset of) the carrier of the other and thus compare.

Expressive power in $\boldsymbol{\mathit{INF}}$. First, we compare LTL and ILTL in the Boolean algebra of sets of infinite behaviors $\boldsymbol{\mathit{INF}}$, i. e., we restrict the intuitionistic semantics to infinite words via inf. The proposition below relates the semantics for formulas in negation normal form. From this proposition follows that intuitionistic entailment of formulas in NNF implies classical entailment and that in $\boldsymbol{\mathit{INF}}$, ILTL is at least as expressive as LTL.

Proposition 2. *If φ is a formula in NNF then* $\mathrm{Mod_c}(\varphi) = \mathrm{inf}(\mathrm{Mod_i}(\varphi))$.

Proof. By induction on φ.

- The base cases (the constant $\top$ and $\bot$, atomic propositions and negated atomic propositions) are obvious.

- Conjunction and disjunction are straightforward since $\inf$ distributes over intersection and union.
- Note that for all $L \in PREF$,
$$\mathrm{next}_{\mathrm{c}}(\inf(L)) = \Sigma \inf(L) = \inf(\Sigma \cup \Sigma L) = \inf(\mathrm{next}_{\mathrm{i}}(L)) \ .$$
Therefore, for the next operator (second equality by induction hypothesis)
$$\begin{aligned} \mathrm{Mod}_{\mathrm{c}}(\mathbf{X}\varphi) &= \mathrm{next}_{\mathrm{c}}(\mathrm{Mod}_{\mathrm{c}}(\varphi)) \\ &= \mathrm{next}_{\mathrm{c}}(\inf(\mathrm{Mod}_{\mathrm{i}}(\varphi))) \\ &= \inf(\mathrm{next}_{\mathrm{i}}(\mathrm{Mod}_{\mathrm{i}}(\varphi))) \\ &= \inf(\mathrm{Mod}_{\mathrm{i}}(\mathbf{X}\varphi)) \ . \end{aligned}$$
- Note that for all $L, L_1, L_2 \in PREF$,
$$\begin{aligned} \mathrm{untilnext}[\inf(L_1), \inf(L_2)]_{\mathrm{c}}(\inf(L)) &= \inf(L_2) \cup (\inf(L_1) \cap \mathrm{next}_{\mathrm{c}}(\inf(L))) \\ &= \inf(L_2) \cup (\inf(L_1) \cap \inf(\mathrm{next}_{\mathrm{i}}(L))) \\ &= \inf(L_2 \cup (L_1 \cap \mathrm{next}_{\mathrm{i}}(L))) \\ &= \inf(\mathrm{untilnext}[L_1, L_2]_{\mathrm{i}}(L)) \ . \end{aligned}$$
Therefore, for the until operator (second equality by induction hypothesis)
$$\begin{aligned} \mathrm{Mod}_{\mathrm{c}}(\varphi \,\mathbf{U}\, \psi) &= \textstyle\bigcup_{n<\omega} \mathrm{untilnext}[\mathrm{Mod}_{\mathrm{c}}(\varphi), \mathrm{Mod}_{\mathrm{c}}(\psi)]_{\mathrm{c}}^{n}(\emptyset) \\ &= \textstyle\bigcup_{n<\omega} \mathrm{untilnext}[\inf(\mathrm{Mod}_{\mathrm{i}}(\varphi)), \inf(\mathrm{Mod}_{\mathrm{i}}(\psi))]_{\mathrm{c}}^{n}(\inf(\emptyset)) \\ &= \textstyle\bigcup_{n<\omega} \inf(\mathrm{untilnext}[\mathrm{Mod}_{\mathrm{i}}(\varphi), \mathrm{Mod}_{\mathrm{i}}(\psi)]_{\mathrm{c}}^{n}(\emptyset)) \\ &= \inf(\textstyle\bigcup_{n<\omega} \mathrm{untilnext}[\mathrm{Mod}_{\mathrm{i}}(\varphi), \mathrm{Mod}_{\mathrm{i}}(\psi)]_{\mathrm{c}}^{n}(\emptyset)) \\ &= \inf(\mathrm{Mod}_{\mathrm{i}}(\varphi \,\mathbf{U}\, \psi)) \ . \end{aligned}$$
Similarly, for the weak until operator
$$\begin{aligned} \mathrm{Mod}_{\mathrm{c}}(\varphi \,\mathbf{W}\, \psi) &= \textstyle\bigcap_{n<\omega} \mathrm{untilnext}[\mathrm{Mod}_{\mathrm{c}}(\varphi), \mathrm{Mod}_{\mathrm{c}}(\psi)]_{\mathrm{c}}^{n}(\Sigma^{\omega}) \\ &= \textstyle\bigcap_{n<\omega} \mathrm{untilnext}[\inf(\mathrm{Mod}_{\mathrm{i}}(\varphi)), \inf(\mathrm{Mod}_{\mathrm{i}}(\psi))]_{\mathrm{c}}^{n}(\inf(\Sigma^{\infty})) \\ &= \textstyle\bigcap_{n<\omega} \inf(\mathrm{untilnext}[\mathrm{Mod}_{\mathrm{i}}(\varphi), \mathrm{Mod}_{\mathrm{i}}(\psi)]_{\mathrm{c}}^{n}(\Sigma^{\infty})) \\ &= \inf(\textstyle\bigcap_{n<\omega} \mathrm{untilnext}[\mathrm{Mod}_{\mathrm{i}}(\varphi), \mathrm{Mod}_{\mathrm{i}}(\psi)]_{\mathrm{c}}^{n}(\Sigma^{\infty})) \\ &= \inf(\mathrm{Mod}_{\mathrm{i}}(\varphi \,\mathbf{U}\, \psi)) \ . \end{aligned}$$
- The operators $\mathbf{F}$ and $\mathbf{G}$ can be reduced to $\mathbf{U}$ and $\mathbf{W}$, respectively. □

Corollary 3. *Let Φ and Ψ be sets of formulas in NNF. If $\Phi \models_{\mathrm{i}} \Psi$ then $\Phi \models_{\mathrm{c}} \Psi$.*

Proof. Assume $\Phi \models_{\mathrm{i}} \Psi$, i. e., $\bigcap_{\varphi\in\Phi} \mathrm{Mod}_{\mathrm{i}}(\varphi) \subseteq \bigcap_{\psi\in\Psi} \mathrm{Mod}_{\mathrm{i}}(\psi)$. Then
$$\begin{aligned} \textstyle\bigcap_{\varphi\in\Phi} \mathrm{Mod}_{\mathrm{c}}(\varphi) &= \textstyle\bigcap_{\varphi\in\Phi} \inf(\mathrm{Mod}_{\mathrm{i}}(\varphi)) \\ &= \inf(\textstyle\bigcap_{\varphi\in\Phi} \mathrm{Mod}_{\mathrm{i}}(\varphi)) \\ &\subseteq \inf(\textstyle\bigcap_{\psi\in\Psi} \mathrm{Mod}_{\mathrm{i}}(\psi)) \\ &= \textstyle\bigcap_{\psi\in\Psi} \inf(\mathrm{Mod}_{\mathrm{i}}(\psi)) \\ &= \textstyle\bigcap_{\psi\in\Psi} \mathrm{Mod}_{\mathrm{c}}(\psi) \ , \end{aligned}$$
where the first and the last equality hold by Proposition 2. Hence $\Phi \models_{\mathrm{c}} \Psi$. □

Corollary 4. *In* $\boldsymbol{INF}$, $ILTL$ *is at least as expressive as* LTL.

Proof. We have to show that for every $\varphi \in Form$ there is $\psi \in Form$ such that $\mathrm{Mod_c}(\varphi) = \inf(\mathrm{Mod_i}(\psi))$. This is true because every φ can be transformed into an equivalent formula ψ in NNF by replacing implications and pushing in negations. Hence $\mathrm{Mod_c}(\varphi) = \mathrm{Mod_c}(\psi) = \inf(\mathrm{Mod_i}(\psi))$ by Proposition 2. $\square$

It is unknown whether the converse of Corollary 4 is also true, i. e., whether for all formulas ψ there exist formulas φ such that $\inf(\mathrm{Mod_i}(\psi)) = \mathrm{Mod_c}(\varphi)$. We conjecture that this is the case. However, this seems difficult to prove since in intuitionistic logics, we cannot use equivalence transformations to normal forms like NNF.

Expressive power in $\boldsymbol{PREF}$**.** Now, we compare LTL and $ILTL$ in the Heyting algebra of prefix-closed sets of behaviors $\boldsymbol{PREF}$, i. e., we extend the classical semantics into prefix-closed sets via pref. The proposition below shows that the two logics cannot be equally expressive in $\boldsymbol{PREF}$.

Proposition 5. *There is no formula* φ *with* $\mathrm{pref}(\mathrm{Mod_c}(\varphi)) = \Sigma = \mathrm{Mod_i}(\mathbf{X}\bot)$.

Proof. Let $\varphi \in Form$. If $\mathrm{Mod_c}(\varphi) = \emptyset$ then $\mathrm{pref}(\mathrm{Mod_c}(\varphi)) = \emptyset \neq \Sigma$. Otherwise there is $w \in \mathrm{Mod_c}(\varphi)$, so $\mathrm{pref}(\mathrm{Mod_c}(\varphi)) \neq \Sigma$ because $w \in \mathrm{pref}(\mathrm{Mod_c}(\varphi))$ and $w \in \Sigma^\omega$. $\square$

This implies that either the two logics are incomparable or $ILTL$ is strictly more expressive than LTL, but it is not known which case holds true. We conjecture that $ILTL$ is more expressive than LTL, yet proving this, i. e., proving that for all formulas φ there exist formulas ψ such that $\mathrm{pref}(\mathrm{Mod_c}(\varphi)) = \mathrm{Mod_i}(\psi)$, might require a lemma similar to Proposition 2. However, such a lemma seems difficult to obtain. In particular, the proof of Proposition 2 cannot be directly adapted since it exploits the fact that $\inf$ distributes over intersections, which pref does not do.

3.4 Application: Assume-Guarantee Specifications

Modular verification naturally demands for so-called assume-guarantee specifications (A-G specs), which are pairs of formulas in some temporal logic. Informally, a component of a system satisfies an A-G spec $\varphi \xrightarrow{+} \psi$ if the component satisfies the guarantee ψ at least as long as its environment (including the other components) meets the assumption φ. Once A-G specs are available for all components, properties of the global system may be deduced from the composition (i. e., conjunction) of these A-G specs instead of the (potentially large) parallel composition of all components. Due to possibly circular dependencies between assumptions and guarantees, composing A-G specs in a sound way requires non-trivial composition rules, see for instance [1], [10] or [13].

In the Heyting algebra of prefix-closed sets of finite behaviors, [3] demonstrates that under a suitable notion of concurrency (shared variables and interleaving execution) an A-G spec $\varphi \xrightarrow{+} \psi$ corresponds to an intuitionistic implication $\varphi \to \psi$, which gave rise to composition rules based on conjunction of intuitionistic implication. Later, Abadi and Merz [2] found a more general interpretation of the operator $\xrightarrow{+}$, which again can be reduced to intuitionistic implication. Here, we present their interpretation of $\xrightarrow{+}$ in $\boldsymbol{PREF}$, the Heyting algebra of prefix-closed sets of finite and infinite behaviors. For $\varphi, \psi \in Form$, the semantics of $\varphi \xrightarrow{+} \psi$ is defined by

$$\begin{aligned}&\mathrm{Mod_i}(\varphi \xrightarrow{+} \psi)\\ &= \{w \in \Sigma^\infty \mid \forall v \in \mathrm{pref}(w) : \mathrm{pref!}(v) \subseteq \mathrm{Mod_i}(\varphi) \text{ implies } v \in \mathrm{Mod_i}(\psi)\}\ ,\end{aligned}$$

where $\mathrm{pref!} : \Sigma^\infty \to PREF$ maps behaviors to their sets of proper prefixes, i. e., $\mathrm{pref!}(v) = \mathrm{pref}(v) \setminus \{v\}$. By well-founded induction on the prefix order, [2] proves that for all $\varphi, \psi \in Form$,

$$\mathrm{Mod_i}(\varphi \xrightarrow{+} \psi) = \mathrm{Mod_i}((\psi \to \varphi) \to \psi)\ .$$

Hence in $\boldsymbol{PREF}$, A-G specs are merely short hands for intuitionistic implication. This fact is exploited in [2] to develop concise soundness proofs of various proof rules for conjoining circularly dependent A-G specs.

A general observation about composition rules for A-G specs is that they essentially only admit circular dependencies on safety properties. In classical linear-time temporal logics, this can be achieved by decomposing properties into their safety and liveness parts — which is always possible thanks to the decomposition theorems in [4] and [5] — and disallowing circular dependencies on the liveness parts. Therefore, it is natural to ask for similar decomposition theorems for intuitionistic temporal logics.

4 Safety and Liveness

In this section, we introduce notions of safety and liveness for the intuitionistic temporal logic $ILTL$ and compare them to the corresponding notions for LTL as proposed by Alpern and Schneider [4]. Actually, Alpern and Schneider did not define safety and liveness for LTL but for the Boolean algebra $\boldsymbol{INF}$ of sets of infinite behaviors, over which LTL formulas are interpreted. Consequently, we define safety and liveness for the Heyting algebra $\boldsymbol{PREF}$ of prefix-closed sets of finite and infinite behaviors.

4.1 Safety and Liveness in Classical Logics

We start by reviewing the standard notions of safety and liveness for classical linear-time temporal logics as introduced in [4]. There, safety and liveness are defined in terms of a topology on Σ^ω — in fact, the Cantor topology on Σ^ω if Σ

is finite — which is induced by the topological closure operator C_{c} on Σ^ω with $C_{\mathrm{c}}(L) = \{w \in \Sigma^\omega \mid \mathrm{pref}(w) \cap \Sigma^+ \subseteq \mathrm{pref}(L)\}$ for all $L \subseteq \Sigma^\omega$. We call $L \in \mathit{INF}$ a *classical safety property* if L is closed, i. e., $C_{\mathrm{c}}(L) = L$, and a *classical liveness property* if L is dense, i. e., $C_{\mathrm{c}}(L) = \Sigma^\omega$.

As closed sets of a topological space, classical safety properties are closed under finitary disjunction and infinitary conjunction. And as dense sets, classical liveness properties are closed under infinitary disjunction and under implication.

Proposition 6. *Let $w \in \Sigma^\omega$, let $L_1, L_2 \in \mathit{INF}$ and let $\mathcal{L} \subseteq \mathit{INF}$.*

1. *Σ^ω is a classical safety property.*
2. *$\emptyset$ is a classical safety property.*
3. *$\{w\}$ is a classical safety property.*
4. *If L_1 and L_2 are classical safety properties then so is $L_1 \cup L_2$.*
5. *If all $L \in \mathcal{L}$ are classical safety properties then so is $\bigcap_{L \in \mathcal{L}} L$.*

Proposition 7. *Let $L_1, L_2 \in \mathit{INF}$ and let $\mathcal{L} \subseteq \mathit{INF}$.*

1. *Σ^ω is a classical liveness property.*
2. *If some $L_0 \in \mathcal{L}$ is a classical liveness property then so is $\bigcup_{L \in \mathcal{L}} L$.*
3. *If L_2 is a classical liveness property then so is $L_1 \Rightarrow L_2 = -L_1 \cup L_2$.*

It is instructive to see which logical operations do not preserve classical safety or liveness properties. In the following examples, let p and q be atomic propositions.

- Neither safety nor liveness properties are closed under negation. For instance, $\mathrm{Mod}_{\mathrm{c}}(\mathbf{G}p)$ is a safety property but $\mathrm{Mod}_{\mathrm{c}}(\neg\mathbf{G}p) = \mathrm{Mod}_{\mathrm{c}}(\mathbf{F}\neg p)$ is a liveness property.
- Safety properties are not closed under implication. E. g., $\mathrm{Mod}_{\mathrm{c}}(\mathbf{G}p)$ and $\mathrm{Mod}_{\mathrm{c}}(\mathbf{G}q)$ are safety properties but $\mathrm{Mod}_{\mathrm{c}}(\mathbf{G}p \rightarrow \mathbf{G}q) = \mathrm{Mod}_{\mathrm{c}}(\mathbf{F}\neg p \vee \mathbf{G}q)$ is a liveness property.
- Safety properties cannot be closed under infinitary disjunction. Otherwise, every $L \in \mathit{INF}$ would be a safety property because $L = \bigcup_{w \in L}\{w\}$.
- Liveness properties are not closed under intersection. E. g., $\mathrm{Mod}_{\mathrm{c}}(\mathbf{GF}p)$ and $\mathrm{Mod}_{\mathrm{c}}(\mathbf{FG}\neg p)$ both are liveness properties but $\mathrm{Mod}_{\mathrm{c}}(\mathbf{GF}p \wedge \mathbf{FG}\neg p) = \mathrm{Mod}_{\mathrm{c}}(\mathbf{GF}p \wedge \neg\mathbf{GF}p)$ is not.

The (trivial) property Σ^ω is the only one which is both a safety and liveness property, but there are many properties which are neither. E. g., $L = \mathrm{Mod}_{\mathrm{c}}(p\,\mathbf{U}\,q)$ is such a property because $C_{\mathrm{c}}(L) = \mathrm{Mod}_{\mathrm{c}}(p\,\mathbf{U}\,q \vee \mathbf{G}p) \neq L$ and $C_{\mathrm{c}}(L) \neq \Sigma^\omega$. However, [4] at least proves that all properties in classical linear-time temporal logics can be decomposed into their safety and liveness parts.

Proposition 8. *Every $L \in INF$ is the conjunction of a classical safety and a classical liveness property. More precisely, $L = C_{\mathrm{c}}(L) \cap (-C_{\mathrm{c}}(L) \cup L)$.*

4.2 Safety and Liveness in Intuitionistic Logics

To transfer the notions of safety and liveness to the Heyting algebra $\boldsymbol{PREF}$, we generalize the closure operator $C_{\mathrm{c}} : \mathcal{P}(\Sigma^\omega) \to \mathcal{P}(\Sigma^\omega)$ to $C_{\mathrm{i}} : \mathcal{P}(\Sigma^\infty) \to \mathcal{P}(\Sigma^\infty)$ by defining $C_{\mathrm{i}}(L) = \{w \in \Sigma^\infty \mid \mathrm{pref}(w) \cap \Sigma^+ \subseteq \mathrm{pref}(L)\}$. It turns out that C_{i} is a topological closure operator on Σ^∞ and hence induces a topology — in fact, it induces the Scott topology on Σ^∞ (ordered by the prefix order) if Σ is countable. Thus, we can reuse the topological definitions of safety and liveness and call $L \in PREF$ an *intuitionistic safety property* if $C_{\mathrm{i}}(L) = L$ and an *intuitionistic liveness property* if $C_{\mathrm{i}}(L) = \Sigma^\infty$.

Note that C_{i} is algebraically definable in $\boldsymbol{PREF}$ because for all $L \in PREF$, $C_{\mathrm{i}}(L) = \{w \in \Sigma^\infty \mid \mathrm{pref}(w) \cap \Sigma^+ \subseteq L\} = \Sigma^+ \Rightarrow L$. Therefore, L is an intuitionistic safety property iff $\Sigma^+ \Rightarrow L = L$ iff $\Sigma^+ \Rightarrow L \subseteq L$, and L is an intuitionistic liveness property iff $\Sigma^+ \Rightarrow L = \Sigma^\infty$ iff $\Sigma^+ \subseteq L$ iff $\Sigma^+ \cup L = L$. For comprehending these algebraic definitions, the following intuition might help. Safety and liveness properties differ fundamentally in the way they constrain finite and infinite behaviors. If a safety property is refuted then it can always be refuted by a finite behavior, whereas a liveness property can never be refuted by a finite behavior. So one could say that a safety property L essentially only constrains finite behaviors in the sense that whenever all finite prefixes of an infinite behavior w satisfy L (i. e., $w \in \Sigma^+ \Rightarrow L$) then w satisfies L. Likewise, a liveness property L essentially only constrains infinite behaviors in the sense that all finite behaviors satisfy L.

Intuitionistic safety and liveness properties are closed under essentially the same logical operations as their classical counterparts. Moreover, intuitionistic safety properties are closed under (intuitionistic) implication and negation, and intuitionistic liveness properties are closed under infinitary conjunction.

Proposition 9. *Let $w \in \Sigma^\infty$, let $L, L_1, L_2 \in PREF$ and let $\mathcal{L} \subseteq PREF$.*

1. *Σ^∞ is an intuitionistic safety property.*
2. *$\emptyset$ is an intuitionistic safety property.*
3. *$\mathrm{pref}(w)$ is an intuitionistic safety property.*
4. *If L_1 and L_2 are intuitionistic safety properties then so is $L_1 \cup L_2$.*
5. *If all $L \in \mathcal{L}$ are intuitionistic safety properties then so is $\bigcap_{L \in \mathcal{L}} L$.*
6. *If L_2 is an intuitionistic safety property then so is $L_1 \Rightarrow L_2$.*
7. *If L is an intuitionistic safety property then so is $-L = L \Rightarrow \emptyset$.*

Proof. Claims 2 and 3 follow from the definition of safety because $\Sigma^+ \Rightarrow \emptyset = \emptyset$ and $\Sigma^+ \Rightarrow \mathrm{pref}(w) = \{v \in \Sigma^\infty \mid \mathrm{pref}(v) \cap \Sigma^+ \subseteq \mathrm{pref}(w)\} = \mathrm{pref}(w)$. All other claims follow from Propositions 15 and 17 and Corollary 16, see next section. $\square$

Proposition 10. *Let $L_1, L_2 \in \mathit{PREF}$ and let $\mathcal{L} \subseteq \mathit{PREF}$.*

1. *Σ^∞ is an intuitionistic liveness property.*
2. *If some $L_0 \in \mathcal{L}$ is an intuitionistic liveness property then so is $\bigcup_{L \in \mathcal{L}} L$.*
3. *If L_2 is a intuitionistic liveness property then so is $L_1 \Rightarrow L_2$.*
4. *If all $L \in \mathcal{L}$ are intuitionistic liveness properties then so is $\bigcap_{L \in \mathcal{L}} L$.*

Proof. Follows from Proposition 18, see next section. $\square$

We notice that intuitionistic safety properties are not closed under infinitary disjunction, for the same reason as in the classical case. And intuitionistic liveness properties are not closed under (intuitionistic) negation. E. g., $\mathrm{Mod_i}(\mathbf{F}p)$ is a liveness property but $\mathrm{Mod_i}(\neg\mathbf{F}p) = \mathrm{Mod_i}(\bot)$ is not.

Similar to the classical case, Σ^∞ is the only property which is both an intuitionistic safety and liveness property, cf. Proposition 20. Again, there are many properties which are neither; this follows from Proposition 13 below. Yet, there is also the following decomposition theorem.

Proposition 11. *Every $L \in \mathit{PREF}$ is the conjunction of an intuitionistic safety and an intuitionistic liveness property. More precisely, $L = (\Sigma^+ \Rightarrow L) \cap (\Sigma^+ \cup L)$.*

Proof. Follows from Proposition 19, see next section. $\square$

So far, our approach to safety and liveness was purely semantical, relying only on the operators of the Heyting algebra $\boldsymbol{PREF}$ and the constant Σ^+. However, these operators correspond to the intuitionistic connectives of ILTL, and Σ^+ is expressible in ILTL, namely $\Sigma^+ = \mathrm{Mod_i}(\mathbf{F}\bot)$. Immediately, this gives us a simple logical characterization of intuitionistic safety and liveness and a logical formulation of the decomposition theorem.

Corollary 12. *Let φ be a formula.*

1. *φ is an intuitionistic safety property if and only if $\models_\mathrm{i} (\mathbf{F}\bot \to \varphi) \to \varphi$.*
2. *φ is an intuitionistic liveness property if and only if $\models_\mathrm{i} \mathbf{F}\bot \to \varphi$.*
3. *$\models_\mathrm{i} \varphi \leftrightarrow (\mathbf{F}\bot \to \varphi) \wedge (\mathbf{F}\bot \vee \varphi)$.*

Proof. The claims 1 and 2 follow from the definitions of safety and liveness, respectively, and from Proposition 1. Claim 3 follows from Proposition 11, Proposition 1 (twice) and the fact that $\models_\mathrm{i} \psi_1 \leftrightarrow \psi_2$ follows from $\models_\mathrm{i} \psi_1 \to \psi_2$ and $\models_\mathrm{i} \psi_2 \to \psi_1$ for all $\psi_1, \psi_2 \in \mathit{Form}$. $\square$

4.3 Classical versus Intuitionistic Safety and Liveness

In Section 3, the mappings $\inf : \mathcal{P}(\Sigma^\infty) \to INF$ and $\mathrm{pref} : \mathcal{P}(\Sigma^\infty) \to PREF$ were used to compare the expressive power of the logics LTL and $ILTL$. Now, we will use the same mappings to investigate the relationship between the classical notions of safety and liveness and their intuitionistic counterparts.

It turns out that the intuitionistic notions of safety and liveness subsume the classical ones because every classical safety resp. liveness property is mapped to a corresponding intuitionistic property via pref. However, only the classical notion of safety subsumes the intuitionistic one in the sense that every intuitionistic safety property is mapped to a corresponding classical property via $\inf$. For liveness this is not the case. For instance, Σ^+ is an intuitionistic liveness property to which no corresponding classical property exists, in particular $\inf(\Sigma^+) = \emptyset$ is not a classical liveness property.

Proposition 13. *Let $L \in INF$.*

1. *L is a classical safety property iff $\mathrm{pref}(L)$ is an intuitionistic one.*
2. *L is a classical liveness property iff $\mathrm{pref}(L)$ is an intuitionistic one.*

Proof. The first claim holds because L is a classical safety property
iff $\forall w \in \Sigma^\omega : \mathrm{pref}(w) \cap \Sigma^+ \subseteq \mathrm{pref}(L)$ implies $w \in L$
iff $\forall w \in \Sigma^\infty : \mathrm{pref}(w) \cap \Sigma^+ \subseteq \mathrm{pref}(L)$ implies $w \in \mathrm{pref}(L)$
iff $\forall w \in \Sigma^\infty : w \in \Sigma^+ \Rightarrow \mathrm{pref}(L)$ implies $w \in \mathrm{pref}(L)$
iff $\Sigma^+ \Rightarrow \mathrm{pref}(L) \subseteq \mathrm{pref}(L)$
iff $\mathrm{pref}(L)$ is an intuitionistic safety property.
The second claim holds because L is a classical liveness property if and only if $\forall w \in \Sigma^\omega : \mathrm{pref}(w) \cap \Sigma^+ \subseteq \mathrm{pref}(L)$ if and only if $\Sigma^+ \subseteq \mathrm{pref}(L)$ if and only if $\mathrm{pref}(L)$ is an intuitionistic liveness property. □

Proposition 14. *Let $L \in PREF$.*

1. *If L is an intuitionistic safety property then $\inf(L)$ is a classical one.*
2. *If $\inf(L)$ is a classical liveness property then L is an intuitionistic one.*

Proof. To show the first claim assume that $\Sigma^+ \Rightarrow L \subseteq L$. To show $C_\mathrm{c}(\inf(L)) = \inf(L)$, let $w \in \Sigma^\omega$ with $\mathrm{pref}(w) \cap \Sigma^+ \subseteq \mathrm{pref}(\inf(L))$ and prove that $w \in \inf(L)$, i. e., $w \in L$. We have $\mathrm{pref}(w) \cap \Sigma^+ \subseteq \mathrm{pref}(\inf(L)) \subseteq \mathrm{pref}(L) = L$. Hence $\mathrm{pref}(w) \subseteq \Sigma^+ \Rightarrow L \subseteq L$, which implies $w \in L$.

To show the second claim assume that $C_\mathrm{c}(\inf(L)) = \Sigma^\omega$. Thus, we have $\Sigma^+ \subseteq \mathrm{pref}(\inf(L)) \subseteq \mathrm{pref}(L) = L$, i. e., $\Sigma^+ \subseteq L$. □

Note that the statements of Proposition 14 cannot be reversed. To see this let $L = \mathrm{Mod}_\mathrm{i}(\mathbf{F}\bot \vee \mathbf{G}p)$, where p is an atomic proposition. Then $\inf(L) = \mathrm{Mod}_\mathrm{c}(\mathbf{G}p)$. Thus, $\inf(L)$ is a classical safety property but $\Sigma^+ \Rightarrow L = \Sigma^\infty \neq L$, so L is not an intuitionistic safety property. However, $\Sigma^+ \subseteq L$, so L is an intuitionistic liveness property but $\inf(L)$ is not a classical one.

5 Algebraic Characterization of Safety and Liveness

In this section, we develop notions of safety and liveness and prove a decomposition theorem for arbitrary Heyting algebras. Thus, we provide abstract algebraic proofs for the claims of the previous section about safety and liveness in the concrete Heyting algebra of prefix-closed sets of behaviors $\boldsymbol{PREF}$.

Let $\boldsymbol{H} = \langle H, \sqcap, \sqcup, \Rightarrow, \top, \bot \rangle$ be a Heyting algebra. We denote the order relation on this algebra by $\sqsubseteq$. Recall that $\langle H, \sqcap, \sqcup \rangle$ is a distributive lattice with $\top$ and $\bot$ and for all $x, y, z \in H$, $z \sqsubseteq x \Rightarrow y$ if and only if $x \sqcap z \sqsubseteq y$. This equivalence can be seen as the definition of $x \Rightarrow y$, the pseudo-complement of x relative to y. For $x \in H$, we denote by $-x$ the pseudo-complement of x, which is defined as $-x = x \Rightarrow \bot$. Note that if the law of excluded middle holds in $\boldsymbol{H}$ (i. e., if $x \sqcup -x = \top$ for all $x \in H$) then $x \Rightarrow y = -x \sqcup y$.

By $\mathcal{J}(\boldsymbol{H})$, we denote the join-irreducible elements in $\boldsymbol{H}$, where $j \in H$ is join-irreducible iff $j \neq \bot$ and for all $x, y \in H$, $j = x \sqcup y$ implies $j = x$ or $j = y$. Note that for $j \in \mathcal{J}(\boldsymbol{H})$ and $x, y \in H$, $j \sqsubseteq x \sqcup y$ implies $j \sqsubseteq x$ or $j \sqsubseteq y$ because $\boldsymbol{H}$ is distributive. We call a subset $S \subseteq H$ join-dense in $\boldsymbol{H}$ iff for every $x \in H$ there exists $T \subseteq S$ such that $x = \bigsqcup T$. We call a subset $S \subseteq H$ a forest iff for each $x \in S$, the set $T = \{y \in S \mid y \sqsubseteq x\}$ induces a linear suborder of $\boldsymbol{H}$, i. e., for all $u, v \in T$, $u \sqsubseteq v$ or $v \sqsubseteq u$.

Throughout this section, we fix an arbitrary element $a \in H$, relative to which we will define safety and liveness. In $\boldsymbol{H}$, this a plays the role of Σ^+ in $\boldsymbol{PREF}$, i. e., it separates the 'finite' from the 'infinite' behaviors. Remarkably, the closure properties (except for closure under negation) and the decomposition theorem below hold independent of the choice of a. Thus in $\boldsymbol{PREF}$, we may well choose non-standard separating elements, for instance Σ, to define interesting non-standard notions of safety and liveness.

5.1 Safe Elements

We define the function $\mathrm{safe}_a : H \to H$ by $\mathrm{safe}_a(x) = a \Rightarrow x$. The function safe_a is a closure operator, hence we call safe_a the *safety closure*. We call an element $x \in H$ *a-safe* if x is a fixpoint of this closure, i. e., $\mathrm{safe}_a(x) = x$.

We investigate whether safe elements are closed under the operations of the Heyting algebra and hence under the corresponding intuitionistic connectives. It turns out that safe elements are closed under implication and conjunction, even under infinitary conjunction. Whether safe elements are closed under negation depends on $\bot$ being safe.

Proposition 15. *Let $x, y \in H$, and let $S \subseteq H$ such that $\bigsqcap S$ exists.*

1. *$\top$ is a-safe.*
2. *If y is a-safe then $x \Rightarrow y$ is a-safe.*
3. *If all $s \in S$ are a-safe then $\bigsqcap S$ is a-safe.*

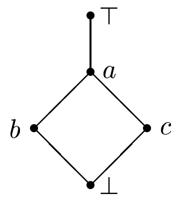

Figure 3: A Heyting algebra where a-safe elements are not closed under join.

Proof. Assume that y and all $s \in S$ are a-safe.

1. $\text{safe}_a(\top) = a \Rightarrow \top = \top$.
2. $\text{safe}_a(x \Rightarrow y) = a \Rightarrow (x \Rightarrow y) = (a \sqcap x) \Rightarrow y = (x \sqcap a) \Rightarrow y = x \Rightarrow (a \Rightarrow y) = x \Rightarrow y$, where the last equality holds because y is a-safe.
3. $\text{safe}_a(\bigsqcap S) = a \Rightarrow \bigsqcap S = \bigsqcap_{s \in S}(a \Rightarrow s) = \bigsqcap_{s \in S} s = \bigsqcap S$, where the second equality holds because $\Rightarrow$ completely distributes over meets on the right-hand side, and the third equality holds because all s are a-safe. □

Corollary 16. *The following statements are equivalent:*

1. *For all $x \in H$, if x is a-safe then $-x$ is a-safe.*
2. *$\bot$ is a-safe.*

Proof. 1 implies 2 because $\top$ is a-safe. 2 implies 1 because $-x = x \Rightarrow \bot$. □

In general, safe elements are not closed under disjunction. For instance, in the Heyting algebra in figure 3, b and c are a-safe because $a \Rightarrow b = b$ and $a \Rightarrow c = c$, but $a \Rightarrow (b \sqcup c) = a \Rightarrow a = \top$, so $b \sqcup c$ is not a-safe. Yet, if the Heyting algebra $\boldsymbol{H}$ satisfies a natural condition, namely that the join-irreducible elements form a join-dense forest, then safe elements are closed under finite disjunction.

Proposition 17. *Let $\mathcal{J}(\boldsymbol{H})$ be a forest, which is join-dense in $\boldsymbol{H}$. Let $x, y \in H$. If x and y are a-safe then $x \sqcup y$ is a-safe.*

Proof. Let x and y be a-safe, i. e., $a \Rightarrow x \sqsubseteq x$ and $a \Rightarrow y \sqsubseteq y$. We have to show that $x \sqcup y$ is a-safe, i. e., $a \Rightarrow (x \sqcup y) \sqsubseteq x \sqcup y$.

As the join-irreducibles are join-dense, there is $J \subseteq \mathcal{J}(\boldsymbol{H})$ such that $\bigsqcup J = a \Rightarrow (x \sqcup y)$. For each $j \in J$, we know that $j \sqsubseteq a \Rightarrow (x \sqcup y)$, so $j \sqcap a \sqsubseteq x \sqcup y$, and we have to show that $j \sqsubseteq x \sqcup y$.

We claim that $j \sqsubseteq a \Rightarrow x$ or $j \sqsubseteq a \Rightarrow y$. This claim implies that $j \sqsubseteq x \sqcup y$. To see this note that in the first case, $j \sqsubseteq a \Rightarrow x \sqsubseteq x \sqsubseteq x \sqcup y$ holds because x is a-safe, and in the second case $j \sqsubseteq a \Rightarrow y \sqsubseteq y \sqsubseteq x \sqcup y$ holds because y is a-safe.

Now, we prove the above claim by contradiction, i. e., we assume that $j \not\sqsubseteq a \Rightarrow x$ and $j \not\sqsubseteq a \Rightarrow y$. This implies $j \sqcap a \not\sqsubseteq x$ and $j \sqcap a \not\sqsubseteq y$.

As the join-irreducibles are join-dense, there is $J' \subseteq \mathcal{J}(\boldsymbol{H})$ such that $\bigsqcup J' = j \sqcap a$. As $j \sqcap a \not\sqsubseteq x$ and $j \sqcap a \not\sqsubseteq y$, there exist $k, l \in J'$ with $k \not\sqsubseteq x$ and $l \not\sqsubseteq y$. Note that $k, l \sqsubseteq j \sqcap a \sqsubseteq j$.

As $\mathcal{J}(\boldsymbol{H})$ is a forest, the set $\{i \in \mathcal{J}(\boldsymbol{H}) \mid i \sqsubseteq j\}$ is linearly ordered, in particular $k \sqsubseteq l$ or $l \sqsubseteq k$.

If $k \sqsubseteq l$ then $l \not\sqsubseteq x$ and $l \not\sqsubseteq y$, so $l \not\sqsubseteq x \sqcup y$ because l is join-irreducible, which contradicts $l \sqsubseteq j \sqcap a \sqsubseteq x \sqcup y$. And if $l \sqsubseteq k$ then $k \not\sqsubseteq x$ and $k \not\sqsubseteq y$, so $k \not\sqsubseteq x \sqcup y$, which contradicts $k \sqsubseteq j \sqcap a \sqsubseteq x \sqcup y$. $\square$

Note that in the Heyting algebra in figure 3, safe elements fail to be closed under disjunction because the join-irreducibles b, c and $\top$ do not form a forest. However, in the Heyting algebra $\boldsymbol{PREF}$ of prefix-closed sets of behaviors, the join-irreducibles are the prefix-closures of single behaviors, i. e., $\mathcal{J}(\boldsymbol{PREF}) = \{\mathrm{pref}(w) \mid w \in \Sigma^\infty\}$. Obviously, $\mathcal{J}(\boldsymbol{PREF})$ forms a forest, which is join-dense in $\boldsymbol{PREF}$. Hence, safety properties in $\boldsymbol{PREF}$ are closed under finite disjunction.

5.2 Live Elements

We define the function $\mathrm{live}_a : H \to H$ by $\mathrm{live}_a(x) = a \sqcup x$. The function live_a is a closure operator, hence we call live_a the *liveness closure*. We call an element $x \in H$ *a-live* if x is a fixpoint of this closure, i. e., $\mathrm{live}_a(x) = x$.

Similar to the case for safe elements, we investigate whether live elements are closed under the operations of the Heyting algebra and hence under the corresponding intuitionistic connectives. It turns out that live elements are closed under implication and under finitary and infinitary conjunction and disjunction.

Proposition 18. *Let $x, y \in H$, and let $S, T \subseteq H$ such that $\bigsqcap S$ and $\bigsqcup T$ exist.*

1. *$\top$ is a-live.*
2. *If y is a-live then $x \Rightarrow y$ is a-live.*
3. *If all $s \in S$ are a-live then $\bigsqcap S$ is a-live.*
4. *If some $t_0 \in T$ is a-live then $\bigsqcup T$ is a-live.*

Proof. Assume that y and all $s \in S$ are a-live, and let $t_0 \in T$ be a-live.

1. $\mathrm{live}_a(\top) = a \sqcup \top = \top$.
2. As $\mathrm{live}_a(y) = a \sqcup y = y$, we have $a \sqsubseteq y = y \sqcap (x \Rightarrow y) \sqsubseteq x \Rightarrow y$. Hence $\mathrm{live}_a(x \Rightarrow y) = a \sqcup (x \Rightarrow y) = x \Rightarrow y$.
3. As $\mathrm{live}_a(s) = a \sqcup s = s$ for all $s \in S$, we have $a \sqsubseteq s$ for all $s \in S$, so $a \sqsubseteq \bigsqcap S$. Hence $\mathrm{live}_a(\bigsqcap S) = a \sqcup \bigsqcap S = \bigsqcap S$.
4. $\mathrm{live}_a(\bigsqcup T) = a \sqcup \bigsqcup T = a \sqcup t_0 \sqcup \bigsqcup T = t_0 \sqcup \bigsqcup T = \bigsqcup T$, where the third equality holds because t_0 is a-live. $\square$

5.3 Decomposition Theorem

With the above notions of safety and liveness, just simple reasoning with the laws of Heyting algebras proves that every element of the algebra can be decomposed into a conjunction of a safe and a live part.

Proposition 19. *Every $x \in H$ is the meet of an a-safe and an a-live element. More precisely, $x = \mathrm{safe}_a(x) \sqcap \mathrm{live}_a(x)$.*

Proof. $\mathrm{safe}_a(x) \sqcap \mathrm{live}_a(x) = (a \Rightarrow x) \sqcap (a \sqcup x) = \big((a \Rightarrow x) \sqcap a\big) \sqcup \big((a \Rightarrow x) \sqcap x\big) = (a \sqcap x) \sqcup x = x$, where the third equality holds due to the cancellation laws for the relative pseudo-complement in Heyting algebras, which say that $y \sqcap (y \Rightarrow z) = y \sqcap z$ and $(y \Rightarrow z) \sqcap z = z$ for all $y, z \in H$. □

The above decomposition might be trivial, for instance in the case that x is both safe and live. However, the following proposition shows that this cannot happen for non-trivial x because safe and live elements are separated.

Proposition 20. *No non-trivial element in $\boldsymbol{H}$ is both a-safe and a-live. More precisely, if $x \in H$ is a-safe and a-live then $x = \top$.*

Proof. Let $x \in H$ be a-safe and a-live. Then $x = \mathrm{safe}_a(x) = \mathrm{safe}_a(\mathrm{live}_a(x)) = a \Rightarrow (a \sqcup x) = \top$, where the last equality holds because $y \Rightarrow z = \top$ for all $y, z \in H$ with $y \sqsubseteq z$. □

Whether there are elements which are neither safe nor live (so that the above decomposition is really non-trivial) depends on the Heyting algebra. For example, all elements in figure 3 are a-safe ($\bot$, b, c, $\top$) or a-live (a, $\top$). However as shown in the previous section, in the Heyting algebra $\boldsymbol{PREF}$ of prefix-closed sets of behaviors, there are elements which are neither Σ^+-safe nor Σ^+-live.

Finally, we note that when the Heyting algebra $\boldsymbol{H}$ happens to be a Boolean algebra, the definition of the liveness closure can be reduced to the safety closure, as it is the case in most decomposition theorems, see for instance [4] or [15].

Proposition 21. *If the law of excluded middle holds in $\boldsymbol{H}$ then for all $x \in H$, $\mathrm{live}_a(x) = \mathrm{safe}_a(x) \Rightarrow x$.*

Proof. $\mathrm{safe}_a(x) \Rightarrow x = x \sqcup -\mathrm{safe}_a(x) = x \sqcup -(a \Rightarrow x) = x \sqcup -(-a \sqcup x) = x \sqcup (a \sqcap -x) = (x \sqcup a) \sqcap (x \sqcup -x) = (x \sqcup a) \sqcap \top = x \sqcup a = live_a(x)$. □

6 Conclusion

We have presented $ILTL$, an intuitionistic variant of the linear-time temporal logic LTL, which is capable of specifying sets of finite and infinite behaviors simultaneously. The intuitionistic nature of $ILTL$ comes in handy when doing assume-guarantee reasoning, because special temporal operators that have been introduced to reason about assume-guarantee specifications are definable via the intuitionistic implication. Furthermore, we have given an abstract algebraic definition of notions of safety and liveness suitable for intuitionistic temporal logics. These intuitionistic notions are similar to the classical ones, yet they are more compatible with the logical connectives; in particular, intuitionistic liveness properties are closed under conjunction. The logic $ILTL$ admits an elegant logical characterization of intuitionistic safety and liveness. It remains to be investigated whether our abstract algebraic definition of safety and liveness also applies to other intuitionistic temporal logics, e. g., to intuitionistic variants of CTL.

There are a still number of unresolved questions concerning the logic $ILTL$. The exact expressive power should be determined, one should give an axiomatization, and one should address decidability and complexity of the satisfiability and model checking problems. Whether $ILTL$ can be considered a useful specification language depends on the answers to these questions.

References

[1] Martín Abadi and Leslie Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17(3):507–534, 1995.

[2] Martín Abadi and Stephan Merz. An abstract account of composition. In *20th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, LNCS 969, pages 499–508. Springer, 1995.

[3] Martín Abadi and Gordon D. Plotkin. A logical view of composition. *Theoretical Computer Science*, 114:3–30, 1993.

[4] Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985.

[5] Bowen Alpern and Fred B. Schneider. Recognizing safety and liveness. *Distributed Computing*, 2(3):117–126, 1987.

[6] Rowan Davies. A temporal-logic approach to binding-time analysis. In *Proceedings of the 11th IEEE Symposium on Logic in Computer Science (LICS)*, pages 184–195. IEEE Computer Society, 1996.

[7] Cindy Eisner, Dana Fisman, John Havlicek, Yoad Lustig, Anthony McIsaac, and David Van Campenhout. Reasoning with temporal logic on truncated paths. In *15th International Conference on Computer Aided Verification (CAV)*, LNCS 2725, pages 27–39. Springer, 2003.

[8] E. Allen Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier, 1990.

[9] H. Peter Gumm. Another glance at the Alpern-Schneider characterization of safety and liveness in concurrent executions. *Information Processing Letters*, 47(6):291–294, 1993.

[10] Bengt Jonsson and Yih-Kuen Tsay. Assumption/guarantee specifications in linear-time temporal logic. *Theoretical Computer Science*, 167:47–72, 1996.

[11] Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE Transactions on Software Engineering*, 3(2):125–143, 1977.

[12] Orna Lichtenstein, Amir Pnueli, and Lenore Zuck. The glory of the past. In *Logic of Programs*, LNCS 193, pages 196–218. Springer, 1985.

[13] Patrick Maier. *A Lattice-Theoretic Framework For Circular Assume-Guarantee Reasoning*. PhD thesis, Universität des Saarlandes, Saarbrücken, July 2003.

[14] Panagiotis Manolios and Richard Trefler. Safety and liveness in branching time. In *Proceedings of the 16th IEEE Symposium on Logic in Computer Science (LICS)*, pages 366–374. IEEE Computer Society, 2001.

[15] Panagiotis Manolios and Richard Trefler. A lattice-theoretic characterization of safety and liveness. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 325–333. ACM Press, 2003.

[16] Gordon Plotkin and Colin Stirling. A framework for intuitionistic modal logics. In *Proceedings of the 1st Conference on Theoretical Aspects of Reasoning about Knowledge (TARK)*, pages 399–406. Morgan Kaufmann, 1986.

[17] Amir Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13:45–60, 1981.

[18] A. Prasad Sistla. Safety, liveness and fairness in temporal logic. *Formal Aspects of Computing*, 6:495–511, 1994.

I N F O R M A T I K

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2004-NWG3-001	M. Magnor	Axisymmetric Reconstruction and 3D Visualization of Bipolar Planetary Nebulae
MPI-I-2004-NWG1-001	B. Blanchet	Automatic Proof of Strong Secrecy for Security Protocols
MPI-I-2004-5-001	S. Siersdorfer, S. Sizov, G. Weikum	Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning
MPI-I-2004-4-004	R. Zayer, C. Rssl, H. Seidel	r-Adaptive Parameterization of Surfaces
MPI-I-2004-4-003	Y. Ohtake, A. Belyaev, H. Seidel	3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs
MPI-I-2004-4-002	Y. Ohtake, A. Belyaev, H. Seidel	Quadric-Based Mesh Reconstruction from Scattered Data
MPI-I-2004-4-001	J. Haber, C. Schmitt, M. Koster, H. Seidel	Modeling Hair using a Wisp Hair Model
MPI-I-2004-2-002	P. Maier	Intuitionistic LTL and a New Characterization of Safety and Liveness
MPI-I-2004-2-001	H.d. Nivelle, Y. Kazakov	Resolution Decision Procedures for the Guarded Fragment with Transitive Guards
MPI-I-2004-1-005	S. Schmitt, L. Fousse	A comparison of polynomial evaluation schemes
MPI-I-2004-1-004	N. Sivadasan, P. Sanders, M. Skutella	Online Scheduling with Bounded Migration
MPI-I-2004-1-003	I. Katriel	On Algorithms for Online Topological Ordering and Sorting
MPI-I-2004-1-002	P. Sanders, S. Pettie	A Simpler Linear Time 2/3 - epsilon Approximation for Maximum Weight Matching
MPI-I-2004-1-001	N. Beldiceanu, I. Katriel, S. Thiel	Filtering algorithms for the Same and UsedBy constraints
MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nrnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nrnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening

MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-004	A. Podelski, A. Rybalchenko	Software Model Checking of Liveness Properties via Transition Invariants
MPI-I-2003-2-003	Y. Kazakov, H. Nivelle	Subsumption of concepts in $DL\ \mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-018	G. Schaefer	A Note on the Smoothed Complexity of the Single-Source Shortest Path Problem
MPI-I-2003-1-017	G. Schfer, S. Leonardi	Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games
MPI-I-2003-1-016	G. Schfer, N. Sivadasan	Topology Matters: Smoothed Competitive Analysis of Metrical Task Systems
MPI-I-2003-1-015	A. Kovcs	Sum-Multicoloring on Paths
MPI-I-2003-1-014	G. Schfer, L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, T. Vredeveld	Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm
MPI-I-2003-1-013	I. Katriel, S. Thiel	Fast Bound Consistency for the Global Cardinality Constraint
MPI-I-2003-1-012		- not published -
MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs
MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Polynomial Time Algorithms for Network Information Flow
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	M. Dietzfelbinger, H. Tamaki	On the probability of rendezvous in graphs
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vcking	Random Knapsack in Expected Polynomial Time
MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vcking	Scheduling and Traffic Allocation for Tasks with Bounded Splittability
MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi
MPI-I-2002-1-008	P. Sanders, J.L. Trff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-005	M. Hoefer	Performance of heuristic and approximation algorithms for the uncapacitated facility location problem
MPI-I-2002-1-004	S. Hert, T. Polzin, L. Kettner, G. Schfer	Exp Lab A Tool Set for Computational Experiments
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Trff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property