A Framework for Circular Assume-Guarantee Rules

Patrick Maier

Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
maier@mpi-sb.mpg.de

The goal in modular verification of concurrent systems is to prove that a complex system, a parallel composition of several components, meets a complex specification, a conjunction of several properties. Ideally, in such a proof one would never construct the system as a whole but only establish properties of single components in isolation and later deduce via a suitable proof rule that the composition meets the conjunction of all desired properties. We propose a framework which supports the construction of such proof rules.

Systems and Specifications. For our purposes, the distinction between system and specification is not relevant, so we may view systems as specifications and vice versa. By $\mathcal{S}$, we denote the set of systems, which is quasi-ordered by $\sqsubseteq$ where for $S, \Psi \in \mathcal{S}$, $S \sqsubseteq \Psi$ may either be read as system S refines system Ψ or specification S entails specification Ψ or system S meets specification Ψ. Furthermore, there is an associative and commutative composition operation $\sqcap$ on $\mathcal{S}$, and again $S \sqcap \Psi$ may either be read as the parallel composition of systems S and Ψ or as the conjunction of specifications S and Ψ or as constraining system S to meet specification Ψ. Of course, composition should be monotone w. r. t. entailment and composition should refine each component, i. e., $\forall S_1, S_1', S_2, S_2' \in \mathcal{S} : S_1 \sqsubseteq S_1' \wedge S_2 \sqsubseteq S_2' \Rightarrow S_1 \sqcap S_2 \sqsubseteq S_1' \sqcap S_2'$ and $\forall S_1, S_2 \in \mathcal{S} : S_1 \sqcap S_2 \sqsubseteq S_1$.

We add two more requirements, namely that the composition $\sqcap$ be idempotent and that entailment $\sqsubseteq$ be an order (the latter can usually be achieved by quotienting $\mathcal{S}$ with the equivalence induced by the quasi-order), which makes $\mathcal{S}$ a meet-semilattice. Idempotency of $\sqcap$ is motivated by the fact that conjunction of properties usually is idempotent and composition of systems can be made idempotent in many specification formalisms (process algebras are a prominent exception).

Proof Rules. In general, there are two kinds of proof rules for modular verification, circular and non-circular ones. Given two systems $S_1, S_2 \in \mathcal{S}$ and two specifications $\Psi_1, \Psi_2 \in \mathcal{S}$, a non-circular rule like (1) below allows us to conclude that the composition of both systems satisfies the conjunction of both specifications, provided that S_1 satisfies Ψ_1, and S_2 satisfies Ψ_2 when constrained to satisfy Ψ_1. The second premise can also be cast as S_2 *guarantees* Ψ_2 if Ψ_1 is *assumed* for the environment (of S_2) – hence the wide-spread term *assume-guarantee rule*.

$$\frac{S_1 \sqsubseteq \Psi_1 \qquad \Psi_1 \sqcap S_2 \sqsubseteq \Psi_2}{S_1 \sqcap S_2 \sqsubseteq \Psi_1 \sqcap \Psi_2} \tag{1}$$

In contrast, the circular rule (2) allows us to draw the same conclusion from the weaker premises that S_1 satisfies Ψ_1 only when constrained to satisfy Ψ_2 whereas S_2 satisfies

Ψ_2 only when constrained to satisfy Ψ_1.

$$\frac{\Psi_2 \sqcap S_1 \sqsubseteq \Psi_1 \qquad \Psi_1 \sqcap S_2 \sqsubseteq \Psi_2}{S_1 \sqcap S_2 \sqsubseteq \Psi_1 \sqcap \Psi_2} \tag{2}$$

Establishing soundness is easy for rule (1) but not for (2) – in fact, the circular rule is unsound. Still, a number of similar rules have been published for specific formalisms, see [4, 1, 2] to name just a few.

Proofs of soundness of these rules must somehow break the circularity. This is usually done by an inductive argument involving approximations of the systems and specifications. While the inductive argument is essentially the same everywhere, the approximations vary from formalism to formalism, thus obscuring the common rationale behind circular assume-guarantee rules.

Instead of proving soundness in a particular formalism, we pursue a different approach. We restrict the unsound circular rule (2) by a circularity-breaking side condition and prove soundness in the very general and natural setting of lattice theory. The benefits will be a clear inductive proof of soundness and still, many of the circular rules in the literature can be recovered as instances of our lattice-theoretic assume-guarantee rule.

Generic Circular Rule. In order to break the circularity by induction, we need a notion of approximation. We can achieve this by embedding the meet-semilattice $\mathcal{S}$ in a complete lattice $\mathcal{L}$; note that this is always possible, e. g., via completion with order ideals. As approximations we take the elements of a join-dense subset $\mathcal{A} \subseteq \mathcal{L}$, thus every system $S \in \mathcal{L}$ can be expressed as $\bigsqcup A$ for some $A \subseteq \mathcal{A}$. Furthermore, the order on $\mathcal{A}$ must be well-founded so that it can be used for proofs by well-founded induction.

We say that two properties $\Psi_1, \Psi_2 \in \mathcal{L}$ *extend* along $\mathcal{A}$ if for all $a \in \mathcal{A}$, either Ψ_1 or Ψ_2 is greater than a whenever $\Psi_1 \sqcap \Psi_2$ is an upper bound of $\{b \in \mathcal{A} \mid b \sqsubset a\}$. I. e., it cannot happen that all strict predecessors of a approximate both Ψ_1 and Ψ_2 whereas a approximates none. We will see that this is the central property needed to get the induction through.

Theorem 1 *Let $\langle \mathcal{L}, \sqsubseteq \rangle$ be a complete lattice and let $S_1, S_2, \Psi_1, \Psi_2 \in \mathcal{L}$. Then the rule*

$$\frac{\Psi_2 \sqcap S_1 \sqsubseteq \Psi_1 \qquad \Psi_1 \sqcap S_2 \sqsubseteq \Psi_2}{S_1 \sqcap S_2 \sqsubseteq \Psi_1 \sqcap \Psi_2} \tag{3}$$

is sound if there exists $\mathcal{A} \subseteq \mathcal{L}$ such that

1. *$\mathcal{A}$ is join-dense,*
2. *$\sqsubseteq$ is well-founded on $\mathcal{A}$, and*
3. *Ψ_1 and Ψ_2 extend along $\mathcal{A}$.*

Proof. To prove rule (3), we assume that its premises $\Psi_2 \sqcap S_1 \sqsubseteq \Psi_1$ and $\Psi_1 \sqcap S_2 \sqsubseteq \Psi_2$ hold and that the side condition is true, i. e., there exists a subset $\mathcal{A} \subseteq \mathcal{L}$ such that conditions 1, 2 and 3 are met. By condition 1, there exists a subset $A \subseteq \mathcal{A}$ such that $\bigsqcup A = S_1 \sqcap S_2$; we can w. l. o. g. assume A to be an order ideal in $\langle \mathcal{A}, \sqsubseteq \rangle$. Furthermore, $\sqsubseteq$ is well-founded on A by condition 2.

By well-founded induction on $\langle A, \sqsubseteq \rangle$, we are going to show that $\Psi_1 \sqcap \Psi_2$ is an upper bound of A. Let $a \in A$ and assume that $\Psi_1 \sqcap \Psi_2$ is an upper bound of $\{b \in A \mid b \sqsubset a\}$. Note that $\{b \in A \mid b \sqsubset a\} = \{b \in \mathcal{A} \mid b \sqsubset a\}$ since A is an order ideal in $\langle \mathcal{A}, \sqsubseteq \rangle$. So by condition 3, $a \sqsubseteq \Psi_1$ or $a \sqsubseteq \Psi_2$. Case distinction:

- If $a \sqsubseteq \Psi_1$ then $a \sqsubseteq \bigsqcup A = S_1 \sqcap S_2 \sqsubseteq S_2$ yields $a \sqsubseteq \Psi_1 \sqcap S_2$. Thus $a \sqsubseteq \Psi_2$ by the second premise of rule (3).
- If $a \sqsubseteq \Psi_2$ then $a \sqsubseteq \bigsqcup A = S_1 \sqcap S_2 \sqsubseteq S_1$ yields $a \sqsubseteq \Psi_2 \sqcap S_1$. Thus $a \sqsubseteq \Psi_1$ by the first premise of rule (3).

Hence $a \sqsubseteq \Psi_1 \sqcap \Psi_2$ in both cases, which ends the inductive proof that $\Psi_1 \sqcap \Psi_2$ is an upper bound of A.

We conclude that $\bigsqcup A \sqsubseteq \Psi_1 \sqcap \Psi_2$ which implies the conclusion of rule (3) since $\bigsqcup A = S_1 \sqcap S_2$. □

Theorem 1 can be generalized to more than two properties as the following corollary shows. The rule presented there omits explicit decomposition of the system S but infers that S has the global property Ψ if it satisfies the local properties Ψ_i under the assumption Φ_i. Of course, S should be decomposed for establishing these local properties, and the decomposition should depend on Φ_i and Ψ_i.

Corollary 1 *Let $\langle \mathcal{S}, \sqsubseteq \rangle$ be a complete lattice and $\Phi_1, \ldots, \Phi_n, \Psi_1, \ldots, \Psi_n, \Psi, S \in \mathcal{L}$. Then the rule*

$$\frac{\Phi_1 \sqcap S \sqsubseteq \Psi_1 \quad \cdots \quad \Phi_n \sqcap S \sqsubseteq \Psi_n \qquad \bigsqcap_j \Psi_j \sqcap S \sqsubseteq \bigsqcap_j \Phi_j \qquad \bigsqcap_j \Psi_j \sqcap S \sqsubseteq \Psi}{S \sqsubseteq \Psi} \quad (4)$$

is sound if there exists $\mathcal{A} \subseteq \mathcal{S}$ such that

1. *$\mathcal{A}$ is join-dense,*
2. *$\sqsubseteq$ is well-founded on $\mathcal{A}$, and*
3. *for all i, Φ_i and Ψ_i extend along $\mathcal{A}$.*

Instantiations. The lattice-theoretic proof rules are not directly useful for verification because their very general side conditions are hard to establish in practice. Therefore, we derive instances of these rules for practical modeling formalisms. Generally, there are two ways to do this, top-down vs. bottom up. Top down, we start with entailment and composition, i. e., with a meet-semilattice, which we embed into a complete lattice $\mathcal{L}$ and identify a join-dense and well-founded subset of approximations $\mathcal{A}$. Then, if we find sufficient conditions $\Theta(S_1, S_2)$ to ensure extension of the systems $S_1, S_2 \in \mathcal{L}$ along $\mathcal{A}$, we can replace the general side condition of rule (3) by the simpler condition $\Theta(S_1, S_2)$. Bottom up, we begin with a well-founded ordered set of approximations $\mathcal{A}$, which we embed into a not too large complete lattice $\mathcal{L}$ (so that $\mathcal{A}$ is join-dense in $\mathcal{L}$); thus we get entailment and composition for free.

Instantiations to synchronous modeling formalisms (Moore machines, Kripke structures) can be found in [3]; in particular, that work recovers a widely known circular assume-guarantee rule for Mealy machines, cf. [2]. Here, we are going to sketch an instantiation of corollary 1 to a linear-time temporal logic formalism, thus essentially proving soundness of the rule in [1] for safety properties. By $\sigma \models M$, we denote that the computation σ satisfies the safety property M, where computations are finite words. By $\sigma \models E \xrightarrow{+} M$, we denote that whenever a prefix of length i of σ satisfies the safety property E then a prefix of length $i+1$ of σ satisfies the safety property M. Note that composition (i. e., conjunction) is asynchronous in this formalism if all properties are invariant under stuttering.

Theorem 2 *For safety properties $E, E_1, \dots, E_n, M, M_1, \dots, M_n$ and a computation σ, the following rule is sound.*

$$\frac{\sigma \models E_1 \xrightarrow{+} M_1 \quad \cdots \quad \sigma \models E_n \xrightarrow{+} M_n \qquad \sigma \models E \wedge \bigwedge_j M_j \Rightarrow \bigwedge_j E_j \qquad \sigma \models \bigwedge_j M_j \Rightarrow M}{\sigma \models E \xrightarrow{+} M}$$

Proof sketch. As we are only dealing with safety, we view a temporal property as the set of finite words which satisfy the property, hence the complete lattice $\mathcal{L}$ is the set of all prefix-closed sets of finite words, ordered by inclusion. The approximations $\mathcal{A}$ are the set of all prefix-closures of finite words. Clearly, $\mathcal{A}$ is join-dense and $\sqsubseteq$ is well-founded on $\mathcal{A}$. Furthermore, $\sigma \models E_i \xrightarrow{+} M_i$ implies that E_i and M_i extend along $\mathcal{A}$. Therefore, we can apply Corollary 1 with Φ_i, Ψ_i, Ψ corresponding to E_i, M_i, M, respectively; S corresponds to σ constrained by E. $\square$

Conclusion. We have presented a circular assume-guarantee rule for the very general yet natural setting of complete lattices. Thus, we got a clear inductive proof of soundness, unobscured by technical details (e. g., synchronous/asynchronous semantics) of more practical modeling formalisms. Yet, the framework is general enough to derive a variety of assume-guarantee rules in more complex formalisms as instances. Note that deriving an instance does not involve circular reasoning any more. Rather, one has to find approximations that ensure the side condition (in particular extension) of the generic rule. In this respect, our framework is a step towards systematic 'engineering' of assume-guarantee rules.

References

1. Martín Abadi and Leslie Lamport. Conjoining specifications. *ACM Transactions on Programming Languages and Systems*, 17(3):507–534, 1995.
2. Rajeev Alur and Thomas A. Henzinger. Reactive modules. In *Proc. 11th Annual Symposium On Logic In Computer Science*, pages 207–218, 1996.
3. Patrick Maier. A set-theoretic framework for assume-guarantee reasoning. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *ICALP 2001: 28th International Colloquium on Automata, Languages and Programming*, pages 821–834. Springer, 2001.
4. Jayadev Misra and K. Mani Chandy. Proofs of networks of processes. *IEEE Transactions on Software Engineering*, 7(4):417–426, 1981.