ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.8, August 2009, pp.2026−2036
doi: 10.3724/SP.J.1001.2009.00569

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

直觉线性μ-演算中的合成推理*

KAZMI Syed Asad Raza[1,2,3], 张文辉[1+]

[1](中国科学院 软件研究所 计算机科学重点实验室,北京 100190)

[2](中国科学院 研究生院 信息与工程学院,北京 100049)

[3](Department of Computer Science, Government College University, Lahore 54000, Pakistan)

Compositional Reasoning in Intuitionistic Linear-Time μ-Calculus

KAZMI Syed Asad Raza[1,2,3], ZHANG Wen-Hui[1+]

[1](Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](School of Information Science and Engineering, Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

[3](Department of Computer Science, Government College University, Lahore 54000, Pakistan)

+ Corresponding author: E-mail: zwh@ios.ac.cn

Kazmi SAR, Zhang WH. Compositional reasoning in intuitionistic linear-time μ-calculus. *Journal of Software*, 2009,20(8):2026−2036. http://www.jos.org.cn/1000-9825/569.htm

Abstract: This paper discusses the use of intuitionistic linear-time μ-calculus (IμTL) whose underlying model is based on Heyting algebra of prefixed closed sets as the basis for the specification of assumption and guarantee paradigm, and then propose an assumption-guarantee rule in IμTL. The rule formulated is more general then previously proposed rules that used linear-time temporal logic (LTL) in the specification of assumption and guarantee paradigm and extends the discussion for safety properties of the form "always φ", and therefore represents more uniform reasoning of assumption and guarantee specifications for also supporting circular compositional reasoning.

Key words: compositional reasoning; propositional linear temporal logic; intuitionistic linear time μ-calculus

摘 要: 讨论了以基于前缀封闭集合的 Heyting 代数的直觉解释的线性μ-演算(IμTL)作为描述"假设-保证"的逻辑基础的问题,提出了一个基于 IμTL 的"假设-保证"规则.该规则比往常应用线性时序逻辑(LTL)作为规范语言的那些规则具有更好的表达能力,扩展了对形如"always φ"等安全性质的"假设-保证"的范围,具备更一般的"假设-保证"推理能力及对循环推理的支持.

关键词: 合成推理;命题线性时序逻辑;直觉线性μ-演算

中图法分类号: TP301 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60721061, 60573012 (国家自然科学基金); the National Basic Research Program of China under Grant No.2002CB312200 (国家重点基础研究发展计划(973))

Received 2008-06-03; Accepted 2008-08-07

1 Introduction

Program verification is concerned with the question of whether certain formal model of a system under investigation satisfies certain properties. One of the most common methods is model checking in which each state is traversed for justifying the question of satisfiability. It is well-known that the major problem in the automatic verification of the concurrent systems is the state-space explosion problem and various techniques have been developed in order to resolve state-space explosion problem, and various techniques have been developed in order to resolve state space explosion problem. These include the symbolic representation of state space through binary decision diagram (BDD)[3,5], the partial order methods[3,15] in which the unnecessary interleaving of transitions are suppressed or by applying abstraction and symmetries[2,5]. To reduce the complexity, systems may be decomposed into simpler systems, and similarly the specifications decomposition may also be achieved. The compositional verification[2,4] is one of the possible ways to deal with complexity. It exploits the hierarchical structure of systems and specifications through divided-and-conquer rule. A compositional verification requires the modal verification of a system through its constituent modules, the decomposed parts of larger system. These compositional techniques have been applied in the model checkers like SMV[10], and the Mocha system[12]. Such methods are originally defined in terms of assumption-guarantee paradigm in which each component of a system is specified in terms of assumptions it makes about its environment, and properties it guarantees about the behavior provided the assumption holds.

Classically the composition of two systems P_1 and P_2 is specified as $P_1||P_2$ a system P, where constituents P_1 and P_2 satisfy the "local" properties while their composition P satisfies "global" properties. Suppose P_1 and P_2 satisfy the properties ψ_1 and ψ_2 respectively, and semantically we could represent as $P_1|=\psi_1$ and $P_2|=\psi_2$. The composition would satisfy $\psi_1\wedge\psi_2=\Psi$ that is $P=P_1||P_2|=\Psi$. This kind of composition is sound for various specification languages and concurrency models, but it is often not helpful[16]. In Ref.[16], a compositional rule for "weak-until" is formulated as:

$$\frac{\begin{array}{ll}\sigma\models(\varphi_{11}U_\omega\varphi_{12})\rhd(\psi_{11}U_\omega\psi_{12}) & \sigma\models(\varphi_{21}U_\omega\varphi_{22})\rhd(\psi_{21}U_\omega\psi_{22})\\ (\varphi_1\wedge\psi_{11}\wedge\psi_{21})\models(\varphi_{11}\wedge\varphi_{21}) & (\varphi_2\wedge\psi_{12}\wedge\psi_{22})\models(\varphi_{21}\wedge\varphi_{22})\\ \psi_{11}\wedge\psi_{21}\models\psi_1 & \psi_{12}\wedge\psi_{22}\models\psi_2\end{array}}{\sigma\models(\varphi_1U_\omega\varphi_2)\rhd(\psi_1U_\omega\psi_2)}$$

We discuss the assumption-guarantee specification along this line, and since the assumption-guarantee specification has natural interpretation in intuitionistic domain, we use intuitionistic linear-time μ-calculus (IμTL) as a specification framework. We formulate an assumption-guarantee rule in IμTL which is more general than the above rule formulated in LTL and therefore extends the usual discussion of assumption-guarantee for safety properties of the form $\Box\varphi$ and also those involving temporal operators like "weak-until" (as shown above) and "release".

The rest of the paper is organized as follows. In Section 2, we provide preliminaries regarding framework of intuitionistic linear-time μ-calculus, and semantic interpretation of IμTL. The compositional reasoning for assumption-guarantee specifications in intuitionistic domain is established in Section 3. In this Section we also discuss some applications of the rule for safety property of the form $\Box\varphi$ such that it establishes a circular reasoning rule for Moore machines. In Section 4 we have an overview of some related work, and in Section 5 we have concluding remarks.

2 Intuitionistic Linear-Time μ-Calculus

Behavior and properties: Reactive systems maintain ongoing interaction with their environment, and thus produce computations i.e. infinite sequences of states. In order to analyze behavior of such systems we have some finite set $AP=\{p,q,r,\ldots\}$ of observable propositions about the states, and its power set is represented as 2^{AP}. We fix $\Sigma=\{a,b,c,\ldots\}$ as a set of alphabets where each elements is a subset of AP, and Σ^{∞} is a set of non-empty words over Σ. The set Σ^{∞} is further divided into Σ^{*} and Σ^{ω} as the sets of finite length and infinite length words respectively. In context of discrete linear-time we take behavior as words of Σ^{∞}. A power set of Σ^{∞} is given as $P(\Sigma^{\infty})$, while this set possesses order $\subseteq$ with join ($\cup$) and meet ($\cap$) that is for all $b_1,b_2\in\Sigma^{\infty}$ then $b_1\cup b_2$ and $b_1\cap b_2$ exist. This constitutes a lattice Σ^{∞}, while its power set forms a power set lattice given as $\boldsymbol{P}(\Sigma^{\infty})=\langle P(\Sigma^{\infty}),\cap,\cup\rangle$. We designate the elements of this lattice as properties or languages. In general we define a temporal property as a language of infinite words.

Set of infinite behaviors as Boolean algebra: We define a function $f_B{:}P(\Sigma^{\infty})\to P(\Sigma^{\infty})$ which maps the language L such as $f_B(L)=L\cap\Sigma^{\omega}$ that is a set of infinite behaviors in L. In general, a lattice is said to be complete if joins and meets of every subset of the set constituting a lattice exit and a lattice will be bounded of join and meet of empty set exist. The map $f_B{:}P(\Sigma^{\infty})\to P(\Sigma^{\infty})$ is an endomorphism of the complete lattice $\boldsymbol{P}(\Sigma^{\infty})$ if it is meet and join preserving that is for all $L_1,L_2\in P(\Sigma^{\infty})$ we have $f_B(L_1\cup L_2)=f_B(L_1)\cup f_B(L_2)$ and $f_B(L_1\cap L_2)=f_B(L_1)\cap f_B(L_2)$. The range of the function f_B is denoted by R_B, and is defined as $R_B=\{f_B(L)|L\subseteq\Sigma^{\infty}\}=P(\Sigma^{\omega})$. Being an endomorphism R_B induces a sublattice of $P(\Sigma^{\infty})$. The induced sublattice is given as $\Lambda_B=\langle R_B,\cup,\cap,-,\varnothing\rangle$, which is in fact a complete Boolean algebra, where-denotes the complement of the language L such that $-L=\{\omega\in\Sigma^{\omega}|\omega\notin L\}$.

Prefix-Closed sets of behaviors as heyting algebra: We represent $\preceq$ as the prefix order on Σ^{∞}. For $\omega,u\in\Sigma^{\infty}$, if u is a prefix of ω then $u\preceq\omega$, while $f_H(\omega)=\{u\in\Sigma^{\infty}\mid u\preceq\omega\}$. The mapping $f_H{:}\Sigma^{\infty}\to P(\Sigma^{\infty})$, maps a behavior Σ^{∞} to language ($P(\Sigma^{\infty})$), while we could extend the domain of f_H, and define $f_H{:}P(\Sigma^{\infty})\to P(\Sigma^{\infty})$ which extends the behavior to language. By $f_H(L)$ we mean that $f_H(L)=\bigcup_{\omega\in L}f_H(\omega)$. We say that language L is prefix closed iff $L=f_H(L)$, the set of prefix closed languages. Despite of not preserving all meets R_H induces a complete sublattice of $P(\Sigma^{\infty})$, which turns out to be a complete lattice of sets. Then $\Lambda_H=\langle R_H,\cup,\cap,\Rightarrow,\Sigma^{\infty},\varnothing\rangle$ constitutes a complete Heyting algebra, i.e. for all languages $L_1,L_2\in R_H$ we have a language $L\in R_H$ known to be greatest language as $L=\{\omega\in\Sigma^{\infty}|f_H(\omega)\cap L_1\subseteq L_2\}$ such that $L_1\cap L\subseteq L_2$. The language L is a relative pseudo-complement of L_1 and L_2 and is denoted as $L_1\Rightarrow L_2$.

Linear-Time μ-calculus: The language of μ-calculus is formulated from propositions, standard Boolean connectives, least fixed point μ, greatest fixed point ν, and the temporal operator *nexttime* $\odot$. The set of formulas Ω_{μ} of the linear time μ-calculus (μTL) is defined by the following convention:

$$\Omega_{\mu}::=Z\mid p\mid\bot\mid\mathrm{T}\mid\psi\wedge\varphi\mid\psi\vee\varphi\mid\odot\psi\mid\mu Z.\psi\mid\nu Z.\psi\mid\psi\to\varphi.$$

We have formulas $\varphi,\psi\in\Omega_{\mu}$, p ranges over atomic propositions, Z ranges over $V=\{X,Y,Z,\ldots\}$, the set of variables, $\mu Z.\psi$ is the least fixed point for ψ and its corresponding counterpart for greatest fixed point is $\nu Z.\psi$, whereas the variable Z in $\mu Z.\psi$ and $\nu Z.\psi$ is in the scope of even number of negations.

In linear-time μ-calculus we have bounded and free variables and the formulas with free variables are interpreted with respect to an environment $\rho{:}V\to P(\Sigma^{\infty})$, which maps all free variables to $\varsigma\subseteq\Sigma^{\omega}$. Variables in the μ-calculus can be either free or bounded by a fixed point operator, a formula is said to be closed if the formula doesn't contain free variables.

Classical semantics of μTL: The semantics of μTL is recursively established in an environment defined ρ in which variables are bounded. We define a classical interpretation function I_c^{ρ} whose domain Ω_{μ} is a set of μTL formulae and which maps them over Boolean algebra i.e. $I_c^{\rho}:\Omega_{\mu}\to\Lambda_B$. This function interprets the classical formula of linear-time μ-calculus, and is environment dependent. Let *Ord* be the set of ordinals. The classical

semantics of the set of μTL formulae under the interpretation function I_c^ρ is given below as:

$$I_c^\rho(\mathrm{T}) = \Sigma^\omega$$
$$I_c^\rho(\bot) = \varnothing$$
$$I_c^\rho(\psi \wedge \varphi) = I_c^\rho(\psi) \cap I_c^\rho(\varphi)$$
$$I_c^\rho(\psi \vee \varphi) = I_c^\rho(\psi) \cup I_c^\rho(\varphi)$$
$$I_c^\rho(\psi \to \varphi) = I_c^\rho(\neg\psi \vee \varphi)$$
$$I_c^\rho(\neg\psi) = -I_c^\rho(\psi)$$
$$I_c^\rho(r) = \Sigma_r\Sigma^\omega = \{\omega \in \Sigma^\omega \mid \exists q \in \Sigma_r, \exists u \in \Sigma^\omega : \omega = qu\}$$
$$I_c^\rho(\odot\psi) = next_c(I_c^\rho(\psi))$$
$$I_c^\rho(Z) = \rho(Z)$$
$$I_c^\rho(\mu Z.\psi) = \bigcup_{n\in Ord}[munext_c^\rho(\lambda Z.\psi)]^n(\varnothing)$$
$$I_c^\rho(\nu Z.\psi) = \bigcap_{n\in Ord}[munext_c^\rho(\lambda Z.\psi)]^n(\Sigma^\omega)$$

where the monotonic function $next_c$ and $munext_c^\rho$ are defined as $next_c(L)=\Sigma L$ and $munext_c^\rho(\lambda Z.\psi)(L) = I_c^{\rho[L/Z]}(\psi)$ while $L\in R_B$, and Σ be a set of alphabets. The representation $\lambda Z.\psi$ is an explicit dependence of formula over the variable Z, and this variable maps to a member of $P(\Sigma^\omega)$ through environment $\rho:V\to P(\Sigma^\omega)$. Where the Σ_r is defined as $\Sigma_r=\{q\in 2^{AP} | r\in q\}$.

Intuitionistic semantics of μTL: Intuitionistic interpretation of semantics of μTL maps each formula into element in the Heyting algebra Λ_H. This interpretation is designated as intuitionistic linear-time μ-calculus (IμTL). The interpretation function $I_i^\rho : \Omega_\mu \to \Lambda_H$ is defined below as:

$$I_i^\rho(\mathrm{T}) = \Sigma^\infty$$
$$I_i^\rho(\bot) = \varnothing$$
$$I_i^\rho(\psi \wedge \varphi) = I_i^\rho(\psi) \cap I_i^\rho(\varphi)$$
$$I_i^\rho(\psi \vee \varphi) = I_i^\rho(\psi) \cup I_i^\rho(\varphi)$$
$$I_i^\rho(\psi \to \varphi) = I_i^\rho(\psi) \Rightarrow I_i^\rho(\varphi)$$
$$I_i^\rho(\neg\psi) = -I_i^\rho(\psi \to \bot)$$
$$I_i^\rho(r) = \Sigma_r\Sigma^\infty = \{\omega \in \Sigma^\infty \mid \exists q \in \Sigma_r, \exists u \in \Sigma^\infty : \omega = qu\}$$
$$I_i^\rho(\odot\psi) = next_i(I_i^\rho(\psi))$$
$$I_i^\rho(Z) = \rho(Z)$$
$$I_i^\rho(\mu Z.\psi) = \bigcup_{n\in Ord}[munext_i^\rho(\lambda Z.\psi)]^n(\varnothing)$$
$$I_i^\rho(\nu Z.\psi) = \bigcap_{n\in Ord}[munext_i^\rho(\lambda Z.\psi)]^n(\Sigma^\infty)$$

Intuitionistic Monotonic functions: In establishment of intuitionistic variant of μTL for the formulation of intuitionistic linear-time μ-calculus (IμTL) we have corresponding functions $next_i$ and $munext_i^\rho$, and are given as:

Intuitionistic Monotonic *next*$_i$ function: For Σ be a set of alphabets, and $next_i$ be a monotonic function, then it generates language inductively as $next_i(L)=\Sigma\cup\Sigma L$ for $L\in R_H$. In defining the language or properties we have Σ^* and Σ^ω as the sets of finite and infinite length words respectively and their superset is Σ^∞. Therefore, the properties we are taking into consideration are only for non-empty words and so the monotonic function $next_i$ is over a non-empty language.

Intuitionistic Monotonic *munext*$_i^\rho$ function: In this case we have language $L\in R_H$, and the environment $\rho:V\to P(\Sigma^\infty)$, the monotonic function $munext_i^\rho$ is defined as:

$$munext_i^{\rho}(\lambda Z.\psi)(L) = I_i^{\rho[L/Z]}(\psi)$$

The monotonic function $munext_i^{\rho}$ specifically used to interpret the fixed points dependence of a formula, but in general it interprets the variable dependency of certain linear-time μ-calculus formula, and its substitution.

Expressiveness: Besides the difference in the semantic domains, the classical and intuitionistic semantics interpretation functions differ in interpreting negation, implication, and next operator. These differentiations mainly depend upon the interpretation of IμTL over Heyting algebra while μTL is defined over Boolean algebra.

The sets of behaviors in different logics are compared in order to expedite the comparative expressive power of μTL and that of IμTL. The formulas in the μTL are interpreted over Boolean algebra Λ_B, while the formulas of IμTL are interpreted over Λ_H, the Heyting algebra. Therefore, we cannot compare them directly; rather their corresponding carriers may be compared. As $f_B:P(\Sigma^{\omega})\to R_B$ and $f_H:P(\Sigma^{\omega})\to R_H$, we may compare the semantics in Boolean algebra Λ_B by restricting the intuitionistic semantics to infinite words through f_B, and then by extending the classical semantics into prefixed closed set through f_H for comparison in Heyting algebra Λ_H.

Definition 2.1. A formula is in a negation normal form (NNF) if it does not contain implication or equivalence, and negation is applied only to atomic propositions.

Suppose φ be a closed formula, then both the interpretation functions $I_c^{\rho}(\varphi)$ and $I_i^{\rho}(\varphi)$ in classical domain and in the intuitionistic domain respectively do not depend on ρ. Then we write $I_c(\varphi)$ and $I_i(\varphi)$ for semantics of the closed formulas. Below is a proposition which relates the formulas in NNF for semantics in Boolean and Heyting algebra[13].

Proposition 2.2. If ψ is a closed formula in NNF then $I_c(\psi)=f_B(I_i(\psi))$.

The position 2.2 reflects that IμTL is as least expressive as μTL, since every formula in the classical interpretation corresponds to a formula in NNF, and the deducibility of formula in prefixed closed set is completely expressible in the μTL domain of infinite behaviors.

Satisfiability with respect to a model: A transitional relation among state of a behavior is usually defined through Kripke structure. A Kprike structure over a set of atomic propositions AP is a tuple $M=\langle S,R,I,L\rangle$ where S is a set of states, $R\subseteq S\times S$ is a transition relation, $I\subseteq S$ is a set of initial states, and $L:=S\to 2^{AP}$ is a labeling function. A path in M is a finite or infinite sequence of states such that if s, t are two consecutive states in the sequence then $(s,t)\in R$. For a finite path $\pi=\pi_0,\pi_1,\pi_2,\ldots,\pi_k$, the string of labels over π is $L(\pi_0)L(\pi_1)\ldots L(\pi_k)$. A string over infinite path is similarly defined. Let $[[M]]$ be the set of strings over all paths of M starting from I. Let φ be a safety formula then $M|=_I\varphi$ iff $[[M]]\subseteq I_i^{\rho}(\varphi)$. When restricting to infinite behaviors, this definition coincides with the traditional interpretation of satisfiability. The satisfiability symbol $|=_I$ indicates that the model M is satisfied under the intuitionistic interpretation.

3 Intuitionistic A-G Specifications

Assumption-guarantee specifications are pair of formulas in some temporal logic. Informally, a component of a system satisfies assumption-guarantee specifications, if the component satisfies the guarantee ψ at least as long as its environment meets assumption φ. This specification is sometimes written as $\varphi\Rightarrow\psi$. In a composed system that is one satisfies $\varphi\Rightarrow\psi$ while other $\psi\Rightarrow\varphi$, this implication has some problem depicted in Ref.[6]. The solution proposed in Ref.[11], while this formulation is elaborated and extended in various contexts in Ref.[8]. By making use of linear-temporal logic of Manna and Pnueli[9] the solution to composition has been proposed in Ref.[6] with aspect of formulation concerning the handling of assumption-guarantees with internal handling, which simply are existential quantified variables in LTL.

In context of Heyting algebra of prefix-closed sets of finite behaviors, it has been illustrated in Ref.[8] for a

suitable notion of concurrency an assumption-guarantee specifications $\varphi \xrightarrow{+} \psi$ corresponds to an intuitionistic implication $\varphi \longrightarrow \psi$. This gives rise to composition rules based on conjunction of intuitionistic implication. Afterwards a more general interpretation of the operator $\xrightarrow{+}$ is provided in Ref.[7]. The interpretation again can be reduced to intuitionistic implication. The interpretation of the operator $\xrightarrow{+}$ over Heyting algebra Λ_H of prefix-closed set is given as for $\varphi, \psi \in \Omega_\mu$:

$$I_i(\varphi \xrightarrow{+} \psi) = \{\omega \in \Sigma^\infty \mid \forall \nu \in f_H(\omega) : \widetilde{f_H}(\nu) \subseteq I_i(\varphi) \; implies \; \nu \in I_i(\psi)\},$$

where $\widetilde{f_H}(\nu) : \Sigma^\infty \to R_H$ maps behaviors to their sets of proper prefixes i.e. $\widetilde{f_H}(\nu) = f_H(\nu) \setminus \{\nu\}$. The connective $\xrightarrow{+}$ introduced in Ref.[7] has interpretation as:

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

Hence in Λ_H, A-G specifications are merely short hands for intuitionistic implication. The circular dependency of assumption-guarantee specifications is treated in Ref.[7], and concise soundness proofs of various proof rules regarding circular dependent assumption-guarantee specifications are established. The composition rules for assumption-guarantee specifications are that they essentially only admit circular dependencies on safety properties.

We represent syntactically $\varphi \triangleright \psi$ to denote the property that under assumptions φ, the property ψ is guaranteed. In intuitionistic linear-time μ-calculus framework the safety formula for assumption (A) φ and guarantee (G) ψ, $M \models_I \varphi \triangleright \psi$ is the same as $M \models_I (\psi \to \varphi) \to \psi$ in which $(\psi \to \varphi) \to \psi$ is an IμTL formulas. For brevity, we still write $\varphi \triangleright \psi$ as an abbreviation for the corresponding IμTL formula when discussing A-G specifications.

3.1 A-G specifications semantics

A circular composition rule that one would like for the system $P=P_1 \| P_2$ would be that if P_1 satisfies the property $\varphi_2 \triangleright \varphi_1$ and P_2 satisfies property $\varphi_1 \triangleright \varphi_2$ then P satisfies $\varphi_1 \wedge \varphi_2$. However, because of the circularity of each component making assumptions about the other components yet to be proven guarantees, such rules are hard to construct and are in fact sound only for some special classes of properties.

Definition 1. A model M satisfies $A \triangleright G$, denoted by $M \models_I A \triangleright G$, is defined in the intuitionistic semantics as

$$[[M]] \cap \Sigma^k \subseteq I_i^\rho(A) implies \exists i > k, \; [[M]] \cap \Sigma^i \subseteq I_i^\rho(G).$$

In the classical domain the assumption-guarantee specifications in context of safety property reflects that guarantee to be satisfied upto time instant k+1 whenever the assumption is satisfied upto time instant k. In case of fixed point interpretation in Ref.[16] the interpretation is only for a single trace and over the fixed points of assumption and guarantee specifications. In our semantic definition we have a set of traces for assumption and guarantee, while guarantee has to satisfy at some later state not necessarily to the very successive of k, the trace instant of assumption.

3.2 Composing A-G specifications

In this section, we describe the inference rule for composing A-G specifications in intuitionistic domain. The inference rules established in Ref.[16], require the conditions that (i) if the global assumption is satisfied currently and one of the local guarantees is satisfied eventually, then the other local assumption is satisfied eventually, (ii) if the local guarantee holds then the global guarantee holds eventually, and (iii) if the global assumption is satisfied, then one of the local guarantee holds eventually. Intuitively, we have the same inference rules but our intuitionistic interpretation through Heyting algebra of prefixed closed set deals not only to a single trace or a string rather a set of strings.

We have a composed system with assumption and guarantee specifications given by Φ and Ψ respectively, while its constituents having sets of assumption and guarantee specifications as φ_1, ψ_1 and φ_2, ψ_2 respectively under

model M. Therefore, for $M \models_I \varphi_1 \triangleright \psi_1$ and $M \models_I \varphi_2 \triangleright \psi_2$ we have the respective interpretations as:

$$[[M]] \cap \Sigma^l \subseteq I_i^\rho(\varphi_1) implies \exists k > l,\ [[M]] \cap \Sigma^k \subseteq I_i^\rho(\psi_1),$$

$$[[M]] \cap \Sigma^{l'} \subseteq I_i^\rho(\varphi_2) implies \exists k' > l',\ [[M]] \cap \Sigma^{k'} \subseteq I_i^\rho(\psi_2).$$

To which similar interpretation may be given for their composed system $M \models_I \Phi \triangleright \Psi$.

Let $\Box\varphi$ stands for always φ. In standard discussion of compositional rules for safety properties, typically a rule is formulated to deduce $M_1 \| M_2 \models_I \Box\Phi \triangleright \Box\Psi$ from $M_1 \models_I \Box\varphi_1 \triangleright \Box\psi_1$ and $M_2 \models_I \Box\varphi_2 \triangleright \Box\psi_2$ provided that certain relations on $\Psi, \psi_1, \psi_2, \Phi, \varphi_1, \varphi_2$ holds. Our aim is to develop a more general rule in order to provide a wider basis for compositional reasoning. This is done in two steps as follows.

Theorem 3.1. Let $\odot^k$ represent k times $\odot$ (with $k \geq 1$). The following inference rule is sound.

$$\frac{\begin{array}{l} M \models_I (\nu Z.(\psi_{12} \wedge (\psi_{11} \vee \odot^k Z))) \\ M \models_I (\nu Z.(\psi_{22} \wedge (\psi_{21} \vee \odot^k Z))) \\ (\psi_{11} \vee \psi_{21}) \models \psi_1 \\ (\psi_{12} \vee \psi_{22}) \models \psi_2 \end{array}}{M \models_I (\nu Z.(\psi_2 \wedge (\psi_1 \vee \odot^k Z)))}$$

Proof: We prove that for k=1. The other cases are similar. Let σ be a trace of M. If ψ_1 does hold at all of σ, then positions according to premises both of ψ_{11} and ψ_{21} do not hold. Therefore ψ_{12} and ψ_{22} holds at all positions. Therefore, $(\nu Z.(\psi_2 \wedge (\psi_1 \vee \odot^1 Z)))$ holds. If ψ_1 holds at some positions of σ, and let i be the first one of such positions. Then according to the premises both of ψ_{11} and ψ_{21} do not hold before position i. Therefore ψ_{12} and ψ_{22} hold at all positions before and on position i. Therefore $(\nu Z.(\psi_2 \wedge (\psi_1 \vee \odot^1 Z)))$ holds.

Theorem 3.2. The following inference rule is sound.

$$\frac{\begin{array}{l} M \models_I (\nu Z.(\varphi_{12} \wedge (\varphi_{11} \vee \odot^k Z))) \triangleright (\nu Z.(\psi_{12} \wedge (\psi_{11} \vee \odot^k Z))) \\ M \models_I (\nu Z.(\varphi_{22} \wedge (\varphi_{21} \vee \odot^k Z))) \triangleright (\nu Z.(\psi_{22} \wedge (\psi_{21} \vee \odot^k Z))) \\ (\psi_{11} \vee \psi_{21}) \models \psi_1 \\ (\psi_{12} \vee \psi_{22}) \models \psi_2 \\ (\varphi_1 \wedge \psi_{12} \wedge \psi_{22}) \models \varphi_{11} \wedge \varphi_{21} \\ (\varphi_2 \wedge \psi_{12} \wedge \psi_{22}) \models \varphi_{12} \wedge \varphi_{22} \end{array}}{M \models_I (\nu Z.(\varphi_2 \wedge (\varphi_1 \vee \odot^k Z))) \triangleright (\nu Z.(\psi_2 \wedge (\psi_1 \vee \odot^k Z)))}$$

Proof: Let $\varphi_i' = (\nu Z.(\varphi_{i2} \wedge (\varphi_{i1} \vee \odot^k Z)))$ and $\psi_i' = (\nu Z.(\psi_{i2} \wedge (\psi_{i1} \vee \odot^k Z)))$. For each trace of M, at the first state, we have ψ_{12}, ψ_{22}, since there is some i>0 such that $[[M]] \cap \Sigma^i$ is a subset of the interpretation of ψ_1' and that of ψ_2'. Then we have ψ_2 and therefore there is an i>0 (more specifically i=1) such that $[[M]] \cap \Sigma^i$ is a subset of the interpretation of $(\nu Z.(\psi_2 \wedge (\psi_1 \vee \odot^k Z)))$.

Suppose $[[M]] \cap \Sigma^1$ is a subset of the interpretation of $(\nu Z.(\varphi_2 \wedge (\varphi_1 \vee \odot^k Z)))$. Then for each trace M, φ_2 holds at the first state. We have two cases. One is that φ_1 also holds. The other is that it does not hold.

In the former case, we have $\varphi_{11}, \varphi_{21}$, and then it follows from Theorem 3.1.

In the later case, we have $\varphi_{12}, \varphi_{22}$, since we have ψ_{12}, ψ_{22} and φ_2. Then $[[M]] \cap \Sigma^1$ is a subset of the interpretation of φ_1' and that of φ_2'. Then $[[M]] \cap \Sigma^k$ is a subset of the interpretation of φ_1' and that of φ_2'.

Then there is an i>k, such that $[[M]] \cap \Sigma^i$ is a subset of interpretation of ψ_1' and that of ψ_2'. Therefore for each trace of M, at (k+1)*st* state, we have ψ_{12}, ψ_{22}.

Similar to the reasoning as when we were at the first state, we obtain that $[[M]] \cap \Sigma^{k+1}$ is the subset of the

interpretation of $(\nu Z.(\psi_2 \wedge (\psi_1 \vee \odot^k Z)))$. Continuing the similar reasoning process to that we were at the first state, we obtain $M \models_I (\nu Z.(\varphi_2 \wedge (\varphi_1 \vee \odot^k Z))) \rhd (\nu Z.(\psi_2 \wedge (\psi_1 \vee \odot^k Z)))$.

3.3 Composition of models

Parallel compositions in many frameworks are interpreted as intersection of behaviors[14,16], the composition of specifications can as well be applied to the composition of models. The following compositional rule can be deduced from Theorem 3.2.

$$\frac{\begin{array}{l} M_1 \models_I (\nu Z.(\varphi_{12} \wedge (\varphi_{11} \vee \odot^k Z))) \rhd (\nu Z.(\psi_{12} \wedge (\psi_{11} \vee \odot^k Z))) \\ M_2 \models_I (\nu Z.(\varphi_{22} \wedge (\varphi_{21} \vee \odot^k Z))) \rhd (\nu Z.(\psi_{22} \wedge (\psi_{21} \vee \odot^k Z))) \\ (\psi_{11} \vee \psi_{21}) \models \psi_1 \\ (\psi_{12} \vee \psi_{22}) \models \psi_2 \\ (\varphi_1 \wedge \psi_{12} \wedge \psi_{22}) \models \varphi_{11} \wedge \varphi_{21} \\ (\varphi_2 \wedge \psi_{12} \wedge \psi_{22}) \models \varphi_{12} \wedge \varphi_{22} \end{array}}{M_1 \| M_2 \models_I (\nu Z.(\varphi_2 \wedge (\varphi_1 \vee \odot^k Z))) \rhd (\nu Z.(\psi_2 \wedge (\psi_1 \vee \odot^k Z)))}$$

The rule extends the discussion for safety properties of the form $\Box\varphi$ and the rule for "weak-until" as described in the introduction. A special case of this rule for safety property of the form $\Box\varphi$ is as follows.

Corollary 3.3. The following rule is sound.

$$\frac{\begin{array}{ll} M_1 \models_I \Box\varphi_1 \rhd \Box\psi_1 & M_2 \models_I \Box\varphi_2 \rhd \Box\psi_2 \\ \Phi \models (\psi_1 \to \varphi_2) & \Phi \models (\psi_2 \to \varphi_1) \quad (\psi_1 \wedge \psi_2) \models \Psi \end{array}}{M_1 \| M_2 \models_I \Box\Phi \rhd \Box\Psi}$$

This corollary is obtained by replacing k with 1 in the above rule, φ_{12}, φ_{22}, ψ_{12} and ψ_{22} with φ_1, φ_2, ψ_1 and ψ_2 respectively, whereas $\Box\varphi$ stands for $\nu Z.(\varphi \wedge \odot Z)$, thus, in constitution of $M_1 \models_I \Box\varphi_1 \rhd \Box\psi_1$ we have φ_{11} and ψ_{11} as false in the above parallel composition rule, and similarly for $M_2 \models_I \Box\varphi_2 \rhd \Box\psi_2$ we have same setting for φ_{21} and ψ_{21}. Next, we discuss the application of this rule to composition of Moore Machines.

Composition of Moore machines: Consider a semantic setting of trace tree with Moore machine defining a programming model. We derive compositional rule for Moore machine with synchronous composition. For this first, we present some basic concepts.

Three structures as prefix closed sets: Let N={0,1,2,…} be a set of all natural numbers then a (finite or infinite) tree is a set $\tau \subseteq N^*$ such that if $xn \in \tau$, for $x \in N^*$ then $xm \in \tau$, for all $0<m<n$. The elements of τ represents nodes: the empty string ε is the root, and for each x. The nodes of the form $xn \in \tau$ for $n \in N$, are the children of a node x. The edges of the tree are pairs $\langle x,xn\rangle$, where $x,xn \in \tau$. The number of children of node x is degree of the node, and it is denoted as deg(x). A tree τ is finite if the set τ is finite; otherwise τ is said to be infinite. The set of finite branches and infinite trees is denoted by T_f^N. A node $x \in \tau$ is said to be a depth $|x|$, where $|x|$ denotes the length of string x. For a tree τ, the subtree rooted at $x \in \tau$, is the tree $\tau|_x=\{i|xi \in \tau\}$.

Given sets A and B, an $\langle A,B\rangle$-*labeled* tree is a triple $\hat{\tau}=\langle \tau,\lambda,\delta\rangle$, where τ is a tree $\lambda:\tau \to A$ is a labeling function that maps each node of the tree to an element of A, and $\delta:\tau \times \tau \to B$ is a function that labels each edge $\langle x,xn\rangle$ in τ with $\delta(x,n) \in B$. Given a labeled tree $\hat{\tau}$, the labeled tree rooted at x is $\hat{\tau}|_x=\langle \tau|_x,\lambda',\delta'\rangle$ where $\lambda'(y)=\lambda(xy)$, and $\delta'(y,n)=\delta(xy,n)$.

Moore machines: A Moore machine is a state transition system with input and output ports. Transition depends on the current state and the current values of the input ports. The transition relation must be non-terminating that is for each state and all possible input values there is at least one successor state, but need not to be deterministic. The

value of each port only depends on the current state, but independent of the current input values.

Definition 3.4 (**Moore machines**). A Moore machine is a tuple $P=\langle S,s_o,I,O,L,R\rangle$ where

- S is the set of states,
- $s_o\in S$ is initial state,
- I is the set of input propositions,
- O is the set of output propositions disjoint from I,
- $L:S\to P(O)$ is a function that labels each state with the set of output propositions true in that state, and
- $R\subseteq S\times P(I)\times S$ is the transition relation.

The state space of a Moore machine is a non-empty set as it contains a non-empty subset of initial states. We do not require finite non-determinism that is the set of initial states may be finite and for all states $s\in S$ and all inputs $i\in I$, the set of successor states may be infinite. We are interested in the non-blocking machines, a condition which is necessary for compositional techniques such as assume-guarantee reasoning[1]. A machine is non-blocking if every state has a successor; that is for all $s_o\in S$ and $i\subseteq I$, there exists a state t such that $R(s_o,i,t)$. For P be a Moore machine, we call t be a trace of P if there is $r\in S^*$ such that r is a run of t in P, while we define the trace language of P as a set of all traces of P. In linear-time semantics for Moore machines the language as the set of finite words which it generates or accepts, while this language is prefixed-closed.

Let $P=\langle S,s_o,I,O,L,R\rangle$ be a Moore machine. Let φ be a safety property with atomic propositions in O. $P|=_I\varphi$ iff for every trace tree $\hat{\tau}=\langle\tau,\lambda,\delta\rangle$ of P, and for every $x\in\tau$, $\lambda(x_o),\lambda(x_1),\ldots,\lambda(x_k)\subseteq I_i(\varphi)$ where $x_o=\varepsilon$, $x_k=x$ and x_i is the prefix of x with length i. Then we have the following theorem.

Theorem 3.5. For Moore machines P and Q such that $P||Q$ exits, the following rule is sound.

$$\frac{P|=_I \Box\varphi_1\triangleright\Box\psi_1 \qquad Q|=_I \Box\varphi_2\triangleright\Box\psi_2}{P||Q|=_I \Box\psi_1\triangleright\Box\psi_1}$$

This rule is deducible from Corollary 3.3 by letting $\psi_1=\varphi_2$, $\psi_2=\varphi_1$, φ=true, $\psi=\psi_1\wedge\varphi_1$. We have $\varphi|=(\psi_1\to\varphi_2)$, $\varphi|=(\psi_2\to\varphi_1)$, and $\psi_1\wedge\psi_2|=\psi$.

Discussion: Viswanahans, *et al*.[16] generalize assume-guarantee specifications. They adopted least and greatest fix points of ω-continuous, and ω-co-continuous functions on properties, respectively, where properties are set of computations. They define the assume-guarantee operator via the chain of iterative approximations that converges to fix point. The authors present a number of generic rules for composing assumptions-guarantee representation are least or greatest fix points-truly circular rules are possible only when the assumptions are confined to greatest fix point. They show the generality of their rules by proving several known assumption-guarantee rules, one for LTL, and one for trace containment of Moore machines to be special instance of their rules. In Ref.[14], Henzingner *et al*. proved soundness of circular rules for Moore machines with simulation as refinement. They require their machines to be finitely non-deterministic, which implies that simulation is equivalent to trace-tree containment. The assumption-guarantee specifications in intuitionistic interpretation is quite general, and it extended the discussion for safety properties of the form $\Box\varphi$ and the temporal operators like “weak-until” and “release” for assumption-guarantee specifications become the subcases of our establishment rule. It also indicates that the intuitionistic way of representing assumption-guarantee is convenient and the compositional principle with intuitionistic linear-time μ-calculus can be applied to specific computational model to obtain relevant circular reasoning principle.

4 Related Work

Assumption-guarantee techniques have a wide range of applications. In Ref.[20], it is used in constructing building blocks for specifications which contains inputs, outputs, external variables. Assumptions about the inputs that they rely upon the goals that they guarantee to achieve, and to build more complex specifications one could continue to use systems specified in the same way, but with more complex rely and guarantee conditions. Alternatively[20] provides operators such as conjunction and *until-requires* to combine system specifications. In Ref.[17], an assumption-guarantee technique is used to specify module behaviors for distinguishing between input and output (input assumption and output guarantees) such that a solution formula for submodule construction is given. The problem of submodule construction or equation solving for module composition has some important applications for the real-time control systems, communication gateway design, and component re-use for system design in general[17]. In Ref.[21], assumption and guarantees are explicitly separated to increase the modeling power of the specification language, in order to propose an interface theory for networks of distributed asynchronous components modeled as input-enabled I/O automata. Assumptions and guarantees about a given component into different automata though each interface consists of an environment and specification. A significant advantage of composite interfaces is that one of the parts can be changed without changing the output part. Assumption-guarantee reasoning is a modular formal analysis technique that uses assumptions when checking components in isolation[19]. The success of compositional reasoning depends on discovering appropriate assumptions for all the components so that assumption checking phase will succeed, while a fully automated framework for assumption-guarantee based composition reasoning by automating decomposition has been developed in Ref.[18]. In Ref.[22], an approach for integrating assume-guarantee verification at different phases of system development is proposed, in order to address the scalability issues associated with the verification of complex software systems. The soundness theorems for compositional reasoning rules depend on underlying computational models and can be very involved[23]. In Ref.[23], a proof-theoretic approach for establishing soundness of rules in automated compositional reasoning is developed.

5 Concluding Remarks

We have proposed dealing with assumption-guarantee specifications in the intuitionistic domain, and we have formulated as assumption-guarantee rule for composition of safety properties with IμTL formulas. For the first the assumption-guarantee specification has a natural interpretation in intuitionistic domain such that IμTL is a natural framework for this kind of specifications. For the second, the rule we have formulated is more general than previously proposed rules that use LTL formula in specification of assumption and guarantee applications, for instance, for supporting circular compositional reasoning.

Acknowledgements The authors thank anonymous referees for their comments that helped improving this paper.

References:

[1] Alur R, Henzinger TA. Reactive modules. Formal Methods in System Design, 1999,(15):7−48.

[2] Clark EM, Filorn T, Jha S. Exploting symmetry in temporal logic model checking. In: Proc. of the CAV'93. Number 697. 1993.

[3] Godefroid P, Wolper P. A partial approach to model checking. In: Proc. of the 6th Annual Symp. on Logic in Computer Science. Amsterdam: IEEE Computer Society Press, 1991. 406−416.

[4] Grumberg O, Long DE. Model checking and modular verification. ACM Trans. on Programming Languages and Systems, 1994, 16(3):843−871.

[5] McMillan KL, Dill D, Burch JR, Clark EM, Hwang LJ. Symbolic model checking 10^{20} states and beyond. In: Proc. of the 5th Annula Symp. on Logic in Computer Science. Philadeliphia: IEEE Computer Society Press, 1990. 428−429.

[6] Jonson B, Tsay YK. Assumption/Guarantee specifications in linear-time temporal logic. Theoretical Computer Science, 1996, (167):47−72.

[7] Abadi M, Lamport L. Conjoining specifications. Technical Report 118, SRC DEC, 1993.

[8] Abadi M, Plotkin GD. A logical view of composition. Theoretical Computer Science, 1993,114(1):3−30.

[9] Manna Z, Pnueli A. A hierarchy of temporal properties. In: Proc. of the Principles of Distributed Computing, the 9th Annual ACM Symp. PODC'90. Quebec, 1990. 337−410.

[10] McMillan KL. Verification of an implementation of tomasulo's algorithm by compositional model checking. In: Hu AJ, Vardi M, eds. Proc. of the Conf. on Computer-Aided Verification (CAV's'98). LNCS 1427, Springer-Verlag, 1998. 100−121.

[11] Misra J, Chandy KM. Proofs of networks of processes. IEEE Trans. on Software Engineering, 1981,7(4):417−426.

[12] Alur R, Henzinger TA, Mang FYC, Qadeer S, Rajamani AK, Tariran S. Mocha: Modularity in model checking. In: Hu AJ, Vardi M, eds. Proc. of the Conf. on Computer-adid Verification (CAV's'98). LNCS 1427, Springer-Verlag, 1998. 521−525.

[13] Kazmi SAR, Zhang WH. Intuitionistic linear-time μ-calculus. Journal of Software, 2008,19(12):3122−3133 (in English with Chinese abstract). http://www.jos.org.cn/1000-9825/19/3122.htm

[14] Rajamani SK, Henzinger TA, Dadeer S, Tasiran S. An assume-guarantee rule for checking simulation. ACM Trans. on Programming Languages and Systems, 2002,24(1):51−64.

[15] Valmari A. A stubborn attack on state explosion. In: Proc. of the CAV'90. 1990. 156−165.

[16] Viswanathan M, Viswanathan R. Foundations for circular compositional reasoning. In: Jos JP, Baeten CM, Lenstra JK, Woeginger GJ, eds. Proc. of the 28th Int'l Colloqium on Automata, Languages and Programming (ICALP 2001). LNCS 2719, Springer-Verlag, 2001. 835−847.

[17] Bochmann GV. Submodule construction for specifications with input assumptions and output guarantees. In: Proc. of the 22nd IFIP WG 6.1 Int'l Conf. Houston on Formal Techniques for Networked and Distributed Systems. LNCS 2529, London: Springer-Verlag, 2002. 17−33.

[18] Nam W, Alur R. Learning-Based assume-guarantee reasoning with automatic decomposition. In: Graf S, Zhang WH, eds. Proc. of the ATVA 2006. 2006. 17−170.

[19] Bobaru MG, Pasareanu CS, Giannakopoulou D. Automated assumption-guarantee reasoning by abstraction refinement. In: Proc. of the CAV 2008. Berlin: Springer-Verlag, 2008. 135−148.

[20] Hayes IJ, Jackson MA, Jones CB. Determining the specification of a control system from that of its environment. In: Proc. of the FME 2003. 2003. 154−169.

[21] Larsen KG, Nyman U, Wasowski A. Interface input/output automata. In: Proc. of the FM 2006. 2006. 82−97.

[22] Giannakopoulou D, Pasareanu CS, Cobleigh JM. Assume-Guarantee verification of source code with design-level ssumptions. In: Proc. of the ICSE 2004. 2004. 211−220.

[23] Wang BY. Automatic derivation of compositional rules in automated compositional reasoning. In: Proc. of the CONCUR 2007. LNCS 4703, Berlin: Springer-Verlag, 2007. 303−316.

附中文参考文献:

[13] Kazmi SAR,张文辉.直觉线性μ-演算.软件学报,2008,19(12):3122−3133. http://www.jos.org.cn/1000-9825/19/3122.htm

KAZMI Syed Asad Raza was born in 1965. He remained a Ph.D. research scholar at the Institute of Software, the Chinese Academy of Sciences, and currently working as Assistant Professor at the Department of Computer Science, Government College University, Lahore, Pakistan. His research areas are formal methods, quantum computing and embedded systems.

ZHANG Wen-Hui was born in 1963. He is a professor at the Institute of Software, the Chinese Academy of Sciences and a CCF senior member. His research areas are formal methods and software reliability for developing high quality software.