

Connecting Algebraic and Logical Descriptions of Concurrent Systems *

Naijun Zhan

Lab. of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
z nj@ios.ac.cn

Abstract

Algebraical approach and logical approach are two different methodologies for designing concurrent systems. In this paper, we show some connections between these two approaches. On one hand, we relate a set of primitives of process algebras which exactly corresponds to the primitives of Basic Process Algebra (BPA for short) to the connectives of modal and temporal logics like Fixpoint Logic with Chop (FLC for short). Thus, we can conclude that these logics could be used to compositionally develop complex systems in an algebra-like way. On the other hand, given a context-free process and an equivalence or preorder such as strong bisimulation, we present a uniform method to construct the characteristic formula of the process up to the relation directly from its syntax. So, all reductions concerning processes that are usually performed in an algebraical framework can be done in a logical framework.

Keywords: Algebraical Approach, Logical Approach, Basic Process Algebra, Fixpoint Logic with Chop, Characteristic Formula

1. Introduction

Algebraical approach and logical approach are thought as two of the dominant methodologies for developing concurrent systems, which are completely different. In general, the former is compositional, that is, a complex system is built by applying an algebraic operator defined in the underlining process algebra to some existing subsystems. So, it is easy to find a connection between the structure of a system to be developed and that of its specification or model. In algebraical approach, specifications or models are usually represented as a process term. Therefore, algebraical approach is suitable for describing simulation properties like synchronization, asynchronization, exclusion and so on. But it is difficult to define global properties such as fairness and liveness because it lacks of abstractness.

In contrast, using logical approach to develop a complex system is from a global point of view, and therefore it is appropriate for specifying global properties because of its abstractness. However, it is hard to find a connection between the structure of a system to be developed and that of its specification defined as a logical formula. Thus, logical approach is not suitable for defining simulation properties.

It is a challenging problem on how to combine the two approaches so that we can get the merits from both of them and avoid their disadvantages in the combined approach.

In fact, lots of attempts have been done in this direction. For example, [6, 12] directly introduced the non-deterministic operator “+” into the modal μ -calculus [8] like logics so that the resulted logics have compositionality; [2, 3] discussed the compositionality of linear temporal logic [15] by introducing the chop into the logic, while [16] investigated some logic properties of the extension; [10, 11] studied the decomposition problem of μ -calculus; [19] investigated the compositionality of a fixpoint logic in assume-guarantee style. [20] proved the definability of “+” in the modal μ -calculus and therefore concluded that the modal logics proposed in [6, 12] can be encoded into the modal μ -calculus. Comparing with the previous work where all the logics studied can only express regular properties, [21] extended the result of [20] to Fixpoint Logic with Chop (FLC) by showing the definability of “+” in the logic. FLC was invented by Markus Müller-Olm [14], which is an extension of the modal μ -calculus with the chop operator, strictly more expressive than the modal μ -calculus as non-regular properties can be defined in FLC. A by-product of the compositionality of FLC is that [21] presented an algorithm to construct the characteristic formula of a context-free process up to strong bisimulation directly from its syntax in contrast to the previous work on deriving the characteristic formula of a process up to some equivalence or preorder from its semantics. For example, [7] gave a method to define characteristic formulae for finite terms of CCS up to observational congruence, [17] deepened the work by presenting an approach to define characteristic formulae

*This work is supported in part by NSFC-60493200, NSFC-60421001, NSFC-60573007 and NKBRPC-2002cb312200.

for regular processes up to some preorders; moreover, [14] gave a method to define the characteristic formula for a context-free process up to some preorder based on its rewriting system.

In this paper, we will extend the results of [21] by investigating the compositionality of FLC under an observational semantics, which is a preorder and generalization of observable equivalence of [13]. Similarly to [21], we will first show the definability of “+” in the observable FLC under the observational semantics and then present algorithm to construct the characteristic formula of a context-free process up to the preorder directly from its syntax.

The rest of this paper is organized as follows: Section 2 briefly reviews Basic Process Algebra with Deadlock and Termination [4] ($\text{BPA}_\delta^\varepsilon$). In Section 3, FLC is briefly reviewed and its observable version is defined. Section 4 is devoted to showing the definability of “+” in the observable FLC. In Section 5, we investigate a connection between the constructors of $\text{BPA}_\delta^\varepsilon$ and the connectives of FLC under the observational semantics. In Section 6, we sketch how to construct a formula Ψ_P for each process $P \in \text{BPA}_\delta^\varepsilon$ according to its syntax and then show the formula obtained by eliminating “+” in Ψ_P is the characteristic formula of P up to the preorder. Finally, a brief conclusion is provided in Section 7.

2. Basic Process Algebra with Termination and Deadlock

Let Act be a set of (atomic) observable actions, ranged over a, b, c, \dots , and τ be an unobservable action. We use Act_τ to stand for $Act \cup \{\tau\}$, ranged over α, β, \dots . Let $\mathcal{X} = \{x, y, z, \dots\}$ be a countable set of process variables. *Sequential process terms*, written \mathcal{P}^s , are generated by the following grammar:

$$E ::= \delta \mid \varepsilon \mid x \mid \alpha \mid E_1;E_2 \mid E_1+E_2 \mid \text{rec } x.E,$$

where $\alpha \in Act_\tau$.

Intuitively, δ stands for a deadlocked process that cannot execute any action and keeps idle for ever; ε denotes a terminated process that cannot proceed, but terminates at once; in isolation, x behaves like δ , which is used to define recursive processes; “;” stands for sequential composition while “+” for non-deterministic choice; $\text{rec } x.E$ represents recursion and informally means that the body E can be performed repeatedly.

In order to define an operational semantics for expressions of the form $E_1;E_2$, we need to define a special predicate \mathcal{T} over \mathcal{P}^s to indicate if a given process term is terminated or not¹. Formally, $\mathcal{T} \subset \mathcal{P}^s$ is the least set which contains ε and is closed under the fol-

lowing rules: (i) if $\mathcal{T}(E_1)$ and $\mathcal{T}(E_2)$ then $\mathcal{T}(E_1;E_2)$ and $\mathcal{T}(E_1+E_2)$; (ii) if $\mathcal{T}(E)$ then $\mathcal{T}(\text{rec } x.E)$.

An occurrence of variable $x \in \mathcal{X}$ is called *free* in term E iff it does not occur within a sub-term of the form $\text{rec } x.E'$, otherwise called *bound*. We will use $fn(E)$ to stand for all variables which have some free occurrence in E , and $bn(E)$ for all variables which have some bound occurrence in E . Variable $x \in \mathcal{X}$ is called *guarded* within term E iff every occurrence of x is within a sub-term F where F is prefixed with a subexpression F' via “;” such that $\neg\mathcal{T}(F')$. A term E is called *guarded* iff all variables occurring in it are guarded. The set of all closed and guarded terms of \mathcal{P}^s essentially corresponds to the *basic process algebra* (BPA) with the terminated process ε and the deadlocked process δ , denoted by $\text{BPA}_\delta^\varepsilon$, ranged over by P, Q, \dots , where BPA is a fragment of ACP [4].

An operational semantics of \mathcal{P}^s is given in the standard Plotkin’s style, yielding a transition system $(\mathcal{P}^s, \rightarrow)$ with $\rightarrow \subseteq \mathcal{P}^s \times Act_\tau \times \mathcal{P}^s$ that is the least relation derived from the rules in the Fig.1.

Act	$\frac{}{\alpha \xrightarrow{\alpha} \varepsilon}$
Rec	$\frac{E[\text{rec } x.E/x] \xrightarrow{\alpha} E'}{\text{rec } x.E \xrightarrow{\alpha} E'}$
Seq-1	$\frac{E_1 \xrightarrow{\alpha} E'_1}{E_1;E_2 \xrightarrow{\alpha} E'_1;E_2}$
Seq-2	$\frac{E_2 \xrightarrow{\alpha} E'_2 \wedge \mathcal{T}(E_1)}{E_1;E_2 \xrightarrow{\alpha} E'_2}$
Nd	$\frac{E_1 \xrightarrow{\alpha} E'_1}{E_1+E_2 \xrightarrow{\alpha} E'_1, E_2+E_1 \xrightarrow{\alpha} E'_1}$

Fig.1. The Operational Semantics of \mathcal{P}^s

Definition 1 A binary relation $\mathcal{S} \subseteq \text{BPA}_\delta^\varepsilon \times \text{BPA}_\delta^\varepsilon$ is called a *strong bisimulation* if $(P, Q) \in \mathcal{S}$ then

- $\mathcal{T}(P)$ iff $\mathcal{T}(Q)$;
- whenever $P \xrightarrow{\alpha} P'$ then, for some $Q', Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{S}$ for any $\alpha \in Act_\tau$;
- whenever $Q \xrightarrow{\alpha} Q'$ then, for some $P', P \xrightarrow{\alpha} P'$ and $(P', Q') \in \mathcal{S}$ for any $\alpha \in Act_\tau$.

Given two processes $P, Q \in \text{BPA}_\delta^\varepsilon$, we say that P and Q are strongly bisimilar, written $P \sim Q$, if $(P, Q) \in \mathcal{S}$ for some strong bisimulation \mathcal{S} . We can extend the definition of \sim over \mathcal{P}^s as: let

¹As in [1], here we adopt the semantics of strict termination in the sense that $P+Q$ is terminated iff P and Q both are terminated. This is because using termination to make choice is impractical, and therefore is thought as not well-formed in many literatures, e.g. [5].

$E_1, E_2 \in \mathcal{P}^S$ and $fn(E_1) \cup fn(E_2) \subseteq \{x_1, \dots, x_n\}$, if $E_1\{P_1/x_1, \dots, P_n/x_n\} \sim E_2\{P_1/x_1, \dots, P_n/x_n\}$ for any $P_1, \dots, P_n \in \text{BPA}_\delta^\varepsilon$, then $E_1 \sim E_2$.

In order to define an observational semantics of \mathcal{P}^S , we need the following notations and definitions.

Given an action $\alpha \in \text{Act}_\tau$, we use $\hat{\alpha}$ to denote α if $\alpha \in \text{Act}$, ε otherwise, where ε stands for empty action. Moreover, $\xrightarrow{\varepsilon}$ denotes the identity relation over \mathcal{P}^S , i.e., for any $E \in \mathcal{P}^S$, $E \xrightarrow{\varepsilon} E$. We use $\xrightarrow{\alpha}$ to stand for the sequence of transitions $(\xrightarrow{\tau})^* \cdot \xrightarrow{\alpha} \cdot (\xrightarrow{\tau})^*$ and $\xrightarrow{\varepsilon}$ for $(\xrightarrow{\tau})^*$. In what follows, $E \xrightarrow{t}$ means that $E \xrightarrow{\alpha_1} E_1 \xrightarrow{\alpha_2} E_2 \cdots \xrightarrow{\alpha_n} E_n$ for some E_1, \dots, E_n , where $t = \alpha_1 \cdots \alpha_n \in \text{Act}_\tau^*$.

We say a term E is *convergent*, denoted by $\downarrow(E)$, if and only if E cannot perform an infinite sequence of τ actions, that is, formally, $E \not\xrightarrow{\tau}^\omega$; otherwise, E is called *divergent*, written $\uparrow(E)$. We say $\downarrow_\tau(E)$ if $\downarrow(E)$, $\downarrow_a(E)$ if $\downarrow(E)$, and for each E' , $E \xrightarrow{\alpha} E'$ implies $\downarrow(E')$, where $a \in \text{Act}$. Also, we say a process term E is weak terminated if $\forall E'. (E \xrightarrow{\varepsilon} E' \wedge E' \not\xrightarrow{\tau}) \Rightarrow \mathcal{T}(E')$, denoted $\mathbb{T}(E)$.

Since we want to distinguish the behavior of deadlocked, terminated and divergent processes, instead of using the standard Milner's observable bisimulation [13], we use the following preorder adapted from [1], which is a generalization of Milner's observable bisimulation, to describe the observational behavior of $\text{BPA}_\delta^\varepsilon$.

Definition 2 Let \preceq be a binary relation over $\text{BPA}_\delta^\varepsilon$ which satisfies that for each $P, Q \in \text{BPA}_\delta^\varepsilon$, $P \preceq Q$ iff

- if $\downarrow(P)$, $\mathbb{T}(P)$ iff $\mathbb{T}(Q)$;
- whenever $P \xrightarrow{\alpha} P'$ then, for some $Q', Q \xrightarrow{\hat{\alpha}} Q'$ and $(P' \preceq Q')$;
- if $\downarrow_\alpha(P)$, then
 - (a) $\downarrow_\alpha(Q)$
 - (b) whenever $Q \xrightarrow{\alpha} Q'$ then, for some $P', P \xrightarrow{\hat{\alpha}} P'$ and $P' \preceq Q'$;

where $\alpha \in \text{Act}_\tau$.

It is obvious that \preceq is a preorder. Denote $\preceq \cap \preceq^{-1}$ by \approx .

It is well-known in process algebra that the above preorder is not congruent, for example, $a \preceq \tau; a$, but $a + b \not\preceq \tau; a + b$. Also, from the above example, we see that \approx is a equivalent relation, but not congruent. However, following Milner [13], we have a standard way of associating a precongruence with \preceq . [1] proved the implicit congruence associated with \preceq coincides with the preorder defined below, denoted as \preceq^* .

Definition 3 For each $P, Q \in \text{BPA}_\delta^\varepsilon$, $P \preceq^* Q$ iff

- $\forall a \in \text{Act}$, whenever $P \xrightarrow{a} P'$ then, for some $Q', Q \xrightarrow{\hat{a}} Q'$ and $P' \preceq Q'$;
- if $P \xrightarrow{\tau} P'$, then
 - (a) $\downarrow(P')$ implies, for some $Q', Q \xrightarrow{\tau} Q'$ and $P' \preceq Q'$;
 - (b) $\uparrow(P')$ implies, for some $Q', Q \xrightarrow{\varepsilon} Q'$ and $P' \preceq Q'$;
- if $\downarrow_\alpha(P)$, then
 - (a) $\downarrow_\alpha(Q)$
 - (b) whenever $Q \xrightarrow{\alpha} Q'$ then, for some $P', P \xrightarrow{\hat{\alpha}} P'$ and $P' \preceq Q'$;
- if $\downarrow(P)$, then $\mathbb{T}(P)$ iff $\mathbb{T}(Q)$;

where $\alpha \in \text{Act}_\tau$.

[1] also proved that \preceq^* can be finitely axiomatized and

Theorem 1 For any $P, Q \in \text{BPA}_\delta^\varepsilon$, $P \preceq Q$ iff $P \preceq^* Q$ or $P \preceq^* \tau; Q$ or $\tau; P \preceq^* Q$.

It is easy to see that $\tau \preceq \varepsilon$, but $\tau \not\preceq^* \varepsilon$. However, $\tau \preceq^* \tau; \varepsilon$.

In what follows, we will use \approx^* to denote $\preceq^* \cap \preceq^{*-1}$. It is easy to see that \approx^* is equivalent as well as congruent.

3. Fixpoint Logic with Chop

FLC, due to Markus Müller-Olm [14], is an extension of the modal μ -calculus with the chop “;”. Intuitively, $P \models \phi; \psi$ means that the behavior of P can be partitioned into two successive parts such that the first has the property ϕ and the second meets ψ . FLC can express non-regular properties, and is therefore strictly more powerful than the μ -calculus.

Let X, Y, Z, \dots range over an infinite set Var of variables, tt and ff be propositional constants as usual, and \surd another special propositional constant that is used to indicate if or not a process is terminated. Formulae of FLC are generated according to the following grammar:

$$\begin{aligned} \phi ::= & tt \mid ff \mid \surd \mid \tau \mid X \mid [\alpha] \mid \langle \alpha \rangle \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \\ & \phi_1; \phi_2 \mid \mu X. \phi \mid \nu X. \phi \end{aligned}$$

where $X \in \text{Var}$ and $\alpha \in \text{Act}_\tau$.

In what follows, we use $@$ to stand for $\langle \alpha \rangle$ or $[\alpha]$, p for tt, ff or \surd , and σ for \vee or μ .

Some notations can be defined as in the modal μ -calculus, for example *free* and *bound* occurrences of variables, *closed* and *open* formulae etc. The two *fix-point operators* μX and νX are treated as quantifiers. We will use $fn(\phi)$ to stand for all variables which have some free occurrence in ϕ and $bn(\phi)$ for all variables that have some bound occurrence in ϕ . Denote by $c\mathcal{FLC}$ the set of all closed formulae of FLC.

It is well-known that formulae of the modal μ -calculus are interpreted as a predicate, i.e. a set of states. However, we cannot interpret FLC in such a way because given a process P and a formula of the form $\phi; \psi$, we really do not know where and how to partition the process into two successive parts such that the first satisfies ϕ and the second meets ψ . Therefore, FLC is interpreted in the second-order, that is, formulae are interpreted as a *monotonic predicate transformer* which is a monotonic function with the type $2^{\text{BPA}_\delta^\varepsilon} \rightarrow 2^{\text{BPA}_\delta^\varepsilon}$. Here, we say that a predicate transformer f is monotonic in the sense that for any given $\mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \text{BPA}_\delta^\varepsilon$, $f(\mathcal{A}_1) \subseteq f(\mathcal{A}_2)$ ². Thus, the chop $;$ is easily interpreted as the functional composition. We use MPT to represent all the monotonic predicate transformers over $\text{BPA}_\delta^\varepsilon$.

The meaning of variables is given by a *valuation* ρ with the type $Var \rightarrow \text{MPT}$, that is, it assigns a monotonic function from sets to sets to each variable in Var . $\rho[X \rightsquigarrow f]$ agrees with ρ except for associating f with X .

Definition 4 *The meaning of a formula ϕ , under a valuation ρ , denoted by $[\phi]_\rho$, is inductively defined as follows:*

$$\begin{aligned}
[tt]_\rho(\mathcal{A}) &= \text{BPA}_\delta^\varepsilon \\
[ff]_\rho(\mathcal{A}) &= \emptyset \\
[\surd]_\rho(\mathcal{A}) &= \{P \in \text{BPA}_\delta^\varepsilon \mid \mathcal{T}(P)\} \\
[\tau]_\rho(\mathcal{A}) &= \mathcal{A} \\
[X]_\rho(\mathcal{A}) &= \rho(X)(\mathcal{A}) \\
[[\alpha]]_\rho(\mathcal{A}) &= \{P \in \text{BPA}_\delta^\varepsilon \mid \neg \mathcal{T}(P) \wedge \\
&\quad \forall P' \in \text{BPA}_\delta^\varepsilon. P \xrightarrow{\alpha} P' \Rightarrow P' \in \mathcal{A}\} \\
[\langle \alpha \rangle]_\rho(\mathcal{A}) &= \{P \in \text{BPA}_\delta^\varepsilon \mid \\
&\quad \exists P' \in \text{BPA}_\delta^\varepsilon. P \xrightarrow{\alpha} P' \wedge P' \in \mathcal{A}\} \\
[\phi_1 \wedge \phi_2]_\rho(\mathcal{A}) &= [\phi_1]_\rho(\mathcal{A}) \cap [\phi_2]_\rho(\mathcal{A}) \\
[\phi_1 \vee \phi_2]_\rho(\mathcal{A}) &= [\phi_1]_\rho(\mathcal{A}) \cup [\phi_2]_\rho(\mathcal{A}) \\
[\phi_1 ; \phi_2]_\rho &= [\phi_1]_\rho \cdot [\phi_2]_\rho \\
[\mu X. \phi]_\rho &= \sqcap \{f \in \text{MPT}_T \mid [\phi]_{\rho[X \rightsquigarrow f]} \subseteq f\} \\
[\nu X. \phi]_\rho &= \sqcup \{f \in \text{MPT}_T \mid [\phi]_{\rho[X \rightsquigarrow f]} \supseteq f\}
\end{aligned}$$

²In fact, FLC can be interpreted more generally as in [14] over a labeled transition system in which some states could not be context-free process terms.

where $\mathcal{A} \subseteq \text{BPA}_\delta^\varepsilon$, and \cdot stands for the composition operator over functions.

Note that because ε and δ have different behavior in the presence of $;$, they should be distinguished in FLC. To this end, we interpret $[\alpha]$ differently from in [14]. According to our interpretation, $P \models [\alpha]$ only if $\neg \mathcal{T}(P)$, whereas in [14] it is always valid that $P \models [\alpha]$ for any $P \in \mathcal{P}^s$. Thus, it is easy to show that $\varepsilon \not\models \bigwedge_{\alpha \in \text{Act}_\tau} [\alpha]; ff$, while $\bigwedge_{\alpha \in \text{Act}_\tau} [\alpha]; ff$ is the characteristic formula of δ up to \sim .

As the meaning of a closed formula ϕ is independent of any environment, we sometimes write $[[\phi]]$ for $[\phi]_\rho$, where ρ is an arbitrary environment. We also abuse $\phi(\mathcal{A})$ to stand for $[\phi]_\rho(\mathcal{A})$ if ρ is clear from the context.

The set of processes *satisfying* a given closed formula ϕ is $\phi(\text{BPA}_\delta^\varepsilon)$. A process P is said to satisfy ϕ iff $P \in [\phi]_\rho(\text{BPA}_\delta^\varepsilon)$ under some valuation ρ , denoted by $P \models_\rho \phi$. If ρ is clear from the context, we directly write $P \models \phi$. $\phi \Rightarrow \psi$ means that $[\phi]_\rho(\mathcal{A}) \subseteq [\psi]_\rho(\mathcal{A})$ for any $\mathcal{A} \subseteq \text{BPA}_\delta^\varepsilon$ and any ρ . $\phi \Leftrightarrow \psi$ means $(\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$. The other notations can be defined in the standard way.

[9] proved that each formula $\phi \in c\mathcal{FLC}$ is an invariant of \sim . I.e.,

Theorem 2 *Given $P, Q \in \text{BPA}_\delta^\varepsilon$, $P \sim Q$ iff for any $\phi \in c\mathcal{FLC}$, $P \models \phi$ iff $Q \models \phi$.*

In order to investigate observable behavior of systems, [18] introduce observable modalities $\langle\langle \rangle\rangle$ and $[\]$ into HML (Hennessy-Milner Logic). Formally, the meaning of $\langle\langle \rangle\rangle$ is defined as

$$[\langle\langle \rangle\rangle]_\rho(\mathcal{A}) = \{P \mid \exists P'. P \xrightarrow{\varepsilon} P' \wedge P' \in \mathcal{A}\}.$$

Dually, the meaning of $[\]$ can be given. [18] pointed out that in HML the two observable modalities are not definable, whereas the two modalities are definable in the modal μ -calculus. The following lemma will show how to define the two observable modalities in FLC.

Lemma 1

$$\begin{aligned}
(1) \quad \langle\langle \rangle\rangle &\Leftrightarrow \tau \vee (\mu X. \langle \tau \rangle; X \vee \langle \tau \rangle), \\
(2) \quad [\] &\Leftrightarrow \tau \wedge (\nu X. [\tau]; X \wedge [\tau])
\end{aligned}$$

Let $\langle\langle \alpha \rangle\rangle \triangleq \langle\langle \rangle\rangle; \langle \alpha \rangle; \langle\langle \rangle\rangle$ and $[\alpha] \triangleq [\]; [\alpha]; [\]$ for any $\alpha \in \text{Act}_\tau$. We call $\langle\langle \alpha \rangle\rangle$ and $[\alpha]$ *weak diamond* α and *weak box* α , respectively. Now, let $w\text{FLC}$ be the set of formulae generated from the grammar of FLC except that $[\alpha]$ and $\langle \alpha \rangle$ are replaced by $[\alpha]$ and $\langle\langle \alpha \rangle\rangle$, respectively. By $w\mathcal{FLC}$ denotes the set of the closed formulae of $w\text{FLC}$. It is easy to see that $w\mathcal{FLC}$ is a proper subset of $c\mathcal{FLC}$.

Similar to [9], we can show that each $\psi \in w\mathcal{FLC}$ is an invariant of \approx^* , i.e.

Theorem 3 Given $P, Q \in \text{BPA}_\delta^\varepsilon$, $P \approx^* Q$ iff for any $\psi \in w\mathcal{FLC}$, $P \models \psi$ iff $Q \models \psi$.

4. Defining the Non-deterministic Choice “+” in wFLC

Compositionality plays an important role in designing reactive systems as it allows one to compose/decompose a complex system from/to several simpler components. Generally speaking, it is hard to design a complex system in a logical frame in a compositional way because it is difficult to find a connection between the structure of a system to be developed and that of its specification given by the logic. In order to describe properties of non-deterministic programs with logical approach in a compositional way, [6, 12] introduced the non-deterministic choice “+” of process algebras into modal logics, and established Synchronization Tree Logic (STL) and Modal Process Logic, respectively. Intuitively, $P \models \phi + \psi$ means that there exist P_1 and P_2 such that $P \sim P_1 + P_2$, $P_1 \models \phi$ and $P_2 \models \psi$.

[20] showed that the non-deterministic choice “+” can be defined by disjunction and conjunction in the modal μ -calculus. [21] extended the results of [20] to FLC by showing that “+” is also definable in FLC.

In what follows, we first briefly review the basic idea of [21] on how to derive “+” by conjunction and disjunction, and then prove that as far as wFLC is concerned, “+” is also definable.

First step, as in [6, 12], is directly to introduce “+” into FLC, written the extension as FLC^+ , formally defined as:

$$\phi ::= p \mid \tau \mid X \mid @ \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 ; \phi_2 \mid \phi_1 + \phi_2 \mid \mu X. \phi \mid \nu X. \phi$$

where $X \in \text{Var}$ and $\alpha \in \text{Act}_\tau$.

Given a valuation ρ , the meaning of $\phi + \psi$ is defined as

$$[\phi_1 + \phi_2]_\rho(\mathcal{A}) = \{P \in \text{BPA}_\delta^\varepsilon \mid P \sim P_1 + P_2 \wedge P_1 \in [\phi_1]_\rho(\mathcal{A}) \wedge P_2 \in [\phi_2]_\rho(\mathcal{A})\} \quad (1)$$

The other constructs are interpreted as in FLC.

In the following, we define what it means for a formula to be a *guard*:

Definition 5 1. $@$ and p are guards;

2. if ϕ and ψ are guards, so are $\phi \wedge \psi$, $\phi \vee \psi$ and $\phi + \psi$;

3. if ϕ is a guard, so are $\phi ; \psi$ and $\sigma X. \phi$, where ψ is any formula of FLC^+ .

X is said to be *guarded* in ϕ if each occurrence of X is within a subformula ψ that is a guard. If all variables in $\text{fn}(\phi) \cup \text{bn}(\phi)$ are guarded, then ϕ is called *guarded*. A formula ϕ is said to be *strictly guarded* if ϕ is guarded and for any $X \in \text{fn}(\phi) \cup \text{bn}(\phi)$, there does not exist a subformula of the forms $X + \psi$, $(X \odot \chi) + \psi$, $(X ; \varphi) + \chi$ or $(X ; \varphi \odot \chi) + \psi$, where $\odot \in \{\vee, \wedge\}$.

Intuitively, a variable X is said to be *guarded* means that each occurrence of X is within the scope of a modality $@$ or a propositional letter p .

Example 1 Formulae $\langle a \rangle ; X ; Y ; \nu X. (\langle a \rangle \vee \langle b \rangle) ; X ; (Y + Z) ; ff ; X$ are guarded, but $X ; \langle a \rangle \wedge X ; \mu X. (X + Y) \vee [a] ; \mu X. (\langle a \rangle ; X \vee \langle b \rangle) ; \mu Y. (Y + \langle a \rangle)$ are not. $\langle a \rangle ; X ; Y$ and $ff ; X$ are strictly guarded, however, $\nu X. (\langle a \rangle \vee \langle b \rangle) ; X ; (Y + Z)$ is not.

We will use $\mathcal{L}_{\text{FLC}^+}$ to denote all formulae of FLC^+ that are closed and guarded, and \mathcal{L}_{FLC} for the subset of \mathcal{FLC} in which all formulae are guarded.

Given a formula ϕ , the set of the atomic subformulae *at the end* of ϕ , denoted by $\text{ESub}(\phi)$, is: $\{\phi\}$ if $\phi = p, \tau, X$ or $@$; $\text{ESub}(\phi_1) \cup \text{ESub}(\phi_2)$ if $\phi = \phi_1 \text{ op } \phi_2$ where $\text{op} \in \{\wedge, \vee, +\}$; if $\phi = \phi_1 ; \phi_2$ then if $\tau \notin \text{ESub}(\phi_1)$ then $\text{ESub}(\phi_2)$ else $(\text{ESub}(\phi_2) \setminus \{\tau\}) \cup \text{ESub}(\phi_1)$; $\text{ESub}(\phi')$ if $\phi = \sigma X. \phi'$. It is said that $\sqrt{\quad}$ only occurs at the end of ϕ if $\sqrt{\quad}$ can only be in $\text{ESub}(\phi)$ as a sub-formula of ϕ .

Second step is to show that for any $\phi \in \mathcal{L}_{\text{FLC}^+}$, there exists a formula $\phi' \in \mathcal{L}_{\text{FLC}}$ such that $\phi \Leftrightarrow \phi'$. This can be obtained via the following three steps. The first is to show that in some special cases “+” can be defined essentially by conjunction and disjunction. Then, to prove that the elimination of “+” in a strictly guarded formula ϕ of FLC^+ can be reduced to one of the above special cases. Finally, to complete the proof by showing that for any $\phi \in \mathcal{L}_{\text{FLC}^+}$ there exists a strictly guarded formula $\phi' \in \mathcal{L}_{\text{FLC}}$ such that $\phi \Leftrightarrow \phi'$. Details can be referred to [21].

Thus, the definability of + in FLC can be represented as the following theorem.

Theorem 4 For any $\phi \in \mathcal{L}_{\text{FLC}^+}$, there exists $\phi' \in \mathcal{L}_{\text{FLC}}$ such that $\phi' \Leftrightarrow \phi$.

In the definition of FLC^+ , if we replace $[\alpha]$ and $\langle \alpha \rangle$ with $[a]$ and $\langle a \rangle$ respectively, then the resulted logic will be written as wFLC^+ . We use $w\mathcal{FLC}^+$ to stand for the set of closed formulae of wFLC^+ . If we replace $[\alpha]$ and $\langle \alpha \rangle$ with $[a]$ and $\langle a \rangle$, respectively in Definition 5, we obtain the definition of *guardness*

for $w\text{FLC}^+$. Consequently, $w\mathcal{L}_{\text{FLC}}$ stands for the subset of $w\mathcal{FLC}$ in which all formulae are guarded, and $w\mathcal{L}_{\text{FLC}^+}$ for the subset of \mathcal{FLC}^+ in which all formulae are guarded.

Then, we can also claim that in $w\mathcal{L}_{\text{FLC}^+}$, “+” can be eliminated, i.e.

Theorem 5 *For any $\phi \in w\mathcal{L}_{\text{FLC}^+}$, there exists $\phi' \in w\mathcal{L}_{\text{FLC}}$ such that $\phi' \Leftrightarrow \phi$.*

Proof: We can prove this theorem with two approaches. The first one is to show it similar to the proof of Theorem 4. The other one is to prove it directly from Theorem 4 and the definitions of $[a]$ and $\langle\langle a \rangle\rangle$. \dashv

In what follows, we will use $en(\phi)$ to denote the resulting formula by applying the above procedure to ϕ by which + is eliminated.

5. Relating $\text{BPA}_\delta^\varepsilon$ to $w\text{FLC}^+$

[21] showed a connection between $\text{BPA}_\delta^\varepsilon$ and $\mathcal{L}_{\text{FLC}^+}$ under the strong bisimulation semantics, and exploited the connection to construct the characteristic formula of a process in $\text{BPA}_\delta^\varepsilon$ up to \sim syntactically. In this section, we will show a similar connection between $\text{BPA}_\delta^\varepsilon$ and $w\mathcal{L}_{\text{FLC}}$ under the observable semantics, which will be used to present an algorithm to derive the characteristic formula of a context-free process up to \preceq^* syntactically in the next section.

Nondeterminism

From the semantics of FLC^+ and Theorem 5, it is clear that “+” of $\text{BPA}_\delta^\varepsilon$ corresponds to the derived operator “+” in $w\mathcal{L}_{\text{FLC}^+}$. That is,

Proposition 1 *For any $P, Q \in \text{BPA}_\delta^\varepsilon$ and $\phi, \psi \in w\mathcal{L}_{\text{FLC}^+}$, if $P \models \phi$ and $Q \models \psi$ then $P + Q \models en(\phi + \psi)$.*

Sequential Composition

Similarly to [21], we can obtain a connection between the sequential composition “;” of $\text{BPA}_\delta^\varepsilon$ and the chop “;” of $w\mathcal{L}_{\text{FLC}^+}$ as follows.

Theorem 6 *For any $\phi, \psi \in w\mathcal{L}_{\text{FLC}^+}$ and any $P, Q \in \text{BPA}_\delta^\varepsilon$, if \surd only occurs at the end of ϕ , $P \models \phi$; \surd and $Q \models \psi$ then $P; Q \models \phi \{ \tau / \surd \}; \psi$.*

Recursion

Here, we show how to relate $rec\ x$ to νX and μX similarly to [21]. To this end, we first employ a relation called *weak syntactical confirmation* between processes

and formulae, with the type $\mathcal{P}^s \times w\text{FLC}^+ \mapsto \{\text{tt}, \text{ff}\}$, denoted by \models_{wsc} .

Definition 6 *Given a formula ϕ , we associate a map from $2^{\mathcal{P}^s}$ to $2^{\mathcal{P}^s}$ with it, denoted by $\tilde{\phi}$, constructed by the following rules:*

$$\begin{aligned} \tilde{\surd}(\mathcal{E}) &\hat{=} \{E \in \mathcal{P}^s \mid E \approx^* \varepsilon\} \\ \tilde{\text{tt}}(\mathcal{E}) &\hat{=} \mathcal{P}^s \\ \tilde{\text{ff}}(\mathcal{E}) &\hat{=} \emptyset \\ \tilde{\tau}(\mathcal{E}) &\hat{=} \mathcal{E} \\ \tilde{X}(\mathcal{E}) &\hat{=} \{x; \tau^n; E \mid E \in \mathcal{E}, 0 \leq n\} \\ \tilde{\langle\langle \alpha \rangle\rangle}(\mathcal{E}) &\hat{=} \{E \mid \exists E' \in \mathcal{E}. E \xrightarrow{\alpha} E'\} \\ \tilde{[a]}(\mathcal{E}) &\hat{=} \{E \mid \neg \mathbb{T}(E) \wedge E \text{ is guarded} \wedge \\ &\quad \forall E'. E \xrightarrow{\alpha} E' \Rightarrow E' \in \mathcal{E}\} \\ \tilde{\phi_1 \wedge \phi_2}(\mathcal{E}) &\hat{=} \tilde{\phi_1}(\mathcal{E}) \cap \tilde{\phi_2}(\mathcal{E}) \\ \tilde{\phi_1 \vee \phi_2}(\mathcal{E}) &\hat{=} \tilde{\phi_1}(\mathcal{E}) \cup \tilde{\phi_2}(\mathcal{E}) \\ \tilde{\phi_1 + \phi_2}(\mathcal{E}) &\hat{=} \{E \mid \exists E_1, E_2. E = E_1 + E_2 \wedge \\ &\quad E_1 \in \tilde{\phi_1}(\mathcal{E}) \wedge E_2 \in \tilde{\phi_2}(\mathcal{E})\} \\ \tilde{\phi_1; \phi_2}(\mathcal{E}) &\hat{=} \tilde{\phi_1} \cdot \tilde{\phi_2}(\mathcal{E}) \\ \tilde{\sigma X. \phi}(\mathcal{E}) &\hat{=} \{(rec\ x.E_1); E_2 \mid E_1 \in \tilde{\phi}(\{\varepsilon\}) \wedge E_2 \in \mathcal{E}\} \end{aligned}$$

where $\alpha \in \text{Act}_\tau$, $\mathcal{E} \subseteq \mathcal{P}^s$.

$\models_{wsc}(E, \phi) = \text{tt}$ iff $E \in \tilde{\phi}(\{\varepsilon\})$; otherwise, $\models_{wsc}(E, \phi) = \text{ff}$. In what follows, we denote $\models_{wsc}(E, \phi) = \text{tt}$ by $E \models_{wsc} \phi$ and $\models_{wsc}(E, \phi) = \text{ff}$ by $E \not\models_{wsc} \phi$.

Informally, $P \models_{wsc} \phi$ means that P and ϕ have a similar syntax in the sense that all occurrences of the τ action in P are abstracted away. Comparing with the notion of *syntactical confirmation* in [21], the clauses of \surd , x , $\langle\langle a \rangle\rangle$, and $[a]$ are very different.

Example 2 *Let $E_0 \hat{=} rec\ x. \tau; x + \tau$, $E_1 \hat{=} (\tau; \tau; a; x; x) + d; \tau$, $E_2 \hat{=} x; (b; \tau + c); \tau; y; \tau$ and $E_3 \hat{=} E_0; a; b; c$. While, let $\phi_0 \hat{=} \surd$, $\phi_1 \hat{=} \langle\langle a \rangle\rangle; X; X$, $\phi_2 \hat{=} X; \langle\langle b \rangle\rangle; Y$ and $\phi_3 \hat{=} [a]; \langle\langle b \rangle\rangle; \langle\langle c \rangle\rangle$. According to the above definition, it follows that $E_0 \models_{wsc} \phi_0$, $E_1 \models_{sc} \phi_1$, $E_2 \models_{sc} \phi_2$, $E_3 \models_{sc} \phi_3$.*

The following lemma indicates that \models_{wsc} itself is compositional as well.

Theorem 7 *Let \surd only appear at the end of ϕ_1 , ϕ_2 and ϕ . Then,*

- i) *if $E_1 \models_{wsc} \phi_1$ and $E_2 \models_{wsc} \phi_2$ then $E_1 + E_2 \models_{wsc} \phi_1 + \phi_2$;*
- ii) *if $E_1 \models_{wsc} \phi_1$ and $E_2 \models_{wsc} \phi_2$ then $E_1; E_2 \models_{wsc} \phi_1 \{ \tau / \surd \}; \phi_2$;*

iii) if $E \models_{wsc} \phi$ then $rec\ x.E \models_{wsc} \sigma X.\phi\{\tau/\sqrt{\cdot}\}$.

Example 3 In Example 2, according to Theorem 7, we obtain $E_1 + E_2 \models_{wsc} \phi_1 + \phi_2$, $E_3; (E_1 + E_2) \models_{wsc} \phi_3; (\phi_1 + \phi_2)$ and $rec\ x. rec\ y.E_3; (E_1 + E_3) \models_{wsc} \nu X.\nu Y.(\phi_3; (\phi_1 + \phi_2))$.

Theorem 8 establishes a connection between \models_{wsc} and \models , so that $rec\ x$ is related to νX and μX .

Theorem 8 For any $P \in BPA_{\delta}^{\varepsilon}$ $\phi \in w\mathcal{L}_{FLC}^+$, if $\sqrt{\cdot}$ only occurs at the end of ϕ , $\phi \not\Rightarrow ff$ and $P \models_{wsc} \phi$, then $P \models \phi; \sqrt{\cdot}$.

6. Constructing Characteristic Formulae for Context-free Processes Syntactically

Given a binary relation \mathcal{R} over processes, which may be an equivalence or a preorder, the characteristic formula for a process P up to \mathcal{R} is a formula ϕ_P such that for any process Q , $Q \models \phi_P$ if and only if $P \mathcal{R} Q$. [14] presented a method to derive the characteristic formula for a context-free process up to strong bisimulation by solving the equation system induced by the rewrite system of the process, while [21] investigated the issue from a syntactical point of view, constructing the characteristic formula directly from the syntax of the process. Here, we will extend the method proposed in [21] to construct the characteristic formula for a process of $BPA_{\delta}^{\varepsilon}$ up to \preceq^* also directly from its syntax.

Let $[\downarrow] = \mu X.\tau \wedge [\tau]; X$ and $\langle\langle \uparrow \rangle\rangle = \nu X.\langle\langle \tau \rangle\rangle; X$. The former formula says that any process that meets the formula must be convergent, that is, the process cannot perform an infinite sequence of unobservable actions; whereas the latter expresses that any process that has the property may potentially perform an infinite sequence of unobservable actions, i.e. divergent. The contrast in meaning between $[\downarrow]$ and $[\uparrow]$ is the difference in their fixpoints. It is clear that a divergent process has the property $[\uparrow]$, but cannot have $[\downarrow]$.

In what follows, let us discuss how to characterize the primitive constructs of $BPA_{\delta}^{\varepsilon}$ up to the preorder \preceq^* .

So that the characteristic formula of a composite process may be built from those of the primitives according to its syntactical structure.

For simplicity, $\bigwedge_{\alpha \in Act} \tau \rightarrow A [\alpha]; ff$ will be abbreviated as Φ_{-A} from now on.

Firstly, we consider the characteristic formulae of δ . It is obvious that for any process Q , if $\delta \preceq^* Q$, then Q should have the following properties: the first is that Q cannot do any observable action; the second

³It is worth noting that characteristic formulae up to \preceq^* for these primitive constructs like δ and ε are more complicated than the ones up to \sim .

is that Q cannot perform infinite many unobservable actions; and the last is that Q cannot terminate. Thus, the characteristic formula of δ up to \preceq^* can be defined as $\Phi_{-\{\tau\}} \wedge [\downarrow]$ (Φ_{δ} for short). Notice that if we see δ as an abbreviation of $rec\ x.x$, then $\delta \models_{wsc} \Phi_{\delta}$ according to Definition 6.

For any process Q with $\varepsilon \preceq^* Q$, it follows that on one hand, it cannot execute any observable action; on the other hand, it may perform a finite sequence of unobservable actions, but any of such executions must terminate. Therefore, ε can be characterized by $\Phi_{-\{\tau\}} \wedge [\downarrow]; \langle\langle \rangle\rangle; \sqrt{\cdot}$, written as Φ_{ε} . Note that Φ_{ε} guarantees that the process is convergent. Obviously, $\varepsilon \models_{wsc} \Phi_{\varepsilon}$.

As for an action $a \in Act$, for any process Q with $a \preceq^* Q$, then Q should have the properties: firstly, it can only perform the action a and then evolves to ε ; secondly, it may perform any finite many unobservable actions before and after executing a , but cannot diverge. Thus, let $\Phi_a = \Phi_{-\{a, \tau\}} \wedge [\downarrow] \wedge (\langle\langle \alpha \rangle\rangle \wedge [\alpha])$, then we can define the characteristic formula of α as $\Phi_a; \sqrt{\cdot}$. It also follows that $\alpha \models_{wsc} \Phi_a$ by Definition 6.

Similarly, as for τ , for any process Q with $\tau \preceq^* Q$, then Q should have the properties: firstly, it cannot perform any observable action; secondly, it at least perform one unobservable action, but can only perform a finite sequence of unobservable actions and any of such executions must terminate. If let $\Phi_{\tau} = \Phi_{-\{\tau\}} \wedge [\downarrow] \wedge (\langle\langle \tau \rangle\rangle \wedge [\tau])$, then we can define the characteristic formula of τ as $\Phi_{\tau}; \sqrt{\cdot}$. It is easy to obtain that $\tau \models_{wsc} \Phi_{\tau}$ by Definition 6.

According to the definition of \preceq^* , on one hand, if τ appearing at the beginning of a process, it behaves like an action in Act ; on the other hand, if it follows an action a via $;$, it can be discarded according to $P; \tau \approx^* P$ where P is a process. In order to deal with such a problem, we introduce a special propositional letter EL . Then present a rule on how to eliminate EL in the derived formula in the procedure all information concerning the redundant τ actions will be discarded in order to keep $P; \tau \approx^* P$.

Since the recursive operator may introduce divergence, we define the characteristic formula according to if or not it gives rise to divergence that can be determined by checking the weak syntactical confirmation between the resulted process term and $\langle\langle \uparrow \rangle\rangle$. It is obvious that the characteristic formula for a divergent processes Ω up to \preceq^* is tt because $\Omega \preceq^* P$ for any $P \in BPA_{\delta}^{\varepsilon}$. On the other hand, it is well-known that if a process E is divergent then $E + F$ and $F + E$ for any process F will be divergent also. Therefore, the characteristic formulae of $E + F$ and $F + E$ are tt as well if the one of E is tt .

In a word, given a process term $E \in \mathcal{P}^s$, we

use the following algorithm to associate a formula of $wFLC^+$ with EL with E according to its syntax.

Definition 7 Given a process term $E \in \mathcal{P}^s$, we associate with it a formula of $wFLC^+$, denoted by Ψ_E , constructed by the following rules:

$$\begin{aligned} \Psi_\delta &\hat{=} \Phi_\delta, \\ \Psi_\varepsilon &\hat{=} \Phi_\varepsilon, \\ \Psi_x &\hat{=} X, \\ \Psi_\alpha &\hat{=} \Phi_\alpha \quad \text{for } \alpha \in Act_\tau, \\ \Psi_{E_1;E_2} &\hat{=} \Psi_{E_1}\{\tau/\sqrt{\cdot}\};EL;\Psi_{E_2}, \\ \Psi_{E_1+E_2} &\hat{=} \begin{cases} tt & \text{if } \Psi_{E_1} = tt \text{ or } \Psi_{E_2} = tt \\ \Psi_{E_1} + \Psi_{E_2} & \text{otherwise,} \end{cases} \\ \Psi_{rec.x.E} &\hat{=} \begin{cases} tt & \text{if } rec.x.E \models_{wsc} \langle\langle \uparrow \rangle\rangle \\ \forall X. \Psi_E\{\tau/\sqrt{\cdot}\} & \text{otherwise} \end{cases} \end{aligned}$$

Regards Definition 7, we have

Lemma 2 1. For any $E \in \mathcal{P}^s$, $\sqrt{\cdot}$ only occurs at the end of Ψ_E ;

2. For any $E \in \mathcal{P}^s$, $E \models_{wsc} \Psi_E\{\tau/EL\}$ and $E \models_{wsc} \Psi_E\{\tau/EL\};\sqrt{\cdot}$;

3. For any $P \in BPA_\delta^\varepsilon$, $\Psi_P\{\tau/EL\};\sqrt{\cdot} \in w\mathcal{L}_{FLC^+}$.

In order to derive the characteristic formula of a process P , we define a rule to eliminate EL as follows:

$$EL;\phi = \begin{cases} EL & \text{if } \phi = EL, [\tau] \text{ or } \langle\langle \tau \rangle\rangle; \\ \phi & \text{otherwise.} \end{cases}$$

It is easy to see that by applying the rule to $en(\Psi_P;\sqrt{\cdot})$, we can get a EL -free formula and denote by Υ_P .

Similar to [21], we can show that Υ_P is the characteristic formula of P up to \preceq^* for each $P \in BPA_\delta^\varepsilon$.

Theorem 9 For any $P \in BPA_\delta^\varepsilon$, if $Q \models \Upsilon_P$ then $P \preceq^* Q$.

Remarks 1 In Theorem 9, the condition that P is guarded is essential. Otherwise, the theorem is not true any more. For instance, $\forall X.(X + (\langle\langle a \rangle\rangle \wedge [a][\downarrow] \wedge \Phi_{-\{a\}}))$ is equivalent to $\Psi_{rec.x.(x+a)}$, nevertheless, $(\forall X.(X + (\langle\langle a \rangle\rangle \wedge [a] \wedge \Phi_{-\{a\}})))\sqrt{\cdot}$ is not the characteristic formula of $rec.x.(x+a)$, since $rec.x.(x+b+a)$ meets the formula, but $rec.x.(x+a) \not\preceq^* rec.x.(x+b+a)$.

Besides, utilizing the results of [1] that \preceq^* can be finitely axiomatized, we can show that

Theorem 10 • For any $P, Q \in BPA_\delta^\varepsilon$, if $P \preceq^* Q$ then $\Upsilon_Q \Rightarrow \Upsilon_P$.

• For any $P, Q \in BPA_\delta^\varepsilon$, if $P \approx^* Q$ then $\Upsilon_Q \Leftrightarrow \Upsilon_P$.

According to our definition, it is easy to see that the characteristic formulae of $rec.x.\tau;x$ is tt and therefore $rec.x.\tau;x \preceq^* Q$ for any $Q \in BPA_\delta^\varepsilon$. Moreover, the characteristic formula of $rec.x.\tau;x + \tau$ is also tt as $rec.x.\tau;x + \tau \models_{wsc} \langle\langle \uparrow \rangle\rangle$.

7. Concluding Remarks

In this paper, we extended the result of [21] by considering the compositionality of the observable FLC. To the end, similarly to [21] we first proved the definability of the non-deterministic choice “+” in the observable FLC and then established a connection between the observable FLC and BPA_δ^ε under the observational semantics \preceq . Furthermore, we also give an algorithm on constructing characteristic formula of a context-free process up to the preorder \preceq^* directly from its syntax.

The significance of this work on developing high reliable software is obvious. By the work of this paper and [21], some connections between algebraical approach and logical approach have been established. As we know, BPA_δ^ε is the basis of process algebras, and most of modal and temporal logics can be reduce to FLC. Therefore, by relating the constructs of BPA_δ^ε to the connectives of FLC, we can obtain the compositionality of modal logics. Thus developing a complex system with modal logics can be done in process algebra-like compositional manner. Compositionality is very important in developing high reliable software for at least the following reasons. Firstly, it allows modular design and verification of complex systems so that the complexity is tractable. Secondly, during re-designing a verified system only the verification concerning the modified parts should be re-done rather than verifying the whole system from scratch. Thirdly, compositionality makes it possible to partially specify a large system. When designing a system or synthesizing a process, it is possible to have undefined parts of a process and still to be able to reason about it. For example, this technique can be applied for revealing inconsistencies in the specification or proving that with the choices already taken in the design no component supplied for the missing parts will ever be able to make the overall system satisfy the original specification. Finally, it can make possible the reuse of verified components; their previous verification can be used to show that they meet the requirements on the components of a large system.

On the other hand, through constructing the characteristic formulae up to different semantics, we can reduce lots of verification problems in an algebraical frame accordingly into a logical frame. In fact, many of these problems cannot be well solved in algebraical frame, but the corresponding problems in a logical

frame may be solvable. For instance, in algebraical frame, verifying refinement (implementation) relation between different abstract-level systems that is normally defined as a equivalence or preorder over processes is impractical in general. However, by reducing such a problem into a logical frame, we could utilize the proof system and corresponding proof assistants of the logic to handle the problem well.

References

- [1] L. Aceto and M. Hennessy. Termination, deadlock, and divergence. *Journal of ACM*, Vol. 39, No.1:147-187. January, 1992.
- [2] H. Barringer, R. Kuiper, A. Pnueli. Now you may compose temporal logic specifications. In Proc. 16th STOC, pp. 51-63. 1984.
- [3] H. Barringer, R. Kuiper, A. Pnueli. A compositional temporal approach to a CSP-like language. In Proc. IFIP conference, The Role of Abstract Models in Information Processing, pp. 207-227. 1985.
- [4] J.A. Bergstra and J.W. Klop. Algebra of communication processes with abstraction. *Theoretical Computer Science*, 37:77-121. 1985.
- [5] R. Gorrieri and A. Rensink. Action refinement. *Handbook of Process Algebra*, Elsevier Science, 1047-1147. 2001.
- [6] S. Graf and J. Sifakis. A logic for the description of non-deterministic programs and their properties. *Information and Control*, 68:254-270. 1986.
- [7] S. Graf and J. Sifakis. A modal characterization of observational congruence on finite terms of CCS. *Information and Control*, 68:125-145. 1986.
- [8] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333-354. 1983.
- [9] M. Lange and C. Stirling. Model checking fixed point logic with chop. FOSSACS'02, LNCS 2303, pp. 250-263.
- [10] K.G. Larsen and X.X. Liu. Compositionality through an operational semantics of contexts. ICALP'90, LNCS 443.
- [11] X.X. Liu. *Specification and Decomposition in Concurrency*. Ph.D. Thesis, Department of Mathematics and Computer Science, Aalborg University Center, Demark, 1992.
- [12] K.G. Larsen and B. Thomsen. A modal process logic. LICS'88, pp.203-210. 1988.
- [13] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [14] M. Müller-Olm. A Modal Fixpoint Logic with Chop. STACS'99, LNCS 1563, pp. 510-520.
- [15] A. Pnueli. The temporal logic of programs. In the Proc. of 18th STOC, pp:232-239. 1977.
- [16] R. Rosner and A. Pnueli. A choppy logic. In the proc. of LICS'86, pp.306-313. IEEE Computer Science Society, 1986.
- [17] B. Steffen. Characteristic formulae, LNCS 372, pp. 723-732. Springer-Verlag, 1989.
- [18] C. Stirling. *Modal and Temporal Properties of Processes*, Spriner, 2001.
- [19] M. Viswanathan and R. Viswanathan. Foundations for circular compositional reasoning. ICALP'01, LNCS 2076, pp. 835-847, 2001.
- [20] N. Zhan and M. Majster-Cederbaum. Deriving nondeterminism from conjunction and disjunction. FORTE'05, LNCS 3731, pp. 351-365. 2005.
- [21] N. Zhan and J. Wu. Compositionality of fixpoint logic with chop. ICTAC'05, LNCS 3722, pp. 136-150. 2005.