Compositionality of Fixpoint Logic with Chop

Naijun Zhan*[1] and Jinzhao Wu[2]

[1] Lab. of Computer Science, Institute of Software,
Chinese Academy of Sciences, 100080, Beijing, P.R. China
[2] Lehrstuhl für Praktische Informatik II, Fakultät für Mathematik und Informatik,
Universität Mannheim, D7,27, 68163, Mannheim, Deutschland

Abstract. 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 this paper, we investigate the compositionality of the Fixpoint Logic with Chop (FLC for short). To this end, we extend FLC with the nondeterministic choice "+" (FLC^+ for the extension) and then establish a correspondence between the logic and the basic process algebra with deadlock and termination (abbreviated $\text{BPA}_\delta^\epsilon$). Subsequently, we show that the choice "+" is definable in FLC.
As an application of the compositionality of FLC, an algorithm is given to construct characteristic formulae of $\text{BPA}_\delta^\epsilon$ up to strong bisimulation directly from the syntax of processes in a compositional manner.

Key words: FLC, compositionality, verification, bisimulation, characteristic formula, basic process algebra

1 Introduction

As argued in [2], compositionality is very important in developing reactive systems 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;

* This work is supported in part by CNSF-60493200 and CNSF-60421001.

their previous verification can be used to show that they meet the requirements on the components of a large system.

The μ-calculus [15] is a popular modal logic as most of modal and temporal logics can be defined in it. However, [8] proved that only "regular" properties can be defined in the μ-calculus, meanwhile [14] proved that all bisimulation invariant properties of Monadic Second Order Logic can be defined in the modal μ-calculus. In order to specify non-regular properties, [21] extended the μ-calculus with the chop operator (denoted by ";"). It seems that the chop operator ";" was first introduced in process logics [12, 6], then adopted as the unique primitive modality in interval-based logics, see [11, 28, 7], for example. In an interval-based logic, it is easy to interpret a formula like $\phi; \psi$ by partitioning the given interval into two parts such that ϕ is satisfied in the first segment and ψ is held in the second one. But it is hard to interpret the operator in modal logics. Therefore, in [21] the meaning of FLC is interpreted in second-order. [21] proved that FLC is strictly more expressive than the μ-calculus as non-regular properties can be expressed in FLC by showing that characteristic formulae of context-free processes can be defined in FLC. Since then, FLC has attracted more attentions in computer science because of its expressiveness. For example, [16, 17] investigated the issues of FLC model checking on finite-state processes.

Let us assume a setting in which the behavior of systems are modeled by some process algebra and behavioral properties of systems are specified by some specification logic. In order to exploit the compositionality inherent in the process algebra it is desirable to be able to mimic the process algebra operators in the logic (see [10]). That is, for any program constructor *cons* there should be an operator **cons** of the logic such that

(a) $P_i \models \phi_i$ for $i = 1, \cdots, n$ implies $cons(P_1, \cdots, P_n) \models \mathbf{cons}(\phi_1, \cdots, \phi_n)$;
(b) $cons(P_1, \cdots, P_n) \models \mathbf{cons}(\phi_1, \cdots, \phi_n)$ is the strongest assertion which can be deduced from $P_i \models \phi_i$ for $i = 1, \cdots, n$.

It seems that FLC does not meet the above conditions. For example, the $+$ operator of process algebra has no counterpart in FLC and in addition it is still an open problem if it is possible to derive a property from ϕ and ψ that holds in $P + Q$ in FLC, where $P \models \phi$ and $Q \models \psi$.

To achieve the goal, we first introduce the non-deterministic choice "+" that was proposed in [10, 18] as a primitive and denote the extension of FLC by FLC$^+$. 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 $P_2 \models \psi$. Thus, it is easy to see that we can use $\phi + \psi$ as a specification for the combined system $P + Q$. Then we show that the constructors of the basic process algebra with termination and deadlock (BPA$_\delta^\epsilon$ for short) correspond to the connectives of FLC$^+$. Subsequently, we prove that the choice "+" can be defined essentially by conjunction and disjunction in FLC.

As a result, we can use FLC to specify systems modeled by BPA$_\delta^\epsilon$ in an algebraical way, typically, this may allow much more concise descriptions of concurrent systems and more easy composing/decomposing the verification of a large systems from/to some similar and simpler ones of the subsystems. As an example, we now show that using “+” as an auxiliary operator could make senses in practice:

i) It means one more step to the goal to exploit the structure of process terms for model checking.
ii) It enables a precise and compact specification of certain nondeterministic systems.
iii) It is very easy to modify the specification of a system when additional alternatives for the behavior of the system should be admitted.
iv) It enhances the possibility of modularity in model checking which is useful in redesigning of systems.

i) depends on if it is possible to work out a syntax-directed model checker for FLC on finite-state processes. In fact, we believe that it may be done exploiting the connection between FLC^{+} and $\mathrm{BPA}^{\epsilon}_{\delta}$ that is presented in this paper. To explain the issues ii), iii) and iv), we present the following example: Consider a car factory that wants to establish an assembly line shown in the Fig. 1.,

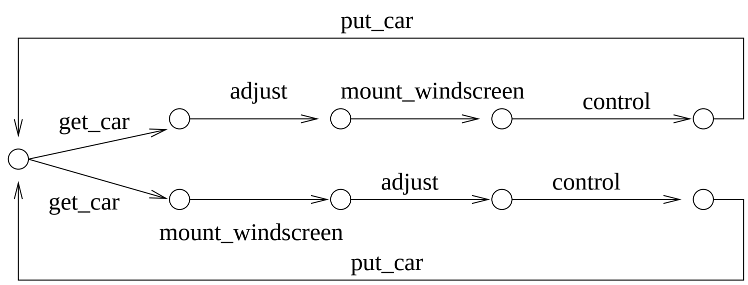

Fig. 1. The Process P

which we denote by the process P, for one production step. If there is a car available for P then P will either get the car, adjust the motor, mount the windscreen, control the car, and then put the car on the conveyer belt or P will get the car, mount the windscreen, adjust the motor, control the car, and then put it back. Afterwards P may start again. The first option can be specified by

$$\begin{aligned}\mathrm{Spec}_1 \mathrel{\widehat{=}} & [\mathrm{get\_car}];\langle\mathrm{adjust}\rangle;\langle\mathrm{mount\_windscreen}\rangle;\langle\mathrm{control}\rangle;\langle\mathrm{put\_car}\rangle\\ & \wedge\langle\mathrm{get\_car}\rangle;\mathrm{true},\end{aligned}$$

whereas the second is described by

$$\begin{aligned}\mathrm{Spec}_2 \mathrel{\widehat{=}} & [\mathrm{get\_car}];\langle\mathrm{mount\_windscreen}\rangle;\langle\mathrm{adjust}\rangle;\langle\mathrm{control}\rangle;\langle\mathrm{put\_car}\rangle\\ & \wedge\langle\mathrm{get\_car}\rangle;\mathrm{true}.\end{aligned}$$

We are now looking for a specification that admits only such systems that offer both alternatives and that can be easily constructed from Spec_1 and Spec_2. Obviously, $\mathrm{Spec}_1 \wedge \mathrm{Spec}_2$ is not suitable whereas $\mathrm{Spec}_1 \vee \mathrm{Spec}_2$ allows for implementations that exhibit only one of the behavior. $\mathrm{Spec}_1 + \mathrm{Spec}_2$ describes the behavior we have in mind and a system that offers this behavior repeatedly is described by Spec $\widehat{=}$ $\nu X.(\mathrm{Spec}_1 + \mathrm{Spec}_2); X$.

It is easy to show that $rec\, x.(P_1 + P_2); x \models \mathrm{Spec}$, where

$$\mathrm{P}_1 \mathrel{\widehat{=}} \mathrm{get\_car}; \mathrm{adjust}; \mathrm{mount\_windscreen}; \mathrm{control}; \mathrm{put\_car}$$

$$\mathrm{P}_2 \mathrel{\widehat{=}} \mathrm{get\_car}; \mathrm{mount\_windscreen}; \mathrm{adjust}; \mathrm{control}; \mathrm{put\_car}.$$

Let us now assume that the system specification should be modified to allow for a third alternative behavior Spec_3, then this specification may be simply "added" to form

$$\mathrm{Spec}' \mathrel{\widehat{=}} \nu X.(\mathrm{Spec}_1 + \mathrm{Spec}_2 + \mathrm{Spec}_3); X.$$

If we establish $P_3 \models \mathrm{Spec}_3$ then we obtain immediately that

$$rec\, x.(P_1 + P_2 + P_3); x \models \mathrm{Spec}'.$$

In addition, if we have to modify Spec_1 to Spec_1' such that $P_1' \models \mathrm{Spec}_1'$, and obtain

$$rec\, x.(P_1' + P_2 + P_3); x \models \nu X.(\mathrm{Spec}_1' + \mathrm{Spec}_2 + \mathrm{Spec}_3); X.$$

Some preliminary results of this paper have been reported in [27].

The remainder of this paper is structured as follows: Section 2 briefly reviews $\mathrm{BPA}_\delta^\epsilon$. In Section 3, FLC^+ is established and some preliminary results are given. Section 4 establishes a connection between the constructors of $\mathrm{BPA}_\delta^\epsilon$ and the connectives of FLC^+ . Section 5 is devoted to showing that the choice "+" can be defined in FLC. In Section 6, we sketch how to construct a formula Ψ_P for each process $P \in \mathrm{BPA}_\delta^\epsilon$ according to its syntax and then show the formula obtained by eliminating "+" in Ψ_P is the characteristic formula of P. Finally, a brief conclusion is provided in Section 7.

2 Basic Process Algebra with Termination and Deadlock

Let $Act = \{a, b, c, \cdots\}$ be a set of (atomic) actions, and $\mathcal{X} = \{x, y, z, ...\}$ a countable set of process variables. *Sequential process terms*, written $\mathcal{P}^s$, are those which do not involve parallelism and communication, which are generated by the following grammar:

$$E ::= \delta \mid \epsilon \mid x \mid a \mid E_1; E_2 \mid E_1 + E_2 \mid rec\, x.E$$

Intuitively, the elements of $\mathcal{P}^s$ represent programs: 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; the other constructors can be understood as the usual ones.

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 not. Formally, $\mathcal{T} \subset \mathcal{P}^s$ is the least set which contains ϵ

and is closed under the following 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 $\mathcal{T}(rec\, x.E)$.

An occurrence of a variable $x \in \mathcal{X}$ is called *free* in a term E iff it does not occur within a sub-term of the form $rec\, x.E'$, 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. A variable $x \in \mathcal{X}$ is called *guarded* within a 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 $\mathrm{BPA}_\delta^\epsilon$, ranged over by $P, Q, \cdots$, where BPA is a fragment of ACP [5].

An operational semantics of $\mathcal{P}^s$ is given in the standard Plotkin's style, yielding a transition system $(\mathcal{P}^s, \rightarrow)$ with $\rightarrow \subseteq \mathcal{P}^s \times Act \times \mathcal{P}^s$ that is the least relation derived from the rules in the Fig.2.

$$\text{Act}\ \frac{}{a \xrightarrow{a} \epsilon} \qquad \text{Rec}\ \frac{E[rec\, x.E/x] \xrightarrow{a} E'}{rec\, x.E \xrightarrow{a} E'} \qquad \text{Seq-1}\ \frac{E_1 \xrightarrow{a} E_1'}{E_1; E_2 \xrightarrow{a} E_1'; E_2}$$

$$\text{Seq-2}\ \frac{E_2 \xrightarrow{a} E_2' \wedge \mathcal{T}(E_1)}{E_1; E_2 \xrightarrow{a} E_2'} \qquad \text{Nd}\ \frac{E_1 \xrightarrow{a} E_1'}{E_1 + E_2 \xrightarrow{a} E_1', \quad E_2 + E_1 \xrightarrow{a} E_1'}$$

Fig.2. The Operational Semantics of $\mathcal{P}^s$

Definition 1. *A binary relation $\mathcal{S} \subseteq \mathrm{BPA}_\delta^\epsilon \times \mathrm{BPA}_\delta^\epsilon$ is called a strong bisimulation if $(P, Q) \in \mathcal{S}$ implies:*

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

Given two processes $P, Q \in \mathrm{BPA}_\delta^\epsilon$, 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 $E_1, E_2 \in \mathcal{P}^s$ and $fn(E_1) \cup fn(E_2) \subseteq \{x_1, \cdots, x_n\}$, if $E_1\{P_1/x_1, \cdots, P_n/x_n\} \sim E_2\{P_1/x_1, \cdots, P_n/x_n\}$ for any $P_1, \cdots, P_n \in \mathrm{BPA}_\delta^\epsilon$, then $E_1 \sim E_2$.

Convention: From now on, we use $\mathcal{A}\ op\ \mathcal{B}$ to stand for $\{E_1\ op\ E_2 \mid E_1 \in \mathcal{A}$ and $E_2 \in \mathcal{B}\}$, $\mathcal{A}\ op\ E$ for $\mathcal{A}\ op\ \{E\}$, where $E \in \mathcal{P}^s, \mathcal{A} \subseteq \mathcal{P}^s, \mathcal{B} \subseteq \mathcal{P}^s$, and $op \in \{+, ;\}$.

3 FLC with the Nondeterministic Operator "+" (FLC$^+$)

FLC, due to Markus Müller-Olm [21], is an extension of the modal μ-calculus that can express non-regular properties, and is therefore strictly more powerful than the μ-calculus. In order to study the compositionality of FLC, we extend

FLC with the nondeterministic operator "+", which is proposed as a primitive operator in [10, 18].

Let $X, Y, Z, \cdots$ 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 a process is terminated. Formulae of FLC$^+$ are generated by the following grammar:

$$\phi \ ::= \ tt \mid ff \mid \surd \mid \tau \mid X \mid [a] \mid \langle a \rangle \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 ; \phi_2 \mid \phi_1 + \phi_2 \mid \mu X.\phi \mid \nu X.\phi$$

where $X \in Var$ and $a \in Act$. The fragment of FLC$^+$ without "+" is called FLC [21]. In what follows, we use @ to stand for $\langle a \rangle$ or $[a]$, p for tt, ff or $\surd$, and σ for ν 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 *fixpoint 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.

Definition 2. *In the following, we define what it means for a formula to be a* guard*:*

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 $fn(\phi) \cup bn(\phi)$ are guarded, then ϕ is called* guarded*. A formula ϕ is said to be* strictly guarded *if ϕ is guarded and for any $X \in fn(\phi) \cup bn(\phi)$, 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 $\mathcal{L}_{\mathrm{FLC}^+}$ to denote all formulae of FLC$^+$ that are closed and guarded, and $\mathcal{L}_{\mathrm{FLC}}$ for the fragment of $\mathcal{L}_{\mathrm{FLC}^+}$ without $+$. In the sequel, we are only interested in closed and guarded formulae.

As in FLC, a formula of FLC$^+$ is interpreted as a *predicate transformer* which is a mapping $f : 2^{\mathrm{BPA}_\delta^\epsilon} \rightarrow 2^{\mathrm{BPA}_\delta^\epsilon}$. We use $\mathrm{MPT_T}$ to represent all these predicate transformers over $\mathrm{BPA}_\delta^\epsilon$.

The meaning of variables is given by a *valuation* ρ: $Var \rightarrow (2^{\mathrm{BPA}_\delta^\epsilon} \rightarrow 2^{\mathrm{BPA}_\delta^\epsilon})$ that assigns variables to functions from sets to sets. $\rho[X \rightsquigarrow f]$ agrees with ρ except for associating f with X.

Definition 3. *The meaning of a formula ϕ, under a valuation ρ, denoted by $[\![\phi]\!]_\rho$, is inductively defined as follows:*

$$
\begin{aligned}
[\![tt]\!]_\rho(\mathcal{A}) &= \mathrm{BPA}_\delta^\epsilon \\
[\![ff]\!]_\rho(\mathcal{A}) &= \emptyset \\
[\![\surd]\!]_\rho(\mathcal{A}) &= \{P \in \mathrm{BPA}_\delta^\epsilon \mid \mathcal{T}(P)\} \\
[\![\tau]\!]_\rho(\mathcal{A}) &= \mathcal{A} \\
[\![X]\!]_\rho(\mathcal{A}) &= \rho(X)(\mathcal{A}) \\
[\![[a]]\!]_\rho(\mathcal{A}) &= \{P \in \mathrm{BPA}_\delta^\epsilon \mid \neg\mathcal{T}(P) \wedge \forall P' \in \mathrm{BPA}_\delta^\epsilon . P \xrightarrow{a} P' \Rightarrow P' \in \mathcal{A}\} \\
[\![\langle a\rangle]\!]_\rho(\mathcal{A}) &= \{P \in \mathrm{BPA}_\delta^\epsilon \mid \exists P' \in \mathrm{BPA}_\delta^\epsilon . P \xrightarrow{a} P' \wedge P' \in \mathcal{A}\} \\
[\![\phi_1 \wedge \phi_2]\!]_\rho(\mathcal{A}) &= [\![\phi_1]\!]_\rho(\mathcal{A}) \cap [\![\phi_2]\!]_\rho(\mathcal{A}) \\
[\![\phi_1 \vee \phi_2]\!]_\rho(\mathcal{A}) &= [\![\phi_1]\!]_\rho(\mathcal{A}) \cup [\![\phi_2]\!]_\rho(\mathcal{A}) \\
[\![\phi_1 ; \phi_2]\!]_\rho &= [\![\phi_1]\!]_\rho \cdot [\![\phi_2]\!]_\rho \\
[\![\phi_1 + \phi_2]\!]_\rho(\mathcal{A}) &= \{P \in \mathrm{BPA}_\delta^\epsilon \mid P \sim P_1 + P_2 \wedge P_1 \in [\![\phi_1]\!]_\rho(\mathcal{A}) \wedge P_2 \in [\![\phi_2]\!]_\rho(\mathcal{A})\} \\
[\![\mu X.\phi]\!]_\rho &= \sqcap\{f \in \mathrm{MPT_T} \ \mid \ [\![\phi]\!]_{\rho[X \rightsquigarrow f]} \subseteq f\} \\
[\![\nu X.\phi]\!]_\rho &= \sqcup\{f \in \mathrm{MPT_T} \ \mid \ [\![\phi]\!]_{\rho[X \rightsquigarrow f]} \supseteq f\}
\end{aligned}
$$

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

Note that because ϵ and δ have different behaviour in the presence of ;, they should be distinguished in FLC$^+$. To this end, we interpret $[a]$ differently from in [21]. According to our interpretation, $P \models [a]$ only if $\neg\mathcal{T}(P)$, whereas in [21] it is always valid that $P \models [a]$ for any $P \in \mathcal{P}^s$. Thus, it is easy to show that $\epsilon \not\models \bigwedge_{a \in Act}[a]; ff$, while $\bigwedge_{a \in Act}[a]; ff$ is the characteristic formula of δ.

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 $\phi(\mathrm{BPA}_\delta^\epsilon)$. A process P is said to satisfy ϕ iff $P \in [\![\phi]\!]_\rho(\mathrm{BPA}_\delta^\epsilon)$ 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 $\mathcal{A} \subseteq \mathrm{BPA}_\delta^\epsilon$ and any ρ. $\phi \Leftrightarrow \psi$ means $(\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$. The other notations can be defined in the standard way.

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

As [16] proved that FLC has the tree model property, we can also show that FLC$^+$ has such property as well, i.e.,

Theorem 1. *Given* $P, Q \in \mathrm{BPA}_\delta^\epsilon$*,* $P \sim Q$ *iff for any* $\phi \in \mathcal{L}_{\mathrm{FLC}^+}$*,* $P \models \phi$ *iff* $Q \models \phi$*.*

4 A Connection Between $\mathrm{BPA}_\delta^\epsilon$ and FLC$^+$

In this section, we discuss how to relate the primitives of $\mathrm{BPA}_\delta^\epsilon$ to the connectives of FLC$^+$.

4.1 Nondeterminism

From Definition 3, it is clear that "+" of $\mathrm{BPA}^{\epsilon}_{\delta}$ corresponds to "+" of FLC^{+}. The connection can be expressed as follows:

Proposition 1. *For any $P, Q \in \mathrm{BPA}^{\epsilon}_{\delta}$, if $P \models \phi$ and $Q \models \psi$ then $P{+}Q \models \phi{+}\psi$.*

4.2 Sequential Composition

In this subsection, we show that under some conditions, the sequential composition ";" of $\mathrm{BPA}^{\epsilon}_{\delta}$ can be related to the chop ";" of FLC^{+}.

From the definition of the semantics of $\mathrm{BPA}^{\epsilon}_{\delta}$, it is clear that as far as the execution of the process $P;Q$ is concerned, Q starts to be executed only if P finishes the execution. A similar requirement on properties concerning P must be considered in order to derive a combined property for $P;Q$ from the properties for P and Q. For example, let $P = a;b$, $Q = c;d$, and it is therefore clear that $P \models \langle a\rangle$ and $Q \models \langle c\rangle$, however $P;Q \not\models \langle a\rangle;\langle c\rangle$. So, we require that the property about P must specify full executions of P, that is, $P \models \phi;\surd$.

On the other hand, it is easy to see that ϵ is a neutral element of ";" in $\mathrm{BPA}^{\epsilon}_{\delta}$. However, $\surd$, the counterpart of ϵ in FLC, is not the neutral element of the chop ";". Thus, we have to replace $\surd$ occurring in properties of P with τ in order to give a connection between ";" of $\mathrm{BPA}^{\epsilon}_{\delta}$ and the chop ";" of FLC^{+}. E.g., let $P = a;\epsilon$ and $Q = b;\delta$, $\phi = \langle a\rangle;\surd$, and $\psi = \langle b\rangle$. It's obvious that $P \models \phi;\surd$ and $Q \models \psi$, but $P;Q \not\models \phi;\psi$. Furthermore, it is required that $\surd$ can only appear at the end of properties of P, because from Definition 3 $\surd$ as a subformula of ϕ makes all subformulae following it with ; no sense during calculating the meaning of ϕ, but they will play a nontrivial role in the resulting formula. E.g. $\epsilon \models \surd;[a];\langle b\rangle$ and $a;c \models \langle a\rangle;\langle c\rangle$, but $\epsilon;(a;c) \not\models (\tau;[a];\langle b\rangle);(\langle a\rangle;\langle c\rangle)$. In fact, such a requirement can be always satisfied because all formulae can be transformed to such kind of the form equivalently.

In summary, the following theorem indicates the connection between the sequential composition ";" of $\mathrm{BPA}^{\epsilon}_{\delta}$ and the chop ";" of FLC^{+}.

Theorem 2. *For any $\phi, \psi \in \mathcal{L}_{\mathrm{FLC}^{+}}$ and any $P, Q \in \mathrm{BPA}^{\epsilon}_{\delta}$, if $\surd$ only occurs at the end of ϕ, $P \models \phi;\surd$ and $Q \models \psi$ then $P;Q \models \phi\{\tau/\surd\};\psi$.*

Remark 1. Generally speaking, the converse of Theorem 2 is not valid.

4.3 Recursion

In this subsection, we sketch how to relate $rec\ x$ to νX. Thus, in the rest of this sub-section all fixed point operators occurring in formulae will be referred to ν if not otherwise stated. To this end, we first employ a relation called *syntactical confirmation* between processes and formulae, with the type $\mathcal{P}^s \times FLC^{+} \mapsto \{\mathrm{tt}, \mathrm{ff}\}$, denoted by $\models_{sc}$.

Definition 4. *Given a formula ϕ, we associate a map from $2^{\mathcal{P}^s}$ to $2^{\mathcal{P}^s}$ with it, denoted by $\widehat{\phi}$, constructed by the following rules:*

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

where $\mathcal{E} \subseteq \mathcal{P}^s$.

$\models_{sc} (E, \phi) = \mathrm{tt}$ *iff* $E \in \widehat{\phi}(\{\epsilon\})$*; otherwise,* $\models_{sc} (E, \phi) = \mathrm{ff}$*. In what follows, we denote* $\models_{sc} (E, \phi) = \mathrm{tt}$ *by* $E \models_{sc} \phi$ *and* $\models_{sc} (E, \phi) = \mathrm{ff}$ *by* $E \not\models_{sc} \phi$ *.*

Informally, $P \models_{sc} \phi$ means that P and ϕ have a similar syntax, e.g.,

Example 2. Let $E_1 \mathrel{\widehat{=}} (a; x; x) + d$, $E_2 \mathrel{\widehat{=}} x; (b + c); y$, $E_3 \mathrel{\widehat{=}} a; b; c$, $\phi \mathrel{\widehat{=}} \langle a\rangle; X; X$, $\psi \mathrel{\widehat{=}} X; \langle b\rangle; Y$ and $\varphi \mathrel{\widehat{=}} [a]; \langle b\rangle; \langle c\rangle$. We have $E_1 \models_{sc} \phi$, $E_2 \models_{sc} \psi$, $E_3 \models_{sc} \varphi$.

The following theorem states that $\models_{sc}$ itself is compositional as well.

Theorem 3. *Let $\surd$ only appear at the end of ϕ_1, ϕ_2 and ϕ. Then,*

i) if $E_1 \models_{sc} \phi_1$ and $E_2 \models_{sc} \phi_2$ then $E_1 + E_2 \models_{sc} \phi_1 + \phi_2$;
ii) if $E_1 \models_{sc} \phi_1$ and $E_2 \models_{sc} \phi_2$ then $E_1; E_2 \models_{sc} \phi_1\{\tau/\surd\}; \phi_2$;
iii) if $E \models_{sc} \phi$ then $rec\, x.E \models_{sc} \sigma X.\phi\{\tau/\surd\}$.

Example 3. In Example 2, according to Theorem 3, we obtain $E_1 + E_2 \models_{sc} \phi + \psi$, $E_3; (E_1 + E_2) \models_{sc} \varphi; (\phi + \psi)$ and $rec\, x.\ rec\, y.E_3; (E_1 + E_3) \models_{sc} \nu X.\nu Y.(\varphi; (\phi + \psi))$.

Theorem 4 establishes a connection between $\models_{sc}$ and $\models$, so that $rec\ x$ is related to νX.

Theorem 4. *If $P \in \mathrm{BPA}_\delta^\epsilon$, $\surd$ only occurs at the end of ϕ and $P \models_{sc} \phi$, then $P \models \phi; \surd$.*

Theorem 4 provides the possibility to compositionally verify a complex system and even this can be done syntactically.

Example 4. For instance, let E_1, E_2, E_3 and ϕ, ψ, φ be as defined in Example 2. In order to verify $rec\ x.\ rec\ y.E_3; (E_1 + E_3) \models \nu X.\nu Y.(\varphi; (\phi + \psi))$, we only need to prove $E_1 + E_2 \models_{sc} \phi + \psi$ and $E_3; (E_1 + E_2) \models_{sc} \varphi; (\phi + \psi)$. This proof can further be reduced to $E_1 \models_{sc} \phi$, $E_2 \models_{sc} \psi$ and $E_3 \models_{sc} \varphi$. From Example 2, this is true.

5 Reducing $\mathcal{L}_{\mathrm{FLC}^+}$ to $\mathcal{L}_{\mathrm{FLC}}$

In this section, we will show that as far as closed and guarded formulae are concerned, the + of FLC$^+$ can be defined essentially by conjunction and disjunction, that is, for any $\phi \in \mathcal{L}_{\mathrm{FLC}^+}$, there exists a formula $\phi' \in \mathcal{L}_{\mathrm{FLC}}$ such that $\phi \Leftrightarrow \phi'$. This can be obtained via the following three steps: firstly, we show that in some special cases "+" can be defined by conjunction and disjunction essentially; then we prove that the elimination of "+" in a strictly guarded formula ϕ of FLC$^+$ can be reduced to one of the above special cases; and finally, we complete the proof by showing that for any $\phi \in \mathcal{L}_{\mathrm{FLC}^+}$ there exists a strictly guarded formula $\phi' \in \mathcal{L}_{\mathrm{FLC}^+}$ such that $\phi \Leftrightarrow \phi'$.

The following lemma claims that in some special cases, "+" can be defined essentially by conjunction and disjunction.

Lemma 1. *Let $n, k \leq m$, $\{a_1, \cdots, a_n\}$ and $\{c_1, \cdots, c_k\}$ be subsets of $\{b_1, \cdots, b_m\}$, where $b_i \neq b_j$ if $i \neq j$. Assume $< a_1, \cdots, a_n > = < b_1, \cdots, b_n >$ and $< c_1, \cdots, c_k > = < b_{l_1}, \cdots, b_{l_k} >$, where $l_j \in \{1, \cdots, m\}$ for $j = 1 \cdots k$. Then*

$$(\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n_i} \langle a_i \rangle; \phi_{i,j} \wedge \bigwedge_{i=1}^{m} [b_i]; \psi_i \wedge q_1) + (\bigwedge_{i=1}^{k} \bigwedge_{j=1}^{k_i} \langle c_i \rangle; \varphi_{i,j} \wedge \bigwedge_{i=1}^{m} [b_i]; \chi_i \wedge q_2)$$

$$\Leftrightarrow \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n_i} \langle a_i \rangle; (\phi_{i,j} \wedge \psi_i) \wedge \bigwedge_{i=1}^{k} \bigwedge_{j=1}^{k_i} \langle c_i \rangle; (\varphi_{i,j} \wedge \chi_{l_i}) \wedge \bigwedge_{i=1}^{m} [b_i]; (\psi_i \vee \chi_i) \wedge q_1 \wedge q_2$$

where $q_1 \Leftrightarrow tt$ or $q_1 \Leftrightarrow \tau$, and $q_2 \Leftrightarrow tt$ or $q_2 \Leftrightarrow \tau$.

Proof (Sketch). According to Definition 3, it is easy to see that + and ; both are monotonic. On the other hand, it is not hard to prove that 1. if $P \models \langle a \rangle; \phi$, then $P + Q \models \langle a \rangle; \phi$ for any $Q \in \mathrm{BPA}_\delta^\epsilon$; 2. $P \models [a]; \phi$ and $Q \models [a]; \psi$, then $P + Q \models [a]; (\phi \vee \psi)$; 3. $([a]; \phi \wedge \langle a \rangle; \psi) \Rightarrow (\langle a \rangle; (\phi \wedge \psi) \wedge [a]; \phi)$. Thus, it is not hard to prove the forward direction.

For the converse direction, we first prove that given a $P \in \mathrm{BPA}_\delta^\epsilon$, there exists a $Q \in \mathrm{BPA}_\delta^\epsilon$ of the form $\sum_{a \in Act} \sum_{j=1}^{i_a} a; Q_{a,j}$ or δ such that $P \sim Q$; then by Theorem 1, P satisfies the formula of the right hand in the lemma iff Q also meets it; subsequently, we design an algorithm to partition all summands of Q into two parts $\mathcal{Q}_1$ and $\mathcal{Q}_2$ such that $\sum \mathcal{Q}_1$ satisfies the first operand of "+" in the left formula of the lemma, $\sum \mathcal{Q}_2$ meets the second operand. Obviously, $\sum \mathcal{Q}_1 + \sum \mathcal{Q}_2 \sim P$. Therefore, the converse direction has been proved. ⊣

By applying the above lemma, induction on the given formula ϕ, we can show that if ϕ is strictly guarded, then there exists ϕ' such that $\phi \Leftrightarrow \phi'$ and no + occurs in ϕ', i.e.

Lemma 2. *For any ϕ of* FLC$^+$ *, if ϕ is strictly guarded, then there exists ϕ' of* FLC *such that $\phi' \Leftrightarrow \phi$.*

In the below, we will apply some rewriting techniques to prove that for any closed and guarded formula ϕ of FLC$^+$, there exists ϕ' that is strictly guarded such that $\phi \Leftrightarrow \phi'$, namely

Lemma 3. *For any $\phi \in \mathcal{L}_{\mathrm{FLC}^+}$, there is $\phi' \in \mathcal{L}_{\mathrm{FLC}^+}$ that is strictly guarded such that $\phi \Leftrightarrow \phi'$.*

Proof (Sketch). In order to prove the lemma, we need to show the following equations:

$$\mu X.\phi_1[@;\phi_2[(X \odot \phi_3) + \phi_4]] \Leftrightarrow \mu X.\phi_1[@;\phi_2[\mu Y.(\phi_1[@;\phi_2[Y]] \odot \phi_3) + \phi_4]] \tag{1}$$

$$\nu X.\phi_1[@;\phi_2[(X \odot \phi_3) + \phi_4]] \Leftrightarrow \nu X.\phi_1[@;\phi_2[\nu Y.(\phi_1[@;\phi_2[Y]] \odot \phi_3) + \phi_4]] \tag{2}$$

$$\mu X.\phi_1[@;\phi_2[(X;\phi_3 \odot \phi_4) + \phi_5]] \Leftrightarrow \mu X.\phi_1[@;\phi_2[\mu Y.(\phi_1[@;\phi_2[Y]];\phi_3 \odot \phi_4) + \phi_5]] \tag{3}$$

$$\nu X.\phi_1[@;\phi_2[(X;\phi_3 \odot \phi_4) + \phi_5]] \Leftrightarrow \nu X.\phi_1[@;\phi_2[\nu Y.(\phi_1[@;\phi_2[Y]];\phi_3 \odot \phi_4) + \phi_5]] \tag{4}$$

where $\odot \in \{\wedge, \vee\}$, $\phi_i[\ ]$ stands for a formula with the hole $[\ ]$, the formula at the left side of each equation is guarded.

We will only prove (3) as an example, the others can be proved similarly. Since $\phi_1[@;\phi_2[(X;\phi_3 \odot \phi_4) + \phi_5]]$ is guarded, by Knaster-Tarski Theorem, it is clear that $\mu X.\phi_1[@;\phi_2[(X;\phi_3 \odot \phi_4) + \phi_5]]$ is the unique least solution of the equation

$$X = \phi_1[@;\phi_2[(X;\phi_3 \odot \phi_4) + \phi_5]] \tag{5}$$

Let Y be a fresh variable and $Y = (X;\phi_3 \odot \phi_4) + \phi_5$. It is easy to see the least solution of (5) is equivalent to the X-component of the least solution of the following equation system:

$$\begin{aligned} X &= \phi_1[@;\phi_2[(X;\phi_3 \odot \phi_4) + \phi_5]] \\ Y &= (X;\phi_3 \odot \phi_4) + \phi_5 \end{aligned}$$

Meanwhile, exploiting some rewriting techniques, it is easy to transform solving the least solution of the above equation system to the following one equivalently,

$$\begin{aligned} X &= \phi_1[@;\phi_2[(X;\phi_3 \odot \phi_4) + \phi_5]] \\ Y &= (\phi_1[@;\phi_2[Y]];\phi_3 \odot \phi_4) + \phi_5 \end{aligned}$$

It is not hard to obtain the least solution of the above equation system as $(\mu X.\phi_1[@;\phi_2[\mu Y.(\phi_1[@;\phi_2[Y]];\phi_3 \odot \phi_4) + \phi_5]], \mu Y.(\phi_1[@;\phi_2[Y]];\phi_3 \odot \phi_4) + \phi_5)$. Therefore, (3) follows.

Repeatedly applying (1)–(4), for any given formula $\phi \in \mathcal{L}_{\mathrm{FLC}^+}$, we can rewrite it to ϕ' which is strictly guarded such that $\phi \Leftrightarrow \phi'$. $\dashv$

Remark 2. In the proof for Lemma 3, we only consider the cases that a variable is guarded by a modality@, and ignore the cases that a variable is guarded by a propositional letter p, because according to Definition 3 it is easy to show that $p;\phi \Leftrightarrow p$.

From the above lemmas, the following result is immediate.

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

We use the following example to demonstrate how to translate a closed and guarded formula ϕ of FLC$^+$ into a formula ϕ' of FLC by applying the above procedure.

Example 5. Let $\phi = \mu X.\nu Y.\langle a\rangle;(X+Y);X;Y;\langle b\rangle \vee \langle c\rangle$. Applying (1), it follows

$$\phi \Leftrightarrow \mu X.\nu Y.\langle a\rangle;[\mu Z.(\nu V.\langle a\rangle;Z;X;V;\langle b\rangle \vee \langle c\rangle)+Y];X;Y;\langle b\rangle \vee \langle c\rangle \ \ \widehat{=}\ \ \phi'$$

where $\phi_1[\ ] \widehat{=} \nu Y.[\ ];X;Y;\langle b\rangle \vee \langle c\rangle$, $\phi_2[\ ] \widehat{=} [\ ]$, $\phi_3 \widehat{=} \begin{cases} tt & \text{if } \odot = \wedge \\ ff & \text{o.w.} \end{cases}$, $\phi_4 \widehat{=} Y$. Furthermore, applying (2), we can get

$$\begin{aligned}\phi' \Leftrightarrow\ & \mu X.\nu Y.\langle a\rangle;[\mu Z.\nu W.(\langle a\rangle;W;X;Y;\langle b\rangle \vee \langle c\rangle)+(\nu V.\langle a\rangle;Z;X;V;\langle b\rangle \vee \langle c\rangle)];\\ & X;Y;\langle b\rangle \vee \langle c\rangle \ \ \widehat{=}\ \ \phi''\end{aligned}$$

where $\phi_1[\ ] \ \widehat{=}\ [\ ];X;Y;\langle b\rangle \vee \langle c\rangle$, $\phi_2[\ ] \ \widehat{=}\ \mu Z.[\ ]$, $\phi_3 \ \widehat{=}\ \begin{cases} tt & \text{if } \odot = \wedge \\ ff & \text{o.w.} \end{cases}$, $\phi_4 \ \widehat{=}\ \nu V.\langle a\rangle;Z;X;V;\langle b\rangle \vee \langle c\rangle$. Thus, using Lemma 2, we can eliminate "+" in ϕ'' as follows:

$$\begin{aligned}\phi'' \Leftrightarrow\ & \mu X.\nu Y.\langle a\rangle;[\mu Z.\nu W.\begin{pmatrix}(\langle a\rangle;W;X;Y;\langle b\rangle+\langle c\rangle)\vee\\ (\langle a\rangle;W;X;Y;\langle b\rangle+\\ \nu V.\langle a\rangle;Z;X;V;\langle b\rangle)\vee\\ (\nu V.\langle a\rangle;Z;X;V;\langle b\rangle+\langle c\rangle)\vee\\ (\langle c\rangle+\langle c\rangle)\end{pmatrix}];X;Y;\langle b\rangle \vee \langle c\rangle\\ \Leftrightarrow\ & \mu X.\nu Y.\langle a\rangle;[\mu Z.\nu W.\begin{pmatrix}((\langle a\rangle;W;X;Y;\langle b\rangle\wedge\langle c\rangle)\vee\\ (\langle a\rangle;W;X;Y;\langle b\rangle\wedge\\ \nu V.\langle a\rangle;Z;X;V;\langle b\rangle)\vee\\ (\nu V.\langle a\rangle;Z;X;V;\langle b\rangle\wedge\langle c\rangle)\vee\\ \langle c\rangle\end{pmatrix};X;Y;\langle b\rangle \vee \langle c\rangle\\ \widehat{=}\ & \phi^*\end{aligned}$$

It is easy to see that $\phi \Leftrightarrow \phi^*$ and no + occurs in ϕ^*. $\dashv$

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

6 Constructing Characteristic Formulae for Context-free Processes Compositionally

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 $Q\mathcal{R}P$. [21] presented a method to derive the characteristic formula for a context-free process up to strong (weak) bisimulation by solving the equation system induced by the rewrite system of the process in FLC. In this section, we present an algorithm to construct the characteristic formula for a process of $\mathrm{BPA}_\delta^\epsilon$ up to strong bisimulation directly from its syntax in a compositional manner based on the above results, in contrast to the semantics-based method given in [21]. We believe that our approach also works for weak bisimulation, but it is necessary to re-interpret modalities of FLC.

It is easy to see that $\bigwedge_{a\in Act}[a];ff$ (Φ_δ for short) is the characteristic formula for δ, and $\surd$ for ϵ.

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

Definition 5. *Given a process term $E \in \mathcal{P}^s$, we associate with it a formula denoted by Ψ_E derived by the following rules:*

$$\begin{array}{llll}
\Psi_\delta & \widehat{=}\ \Phi_\delta, & \Psi_\epsilon & \widehat{=}\ \surd, \\
\Psi_x & \widehat{=}\ X, & \Psi_a & \widehat{=}\ \Phi_{-\{a\}} \wedge (\langle a\rangle \wedge [a]), \\
\Psi_{E_1;E_2} & \widehat{=}\ \Psi_{E_1}\{\tau/\surd\};\Psi_{E_2}, & \Psi_{E_1+E_2} & \widehat{=}\ \Psi_{E_1}+\Psi_{E_2}, \\
\Psi_{rec\ x.E} & \widehat{=}\ \nu X.\Psi_E\{\tau/\surd\}. & &
\end{array}$$

Regards Definition 5, we have

Lemma 4. *1. For any $E \in \mathcal{P}^s$, $\surd$ only occurs at the end of Ψ_E;*
2. For any $E \in \mathcal{P}^s$, $E \models_{sc} \Psi_E$ and $E \models_{sc} \Psi_E;\surd$;
3. For any $P \in \mathrm{BPA}_\delta^\epsilon$, $\Psi_P;\surd$ is closed and guarded.

The following theorem states if two processes are strong bisimilar then the derived formulae are equivalent.

Theorem 6 (Completeness). *If $E_1 \sim E_2$, then $\Psi_{E_1} \Leftrightarrow \Psi_{E_2}$.*

We can show that $en(\Psi_P;\surd)$ is the characteristic formula of P up to $\sim$ for each $P \in \mathrm{BPA}_\delta^\epsilon$.

Theorem 7. *For any $P \in \mathrm{BPA}_\delta^\epsilon$, $en(\Psi_P;\surd)$ is the characteristic formula of P up to $\sim$.*

Remark 3. In Theorem 7, the condition that P is guarded is essential. Otherwise, the theorem is not true any more. For instance, $\nu X.(X + (\langle a\rangle \wedge [a] \wedge \Phi_{-\{a\}}))$ is equivalent to $\Psi_{rec\ x.(x+a)}$, nevertheless, $(\nu X.(X + (\langle a\rangle \wedge [a] \wedge \Phi_{-\{a\}})));\surd$ is not the characteristic formula of $rec\ x.(x+a)$, since $rex\ x.(x+b+a)$ meets the formula, but $rex\ x.(x+b+a) \not\sim rec\ x.(x+a)$.

Example 6. Let $P \widehat{=} a;\epsilon$ and $Q \widehat{=} b;\delta$. Then, $\Psi_P \widehat{=} (\langle a\rangle \wedge [a] \wedge \Phi_{-\{a\}});\surd$, and , $\Psi_Q \widehat{=} (\langle b\rangle \wedge [b] \wedge \Phi_{-\{b\}});\Phi_\delta$ by Definition 5.

It's obvious that $en(\Psi_P;\surd) = \Psi_P;\surd$ is the characteristic formula of P and $en(\Psi_Q;\surd) = \Psi_Q;\surd$ is the one of Q. Furthermore, by Definition 5,

$$\begin{array}{l}
\quad en(\Psi_{rec\ x.(P;x;x;Q+P)};\surd) \\
\widehat{=}\ en([\nu X.\begin{pmatrix} (\langle a\rangle \wedge [a] \wedge \Phi_{-\{a\}});X;X;((\langle b\rangle \wedge [b] \wedge \Phi_{-\{b\}});\Phi_\delta) \\ +\ (\langle a\rangle \wedge [a] \wedge \Phi_{-\{a\}}) \end{pmatrix}];\surd) \\
\Leftrightarrow [\nu X.\begin{pmatrix} \langle a\rangle;X;X;(\langle b\rangle \wedge [b] \wedge \Phi_{-\{b\}});\Phi_\delta \wedge \langle a\rangle \wedge \\ [a];(\tau \vee X;X;(\langle b\rangle \wedge [b] \wedge \Phi_{-\{b\}});\Phi_\delta) \wedge \Phi_{-\{a\}} \end{pmatrix}];\surd
\end{array}$$

which is exactly the characteristic formula of $rec\ x.(a;x;x;b;\delta + a;\epsilon)$. $\dashv$

7 Concluding Remarks

In this paper, we investigated the compositionality of FLC. To this end, inspired by [10, 18], we first extended FLC with the non-deterministic choice "+" and then established a connection between the primitives of $\mathrm{BPA}_{\delta}^{\epsilon}$ and the connectives of FLC^{+}, and finally, we proved that as far as closed and guarded formulae are concerned, "+" can be defined essentially by conjunction and disjunction in FLC.

Although introducing "+" cannot improve the expressive power of FLC, using it as an auxiliary can be applied to compositional specification and verification of a complex system, some advantages have been argued in the Introduction. As an application of the compositionality of FLC, we presented an algorithm to construct the characteristic formula of each process of $\mathrm{BPA}_{\delta}^{\epsilon}$ directly according to its syntax in contrast to the method in [21] which derives the characteristic formula for a process from the transition graph of the process. We believe that our approach also works for weak bisimulation, but it is necessary to re-interpret modalities of FLC.

Various work concerning compositionality of modal and temporal logics have been done, for example, [9, 18] directly introduced the non-deterministic operator "+" into the modal μ-calculus like logics so that the resulted logics have compositionality; [3, 4] discussed the compositionality of linear temporal logic [23] by introducing the chop into the logic, while [24] investigated some logic properties of the extension; [19, 20] studied the compositionality of μ-calculus; [26] investigated the compositionality of a fixpoint logic in assume-guarantee style. Comparing with the previous work, the logics studied in previous work can only express regular properties, but FLC which is investigated in this paper can define non-regular properties. [9] gave a method to define characteristic formulae for finite terms of CCS up to observational congruence, [25] furthered the work by presenting an approach to define characteristic formulae for regular processes up to some preorders; Moreover, [21] gave a method to define characteristic formulae for context-free processes up to some preorders based on the rewriting system of a given process. In contrast to [21], in our approach characteristic formulae of $\mathrm{BPA}_{\delta}^{\epsilon}$ are constructed directly from syntax.

As future work, it is worth investigating the parallel operator and establishing a proof system for FLC.

References

1. L. Aceto and M. Hennessy. Termination, deadlock, and divergence. *Journal of ACM,* Vol. 39, No.1: 147-187. January, 1992.
2. H.R. Andersen, C. Stirling, G. Winskel. A compositional proof system for the modal mu-Calculus. LICS'94, pp.144-153.
3. H. Barringer, R. Kuiper, A. Pnueli. Now you may compose temporal logic specifications. In Proc. 16^{th} STOC, pp. 51-63. 1984.
4. 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.

5. J.A. Bergstra and J.W. Klop. Algebra of communication processes with abstraction. *Theoretical Computer Science,* 37:77-121. 1985.
6. A. Chandra, J. Halpern, A. Meyer and R. Parikh. Equations between regular terms and an application to process logic. In Proc. 13^{th} STOC, pp.384-390. 1981.
7. B. Dutertre. On first order interval logic. LICS'95, pp. 36-43, 1995.
8. E.A. Emerson and C.S. Jutla. Tree automata, μ-calculus, and determinacy. In Proc. 33^{rd} FOCS, pp.368-377. 1991.
9. S. Graf and J. Sifakis. A modal characterization of observational congruence on finite terms of CCS. *Information and Control,* 68:125-145. 1986.
10. S. Graf and J. Sifakis. A logic for the description of non-deterministic programs and their properties. *Information and Control,* 68:254-270. 1986.
11. J. Halpern, B. Moskowski, and Z. Manna. A hardware semantics based on temporal intervals. ICALP'83, LNCS 154, pp. 278-291, 1983.
12. D. Harel, D. Kozen and R. Parikh. Process Logic: Expressiveness, decidability, completeness. In IEEE FOCS'80, pp. 129-142. 1980.
13. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
14. D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ-calculus with respect to monadic second order logic. CONCUR'96, LNCS 1119, pp.263-277. 1996.
15. D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science,* 27:333-354. 1983.
16. M. Lange and C. Stirling. Model checking fixed point logic with chop. FOSSACS'02, LNCS 2303, pp. 250-263. 2002.
17. M. Lange. Local model checking games for fixed point logic with chop. CONCUR'02, LNCS 2421, pp. 240-254. 2002.
18. K.G. Larsen and B. Thomsen. A modal process logic. In the proc. of LICS'88, pp.203-210. 1988.
19. K.G. Larsen and X.X. Liu. Compositionality through an operational semantics of contexts. ICALP'90, LNCS 443, pp.526-539. 1990.
20. K.G. Larsen and X.X. Liu. Equation solving using modal transition systems. LICS'90, pp. 108-107. 1990.
21. M. Müller-Olm. A modal fixpoint logic with chop. STACS'99, LNCS 1563, pp.510-520. 1999.
22. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.
23. A. Pnueli. The temporal logic of programs. In Proc. 18^{th} STOC, pp.232-239. 1977.
24. R. Rosner and A. Pnueli. A choppy logic. In the proc. of LICS'86, pp.306-313. 1986.
25. B. Steffen, A. Ingólfsdóttir. Characteristic formulae for processes with divergence. *Information and Computation,* 110:149-163. 1994.
26. M. Viswanathan and R. Viswanathan. Foundations for circular compositional reasoning. ICALP'01, LNCS 2076, pp. 835-847, 2001.
27. Naijun Zhan. Compositional properties of sequential processes. In the proc. of SVV'03, *ENTCS* 118, pp.111-128. 2005.
28. C.C. Zhou, C.A.R. Hoare, and A. Ravn. A calculus of durations. *Information Processing Letters*, 40(5):269-276, 1991.