Deriving Non-determinism from Conjunction and Disjunction

Naijun Zhan⋆[1] and Mila Majster-Cederbaum[2]

[1] Lab. of Computer Science, Institute of Software, Chinese Academy of Sciences, South Fourth Street No.4, Zhong Guan Cun, 100080, Beijing, P.R. China
[2] Lehrstuhl für Praktische Informatik II, Fakultät für Mathematik und Informatik, Mannheim Universität, D7,27, 68163, Mannheim, Deutschland

Abstract. In this paper, we show that the non-deterministic choice "+", which was proposed as a primitive operator in *Synchronization Tree Logic* (STL for short) can be defined essentially by conjunction and disjunction in the μ-calculus (μM for short). This is obtained by extending the μ-calculus with the non-deterministic choice "+" (denoted by μM^+) and then showing that μM^+ can be translated into μM. Furthermore, we also prove that STL can be encoded into μM^+ and therefore into μM.

Keywords: non-determinism, Synchronization Tree Logic, μ-calculus, process algebra

1 Introduction

Compositional methods allow one to build up a large system by composing existing systems with the defined constructors and reduce the problem of correctness for a complex system to similar and simpler correctness problems for the subsystems. Because the complexity of large systems is normally untractable, it is necessary that a method for developing these systems is compositional (vertically or horizontally) in order to avoid combinatorial explosion in specifying and verifying these systems.

It is widely agreed that modal and temporal logics such as the μ-calculus [5] and Hennessy-Milner Logic (HML for short) [4], are an appropriate tool for the specification and proof of reactive systems. In many cases, these systems can be modelled by the term language $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ of an algebra with a congruence relation $\sim$, where $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ is constructed from a constant ϵ by using a set *Act* of unary operators, a binary operator + and *recursion*. $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ is at the base of many process algebras, where *Act* represents a set of *action names*, + the *non-deterministic choice* and ϵ the system performing no actions. The terms can be interpreted over trees labeled over *Act* - *synchronization trees* - following the terminology of [8]. It is required that modal logics $\mathcal{L}$ meet the condition of adequacy, namely,

$$\forall t_1, t_2 \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]\, (t_1 \sim t_2 \text{ iff } \forall \phi \in \mathcal{L}(t_1 \models \phi \text{ iff } t_2 \models \phi)).$$

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

I.e, the congruence $\sim$ and the equivalence relation induced by the logic agree. For example, HML has the property, i.e., two CCS terms are equal up to strong bisimulation if and only if they satisfy the same HML properties, see [4].

On the other hand, it is desirable that the logics have compositionality, i.e. there exists a connection between the connectives of these logics and the constructors of programs so that one can reduce the problem of correctness for a complex system to similar and simpler correctness problems for the subsystems. It seems that many classic modal logics like the μ-calculus and HML do not have such a property.

Motivated by the above two requirements, Graf and Sifakis proposed a modal logic, called *Synchronization Tree Logic* (STL) [2]. The language of formulae of STL is generated from the constants $\epsilon, \top$ by using the *boolean connectives*, the set 2^{Act} of unary operators where Act is a set of actions, the binary operator $+$ and *fixpoint operators*. The operator $+$ of the logic is an extension of the one $+$ of programs. $P \models \phi_1 + \phi_2$ means that there exist P_1 and P_2 such that $P \sim P_1 + P_2$, $P_1 \models \phi_1$ and $P_2 \models \phi_2$. Therefore, $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ is contained in STL, i.e., programs are formulae of the logic. In order to avoid confusions, we will use ϕ_P to denote the formula corresponding to the program P. So, the verification of an assertion $P \models \phi$ can be reduced to the syntax-directed proof of the validity of the formula $\phi_P \Rightarrow \phi$.

It is clear that STL is more expressive than μM since it is not hard to encode μM into STL, for example, $[A]\phi$ can be defined as $\neg(A\neg\phi_{\mathrm{STL}} + \top)$ and $\langle A\rangle\phi$ as $A\phi_{\mathrm{STL}} + \top$, where $A \subseteq Act$ and ϕ_{STL} stands for the counterpart of ϕ in STL. But for the converse direction, by our knowledge, it seems that until up to now it is still open.

In this paper, we will study the issue of the definability of $+$ in μM and give an affirmative answer. We show that the choice $+$ can be defined essentially by conjunction and disjunction in μM. This is captured by extending μM with the choice $+$ to $\mu\mathrm{M}^+$ and then encoding $\mu\mathrm{M}^+$ into μM. Furthermore, we show that STL can be translated into $\mu\mathrm{M}^+$, and we can thus claim that μM is as expressive as STL.

The rest of this paper is organized as follows: Some basic notions are defined in Section 2. Section 3 briefly reviews μM firstly, then extends it with the nondeterministic choice $+$ to $\mu\mathrm{M}^+$. Section 4 is devoted to encoding $\mu\mathrm{M}^+$ into μM. STL and some related results are provided in Section 5. Section 6 is devoted to translating STL into $\mu\mathrm{M}^+$. A short conclusion is given in Section 7.

2 Preliminaries

Consider a term language $\mathcal{T}$ built from the constants ϵ, τ, and a set $\mathcal{X}$ of process variables by using a set Act of unary operators, a binary operator $+$, and recursion.

Formally, $\mathcal{T}$ is formed according to the following rules:

- $\epsilon, \tau \in \mathcal{T}$, $\mathcal{X} \subseteq \mathcal{T}$,
- $aP, P_1 + P_2, rec\ x.P \in \mathcal{T}$ if $a \in Act, x \in \mathcal{X}, P, P_1, P_2 \in \mathcal{T}$.

We denote by $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$ the sub-language which consists of all the well-guarded and closed terms in $\mathcal{T}$, where $rec\ x.P$ is well-guarded means that any occurrence of the variable x in P is within the scope of an operator of Act.

For a given $P \in \mathcal{T}$, the set of actions that occur in P is called its *sort*, denoted $S(P)$, inductively defined by $S(\epsilon) \widehat{=} \emptyset$, $S(\tau) \widehat{=} Act$, $S(x) \widehat{=} \emptyset$, $S(a\,P) \widehat{=} \{a\} \cup S(P)$, $S(P_1 + P_2) \widehat{=} S(P_1) \cup S(P_2)$, $S(rec\,x.P) \widehat{=} S(P)$.

Intuitively, we consider that elements of $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$ represent programs: Act is a set of atomic actions; $+$ stands for *non-deterministic choice*; and ϵ for the program performing no actions; τ can be conceived as a program that behaves like **chaos** in CSP [3] which can do anything.

A structured operational semantics of $\mathcal{T}$ in Plotkin's Style is defined as follows:

Act $\dfrac{}{aP \xrightarrow{a} P}$ Nd $\dfrac{P_1 \xrightarrow{a} P_1'}{P_1 + P_2 \xrightarrow{a} P_1', \quad P_2 + P_1 \xrightarrow{a} P_1'}$

Rec $\dfrac{P_1[rec\ x.P_1/x] \xrightarrow{a} P_1'}{rec\ x.P_1 \xrightarrow{a} P_1'}$ Chaos $\dfrac{}{\tau \xrightarrow{a} Q}$ for any $a \in Act$ and $Q \in \mathcal{T}$.

A process term $P \in \mathcal{T}$ determines a labelled *transition system*, i.e., a tuple $\mathcal{T}(P) = (\Sigma, S(P), \rightarrow, P)$, where Σ is the set of states which is reachable from P, and $P \in \Sigma$ is the initial state, $\rightarrow \subseteq \Sigma \times S(P) \times \Sigma$ is the set of transitions, derived from the above operational semantics.

Remark 1. 1. Any transition system representing a term of $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$ is always finitely branching as only well-guarded terms are admitted;
2. The sort of each term of $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$ is finite as so is its syntax.

Definition 1. *A binary relation $\mathcal{S}$ over $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$ is called a strong bisimulation if $(P,Q) \in \mathcal{S}$ implies*

- *whenever $P \xrightarrow{a} P'$ then, for some $Q', Q \xrightarrow{a} Q'$ and $(P',Q') \in \mathcal{S}$, for any $a \in Act$; and*
- *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 $P, Q \in \mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$, P and Q are strongly bisimilar, written $P \sim Q$, if $(P,Q) \in \mathcal{S}$ for some strong bisimulation $\mathcal{S}$.

It is shown in [7] that $\sim$ is a congruence on $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$. [1] proved the following result, namely,

Lemma 1. *For each $P \in \mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$, there exists a process of the form $\Sigma_{i=1}^{m}\Sigma_{j=1}^{i_{a_i}} a_i P_{i,j}$ such that $P \sim \Sigma_{i=1}^{m}\Sigma_{j=1}^{i_{a_i}} a_i P_{i,j}$, where $a_i \neq a_j$ if $i \neq j$.*

Note that an empty sum is abbreviated as ϵ.

3 The μ-calculus and Its Extension with "+"

In this section, we first briefly review the μ-calculus; then extend the logic with the non-deterministic operator "+". We denote by $\mu\mathrm{M}^{+}$ the extension.

For easing to encode STL into μM, we use the slightly generalized version of the μ-calculus (see [9]) in the sense that modalities on sets of actions are adopted rather than modalities on a single action, although the two formalisms are equivalent if the set of actions is assumed to be finite.

3.1 μM

Let Act be a set of atomic actions, ranged over by $a, b, c, \ldots$. $A, B, \ldots$ stand for the subsets of Act. Let tt be propositional constant as usual, and $\mathcal{X}$ be a set of variables, ranged over by $x, y, z, \ldots$.

Formulae of μM are generated by:

$$\phi \ ::= \ tt \mid x \mid \neg\phi \mid \phi \vee \phi \mid \langle A \rangle \phi \mid [A]\phi \mid \mu x.\phi,$$

where $A \subseteq Act$ and $x \in \mathcal{X}$.

The notions of *scope*, *bound* and *free occurrences* of variables, *closed* and *open formulae*, etc. are the same as in first-order predicate logic, where μx is treated as *quantifier*. We will use $fn(\phi)$ to stand for the variables that have some free occurrence in ϕ, and $bn(\phi)$ for the variables that have some bound occurrence in ϕ. We say that ϕ is *positive (negative) in the variable* x if every free occurrence of x in ϕ occurs within the scope of an even (odd) number of negations $\neg$. A formula ϕ is said *positive (negative)* if for every $x \in bn(\phi)$, its scope in ϕ is positive (negative) in x. A formula ϕ is called *strongly positive* if it is positive and each occurrence of x is within the scope of an even number of negations $\neg$ for any $x \in fn(\phi)$. For example, let $\phi_1 \widehat{=} x \vee \mu x.\neg\neg x$, $\phi_2 \widehat{=} \neg y \vee \mu x.\neg\neg x$. It is clear that ϕ_1 and ϕ_2 both are positive; however, ϕ_1 is strongly positive as well, but ϕ_2 is not. We say that x is *guarded* in ϕ if every occurrence of x in ϕ is within the scope of $\langle A \rangle$ or $[A]$ for some $A \subseteq Act$. A formula ϕ is called *guarded* if each variable in $bn(\phi)$ is guarded.

If $A = \{a\}$, we directly write $\langle a \rangle \phi$ and $[a]\phi$ instead of $\langle \{a\} \rangle \phi$ and $[\{a\}]\phi$ respectively.

We denote by $\mathcal{L}_\mu(Act)$ the language of formulae of μM that are positive and guarded, by $c\mathcal{L}_\mu(Act)$ the set of all closed formulae in $\mathcal{L}_\mu(Act)$. As [11] showed that any formula $\phi \in \mu$M is equivalent to a positive guarded formula ϕ', we theorefore only focus on $\mathcal{L}_\mu(Act)$ and $c\mathcal{L}_\mu(Act)$ in what follows.

A valuation ρ is a mapping with the type $\rho : \mathcal{X} \rightarrow 2^{\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]}$, which associates a set of processes with each propositional variable. $\rho[x \rightsquigarrow \mathcal{A}]$ agrees with ρ except for assigning $\mathcal{A}$ to x.

Definition 2. *The semantics of $\mathcal{L}_\mu(Act)$ under a valuation ρ is given by a satisfaction relation between $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ and $\mathcal{L}_\mu(Act)$ relative to ρ, denoted by $\models^\rho_{\mu\mathrm{M}}$, inductively defined as follows:*

$$
\begin{aligned}
P &\models^{\rho}_{\mu\mathrm{M}} tt,\\
P &\models^{\rho}_{\mu\mathrm{M}} x, \textit{ iff } P \in \rho(x),\\
P &\models^{\rho}_{\mu\mathrm{M}} \neg\phi \textit{ iff } P \not\models^{\rho}_{\mu\mathrm{M}} \phi,\\
P &\models^{\rho}_{\mu\mathrm{M}} \phi_1 \vee \phi_2 \textit{ iff } P \models^{\rho}_{\mu\mathrm{M}} \phi_1 \textit{ or } P \models^{\rho}_{\mu\mathrm{M}} \phi_2,\\
P &\models^{\rho}_{\mu\mathrm{M}} \langle A\rangle\phi \textit{ iff } \exists a \in A, \exists P'.P \xrightarrow{a} P' \textit{ and } P' \models^{\rho}_{\mu\mathrm{M}} \phi,\\
P &\models^{\rho}_{\mu\mathrm{M}} [A]\phi \textit{ iff } \forall a \in A, \forall P'.P \xrightarrow{a} P' \textit{ implies } P' \models^{\rho}_{\mu\mathrm{M}} \phi,\\
P &\models^{\rho}_{\mu\mathrm{M}} \mu x.\phi \textit{ iff } P \in \bigcap\{\mathcal{A} \mid \{Q \mid Q \models^{\rho[x\rightsquigarrow\mathcal{A}]}_{\mu\mathrm{M}} \phi\} \subseteq \mathcal{A}\},
\end{aligned}
$$

where $P, P' \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ *and* $\mathcal{A} \subseteq \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$.

Note that the restriction that all formulae of $\mathcal{L}_{\mu}(Act)$ are positive guarantees that the interpretation of a formula of the form $\mu x.\phi$ is well defined by the Tarski-Knaster Theorem [10].

Since the meaning of a closed formula ϕ is independent of valuations, we will abbreviate $P \models^{\rho}_{\mu\mathrm{M}} \phi$ as $P \models_{\mu\mathrm{M}} \phi$ for any valuation ρ.

The following derived operators are useful:

$$
\begin{aligned}
ff &\;\widehat{=}\; \neg tt,\\
\phi_1 \wedge \phi_2 &\;\widehat{=}\; \neg((\neg\phi_1) \vee (\neg\phi_2)),\\
\phi_1 \Rightarrow \phi_2 &\;\widehat{=}\; (\neg\phi_1) \vee \phi_2,\\
\phi_1 \Leftrightarrow \phi_2 &\;\widehat{=}\; (\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_1),\\
\nu x.\phi &\;\widehat{=}\; \neg(\mu x.\neg\phi\{\neg x/x\}).
\end{aligned}
$$

Convention: In order to improve the readability, in the later, we assume the binding precedence among the operators as "$\neg$" > "$\vee$" = "$\wedge$" > "μx" = "νx" > "$\Rightarrow$" = "$\Leftrightarrow$".

3.2 μM$^+$

μM$^+$ is an extension of μM with the non-deterministic choice "+". Informally, $\phi+\psi$ holds in a process P means that there exist P_1 and P_2 such that $P \sim P_1+P_2$, P_1 satisfies ϕ and P_2 meets ψ.

Given a set Act of atomic actions and a set $\mathcal{X}$ of variables, formulae of μM$^+$ are generated as follows:

$$\phi \;::=\; tt \mid x \mid \neg\phi \mid \phi \vee \phi \mid \langle A\rangle\phi \mid [A]\phi \mid \phi + \phi \mid \mu x.\phi,$$

where $x \in \mathcal{X}$ and $A \subseteq Act$.

Some notions for μM$^+$ can be defined same as in μM. We will use $\mathcal{L}^{+}_{\mu}(Act)$ to denote the language of formulae of μM$^+$ that are guarded and positive and $c\mathcal{L}^{+}_{\mu}(Act)$ to stand for the set of closed formulae in $\mathcal{L}^{+}_{\mu}(Act)$.

Definition 3. *A formula* $\phi \in \mathcal{L}_\mu^+(Act)$ *is called* strictly guarded*, if each variable* $x \in fn(\phi) \cup bn(\phi)$ *is guarded and does not occur in any sub-formula of the forms* $x + \psi$ *or* $\neg x + \psi$*.*

Note that strictly guarded is stronger than guarded, for instance, $\langle A\rangle(x + y)$ is guarded, but not strictly guarded.

Definition 4. *The semantics of* $\mathcal{L}_\mu^+(Act)$ *under a given valuation* ρ *is given by a satisfaction relation between* $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ *and* $\mathcal{L}_\mu^+(Act)$ *relative to* ρ*, denoted by* $\models^\rho_{\mu\mathrm{M}^+}$*. The definition of* $\models^\rho_{\mu\mathrm{M}^+}$ *contains all clauses listed in Definition 2, in addition to including the following clause for interpreting "+":*

$$P \models^\rho_{\mu\mathrm{M}^+} \phi_1 + \phi_2 \ \text{ iff } \ \exists P_1 \exists P_2 . P \sim P_1 + P_2, P_1 \models^\rho_{\mu\mathrm{M}^+} \phi_1 \text{ and } P_2 \models^\rho_{\mu\mathrm{M}^+} \phi_2,$$

where $P, P_1, P_2 \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$*.*

Since the meaning of a closed formula ϕ is independent of valuations, we will abbreviate $P \models^\rho_{\mu\mathrm{M}^+} \phi$ as $P \models_{\mu\mathrm{M}^+} \phi$ for any valuation ρ. A formula ϕ is valid, written $\models_{\mu\mathrm{M}^+} \phi$, if $P \models^\rho_{\mu\mathrm{M}^+} \phi$ for any $P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ and any valuation ρ. Sometimes, we write ϕ directly instead of $\models_{\mu\mathrm{M}^+} \phi$ for simplicity.

It is clear that $\mathcal{L}_\mu(Act) \subseteq \mathcal{L}_\mu^+(Act)$ and $c\mathcal{L}_\mu(Act) \subseteq c\mathcal{L}_\mu^+(Act)$.

Convention We will assume that "+" has a priority over all other binary operators, but "¬" has a higher priority to it. Given a set $A \subset B$, we will use $\bar{A}$ to stand for the complement $B - A$.

3.3 Some Results on μM and μM$^+$

From Definition 4, it is easy to see that "+" is monotonic. That is,

Proposition 1. *If* $\phi_1 \Rightarrow \phi_2$ *and* $\psi_1 \Rightarrow \psi_2$ *then* $\phi_1 + \psi_1 \Rightarrow \phi_2 + \psi_2$*.*

Definition 5. *Given a set of process* $\mathcal{A} \subseteq \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$*,* $\mathcal{A}$ *is* bisimulation closed *if* $\forall P \in \mathcal{A}$ *and* $\forall Q \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$*,* $P \sim Q$ *implies that* $Q \in \mathcal{A}$*. For convenience, from now on, we will abbreviate* bisimulation closed *as* B.C.*. A valuation* ρ *is* B.C. *if for all* $x \in \mathcal{X}$ $\rho(x)$ *is* B.C.*.*

Regarding the above definition, we have the following results:

Lemma 2. *If* $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ *are* B.C.*, then*

1. $\bar{\mathcal{A}}_1, \mathcal{A}_1 \cap \mathcal{A}_2$ *and* $\mathcal{A}_1 \cup \mathcal{A}_2$ *are* B.C.*,*
2. $\{P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}] \mid if\, P \xrightarrow{a} P' \text{ and } a \in A \text{ then } P' \in \mathcal{A}_1\}$ *is* B.C.*,*
3. $\{P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}] \mid \exists P' \in \mathcal{A}_1 . \exists a \in A . P \xrightarrow{a} P'\}$ *is* B.C.*,*
4. $\mathcal{A}_1 + \mathcal{A}_2$ *is* B.C.*, where* $\mathcal{A}_1 + \mathcal{A}_2$ *denotes the set* $\{P \mid \exists P_1 \in \mathcal{A}_1 . \exists P_2 \in \mathcal{A}_2 . P \sim P_1 + P_2\}$*.*

For any set of processes $\mathcal{A} \subseteq \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, we can associate with it the following subset:

$$\mathcal{A}^d \ \widehat{=} \ \{P \in \mathcal{A} \mid \text{ if } P \sim Q \text{ and } Q \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}] \text{ then } Q \in \mathcal{A}\}.$$

The set $\mathcal{A}^d$ is the largest bisimulation closed set contained in $\mathcal{A}$.

Lemma 3. *For any set* $\mathcal{A}, \mathcal{A}_i \subseteq \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ *for* $i = 1, 2$,

1. $\mathcal{A}^d$ *is* B.C.,
2. $\mathcal{A}^d \subseteq \mathcal{A}$,
3. $\mathcal{A}^d = \mathcal{A}$ *if* $\mathcal{A}$ *is* B.C.,
4. $\mathcal{A}_1^d \subseteq \mathcal{A}_2^d$ *if* $\mathcal{A}_1 \subseteq \mathcal{A}_2$,
5. $\mathcal{A}_1^d + \mathcal{A}_2^d \subseteq \mathcal{A}^d$ *if* $\mathcal{A}_1 + \mathcal{A}_2 \subseteq \mathcal{A}$.

We use ρ^d to stand for the valuation defined by $\rho^d(x) = \rho(x)^d$. By Lemma 3, it is clear that ρ^d is *B.C.* for any valuation ρ. From now on, we will use BCV to stand for the set of bisimulation closed valuations.

In the following, we will use $\llbracket \phi \rrbracket_\rho$ to denote the set of processes that meet ϕ under the valuation ρ, i.e., $\llbracket \phi \rrbracket_\rho \mathrel{\widehat{=}} \{P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}] \mid P \models^\rho_{\mu\mathrm{M}^+} \phi\}$. We will write $\rho \subseteq \rho'$ if $\rho(x) \subseteq \rho'(x)$ for any $x \in \mathcal{X}$.

Proposition 2. *For any* $\phi \in \mathcal{L}_\mu^+(Act)$, *if* ϕ *is strongly positive and* $\rho \subseteq \rho'$, *then* $\llbracket \phi \rrbracket_\rho \subseteq \llbracket \phi \rrbracket_{\rho'}$.

Lemma 4. *For any* $\phi \in \mathcal{L}_\mu^+(Act)$ *which is strongly positive, any valuation* ρ, *and* $\mathcal{A} \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, *then*

1. *If* ρ *is* B.C., *then* $\llbracket \phi \rrbracket_\rho$ *is* B.C. *as well;*
2. $\llbracket \phi \rrbracket_{\rho^d} \subseteq \mathcal{A}^d$ *if* $\llbracket \phi \rrbracket_\rho \subseteq \mathcal{A}$.

Proof. Similar to the proof for Proposition 3 in Section 5.4 in [9], simultaneously proving these two statements by induction on ϕ, the proof is done. $\dashv$

As [9] pointed out that each formula of $c\mathcal{L}_\mu(Act)$ defines a bisimulation invariant property, the following theorem indicates that every formula in $c\mathcal{L}_\mu^+(Act)$ is bisimulation invariant as well. The forward direction of the theorem follows immediately from the above lemma; the converse direction comes from the fact $c\mathcal{L}_\mu(Act) \subseteq c\mathcal{L}_\mu^+(Act)$.

Theorem 1. *For any* $P, Q \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, $P \sim Q$ *iff for each* $\phi \in c\mathcal{L}_\mu^+(Act)$, $P \models_{\mu\mathrm{M}^+} \phi$ *iff* $Q \models_{\mu\mathrm{M}^+} \phi$.

The following lemmas can be proved by Definition 4.

Lemma 5.

(1) $\phi + ff \;\Leftrightarrow\; ff$
(2) $tt + tt \;\Leftrightarrow\; tt$
(3) $[A]tt \;\Leftrightarrow\; tt$
(4) $\langle A\rangle ff \;\Leftrightarrow\; ff$
(5) $\phi + \psi \;\Leftrightarrow\; \psi + \phi$
(6) $(\phi + \psi) + \varphi \;\Leftrightarrow\; \phi + (\psi + \varphi)$
(7) $\langle A\rangle\phi_1 \wedge [A \cup B]\phi_2 \Rightarrow \langle A\rangle(\phi_1 \wedge \phi_2)$
(8) $\phi + (\varphi \vee \psi) \Leftrightarrow (\phi + \varphi) \vee (\phi + \psi)$
(9) $\langle A\rangle\phi_1 \vee \langle A\rangle\phi_2 \Leftrightarrow \langle A\rangle(\phi_1 \vee \phi_2)$
(10) $[A]\phi_1 \wedge [A]\phi_2 \Leftrightarrow [A](\phi_1 \wedge \phi_2)$
(11) $\langle A_1 \cup A_2\rangle\phi \Leftrightarrow \langle A_1\rangle\phi \vee \langle A_2\rangle\phi$
(12) $[A_1 \cup A_2]\phi \Leftrightarrow [A_1]\phi \wedge [A_2]\phi$

Lemma 6.
(1) $\neg[A]\phi \Leftrightarrow \langle A\rangle\neg\phi$
(2) $\neg\langle A\rangle\phi \Leftrightarrow [A]\neg\phi$
(3) $[A_1]\phi_1 \wedge [A_2]\phi_2 \Leftrightarrow [A_1 - (A_1 \cap A_2)]\phi_1 \wedge [A_1 \cap A_2](\phi_1 \wedge \phi_2) \wedge [A_2 - (A_1 \cap A_2)]\phi_2$

4 Reducing $c\mathcal{L}_\mu^+(Act)$ to $c\mathcal{L}_\mu(Act)$

In this section, we show that "+" is definable in μM by reducing $c\mathcal{L}_\mu^+(Act)$ into $c\mathcal{L}_\mu(Act)$. The encoding is completed via the following three steps: firstly, we prove that in some special cases, "+" can be defined by conjunction and disjunction; then we show that the problem of eliminating "+" in a strongly positive and strictly guarded formula ϕ can be reduced to one of the above special cases; and finally we complete the encoding by proving that for any $\phi \in c\mathcal{L}_\mu^+(Act)$, there is a formula $\phi' \in c\mathcal{L}_\mu^+(Act)$ which is strictly guarded such that $\phi \Leftrightarrow \phi'$.

We say that ϕ implies ψ w.r.t. bisimulation closed valuations, denoted by $\phi \overset{bc}{\Rightarrow} \psi$, if $[\![\phi]\!]_\rho \subseteq [\![\psi]\!]_\rho$ for any $\rho \in BCV$. $\phi \overset{bc}{\Leftrightarrow} \psi$ means $\phi \overset{bc}{\Rightarrow} \psi$ and $\psi \overset{bc}{\Rightarrow} \phi$. It is clear that $\phi \Rightarrow \psi$ implies $\phi \overset{bc}{\Rightarrow} \psi$, and $\phi \overset{bc}{\Rightarrow} \psi$ iff $\phi \Rightarrow \psi$ if $\phi, \psi \in c\mathcal{L}_\mu^+(Act)$.

In order to attain the first step, we need the following proposition:

Proposition 3. **1.** *For any $P, Q \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, if $P \models^\rho_{\mu\mathrm{M}^+} \langle A\rangle\phi$ then $P + Q \models^\rho_{\mu\mathrm{M}^+} \langle A\rangle\phi$; and*
2. *If $P \models^\rho_{\mu\mathrm{M}^+} [A]\phi_1$ and $Q \models^\rho_{\mu\mathrm{M}^+} [A]\phi_2$ then $P + Q \models^\rho_{\mu\mathrm{M}^+} [A](\phi_1 \vee \phi_2)$.*

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

Lemma 7.

$$\Big(\bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n_i}\langle A_i\rangle\phi_{i,j} \wedge \bigwedge_{i=1}^{m}[B_i]\psi_i\Big) + \Big(\bigwedge_{i=1}^{k}\bigwedge_{j=1}^{k_i}\langle C_i\rangle\varphi_{i,j} \wedge \bigwedge_{i=1}^{m}[B_i]\chi_i\Big)$$
$$\overset{bc}{\Leftrightarrow} \bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n_i}\langle A_i\rangle(\phi_{i,j} \wedge \psi_i) \wedge \bigwedge_{i=1}^{k}\bigwedge_{j=1}^{k_i}\langle C_i\rangle(\varphi_{i,j} \wedge \chi_i) \wedge \bigwedge_{i=1}^{m}[B_i](\psi_i \vee \chi_i)$$

where all conjuncts in the formula of the left side of $\overset{bc}{\Leftrightarrow}$ are strongly positive, $n, k \leq m$, $\forall 1 \leq i \leq n. A_i \subseteq B_i$, $\forall 1 \leq i \leq k. C_i \subseteq B_i$, and for any $1 \leq i, j \leq m$, if $i \neq j$ then $B_i \cap B_j = \emptyset$.

Proof. "$\overset{bc}{\Rightarrow}$" can be easily proved by Proposition 3 and Lemma 4. So, we only give a sketch for the proof of the converse direction. Assume

$$P \models^\rho_{\mu\mathrm{M}^+} \bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n_i}\langle A_i\rangle(\phi_{i,j} \wedge \psi_i) \wedge \bigwedge_{i=1}^{k}\bigwedge_{j=1}^{k_i}\langle C_i\rangle(\varphi_{i,j} \wedge \chi_i) \wedge \bigwedge_{i=1}^{m}[B_i](\psi_i \vee \chi_i), \quad (1)$$

where ρ is $B.C.$. By Lemma 1, $P \sim \Sigma_{i=1}^{l}\Sigma_{j=1}^{i_{a_i}} a_i P_{i,j}$, where $l \geq m$ and for any $1 \leq i, j \leq l$, if $i \neq j$ then $a_i \neq a_j$. So, we have $\Sigma_{i=1}^{l}\Sigma_{j=1}^{i_{a_i}} a_i P_{i,j} \models^\rho_{\mu\mathrm{M}^+} \bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n_i}\langle A_i\rangle(\phi_{i,j} \wedge \psi_i)$ by Lemma 4. This implies that for each $1 \leq i \leq n$ and

$1 \le j \le n_i$, there exist $1 \le r_i \le l$ and $1 \le h_j \le i_{a_{r_i}}$ such that $a_{r_i} \in A_i$ and $P_{r_i,h_j} \models^{\rho}_{\mu\mathrm{M}^+} \phi_{i,j} \wedge \psi_i$. Let $P' \mathrel{\widehat{=}} \Sigma_{i=1}^{n} \Sigma_{j=1}^{n_i} a_{r_i} P_{r_i,h_j}$. It is obvious that

$$P' \models^{\rho}_{\mu\mathrm{M}^+} \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n_i} \langle A_i \rangle \phi_{i,j} \wedge \bigwedge_{i=1}^{m} [B_i] \psi_i. \tag{2}$$

Similarly, we get that for each $1 \le i \le k$ and $1 \le j \le k_i$, there exist $1 \le r_i \le l$ and $1 \le h_j \le i_{a_{r_i}}$ such that $a_{r_i} \in C_i$ and $P_{r_i,h_j} \models^{\rho}_{\mu\mathrm{M}^+} \varphi_{i,j} \wedge \chi_i$. Let $P'' \mathrel{\widehat{=}} \Sigma_{i=1}^{k} \Sigma_{j=1}^{n_i} a_{r_i} P_{r_i,h_j}$. It is easy to show that

$$P'' \models^{\rho}_{\mu\mathrm{M}^+} \bigwedge_{i=1}^{k} \bigwedge_{j=1}^{k_i} \langle C_i \rangle \varphi_{i,j} \wedge \bigwedge_{i=1}^{m} [B_i] \chi_i. \tag{3}$$

Then, we add each summand of $\Sigma_{i=1}^{l} \Sigma_{j=1}^{i_{a_i}} a_i P_{i,j}$ to P' or P'' according to the following algorithm: For each $1 \le i \le l$, if $a_i \in B_j$ for some $j \in \{1, \ldots, m\}$ then let $I_1 \mathrel{\widehat{=}} \{h \mid P_{i,h} \models \psi_j\}$ and $I_2 \mathrel{\widehat{=}} \{h \mid P_{i,h} \models \chi_j\}$; otherwise, $I_1 \mathrel{\widehat{=}} \{1, \ldots, i_{a_i}\}$ and $I_2 = \emptyset$. Since $P \models^{\rho}_{\mu\mathrm{M}^+} [B_j](\psi_j \vee \chi_j)$, it is clear that $I_1 \cup I_2 = \{1, \ldots, i_{a_i}\}$. Then, let $P' := P' + \sum_{h \in I_1} a_i P_{i,h}$ and $P'' := P'' + \sum_{h \in I_2} a_i P_{i,h}$. Because $B_i \cap B_j = \emptyset$ if $i \neq j$, it is easy to show that (2) and (3) keep invariant for each cojoining.

Additionally, it is easy to see that $P' + P'' \sim P$. Hence, from Lemma 4,

$$P \models^{\rho}_{\mu\mathrm{M}^+} (\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n_i} \langle A_i \rangle \phi_{i,j} \wedge \bigwedge_{i=1}^{m} [B_i] \psi_i) + (\bigwedge_{i=1}^{k} \bigwedge_{j=1}^{k_i} \langle C_i \rangle \varphi_{i,j} \wedge \bigwedge_{i=1}^{m} [B_i] \chi_i). \qquad \dashv$$

Furthermore, applying the above lemma, we can complete the second step by proving the following results:

Lemma 8. *For any $\phi \in \mathcal{L}^{+}_{\mu}(Act)$, if ϕ is strictly guarded and strongly positive, then there exists ϕ' in which no $+$ occurs such that $\phi' \overset{bc}{\Leftrightarrow} \phi$ and ϕ' is strictly guarded and strongly positive.*

Proof. By induction on the structure of ϕ. Here, we only list the proofs for some interesting cases.

. $\phi = \neg\psi$

Suppose $\rho \in BCV$ and $fn(\phi) \subseteq \{x_1, \ldots, x_n\}$. Let $\neg\rho$ be defined by $\neg\rho(x) = \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}] - \rho(x)$ for any $x \in \mathcal{X}$. By Lemma 2.1., $\neg\rho$ is *B.C.*. It is easy to see that $[\![\varphi]\!]_{\rho} = [\![\varphi\{\neg x_1/x_1, \ldots, \neg x_n/x_n\}]\!]_{\neg\rho}$ for any $\varphi \in \mathcal{L}^{+}_{\mu}(Act)$ whose free variables are in $\{x_1, \ldots, x_n\}$.

Since ϕ is strictly guarded and strongly positive, so is $\psi\{\neg x_1/x_1, \ldots, \neg x_n/x_n\}$. By the induction hypothesis, there is ψ' in which no $+$ occurs such that $\psi\{\neg x_1/x_1, \ldots, \neg x_n/x_n\} \overset{bc}{\Leftrightarrow} \psi'$ and ψ' is strictly guarded and strongly positive. Besides,

$$\begin{aligned}[\![\phi]\!]_\rho &= \mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}] - [\![\psi]\!]_\rho \\ &= \mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}] - [\![\psi\{\neg x_1/x_1,\ldots,\neg x_n/x_n\}]\!]_{\neg\rho} \\ &= \mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}] - [\![\psi'\{\neg x_1/x_1,\ldots,\neg x_n/x_n\}]\!]_\rho \\ &= [\![\neg\psi'\{\neg x_1/x_1,\ldots,\neg x_n/x_n\}]\!]_\rho\end{aligned}$$

Hence, let $\phi' \widehat{=} \neg\psi'\{\neg x_1/x_1,\ldots,\neg x_n/x_n\}$. It is obvious that no $+$ occurs in ϕ', ϕ' is strictly guarded and strongly positive and $\phi \overset{bc}{\Leftrightarrow} \phi'$.

. $\phi = \langle A\rangle\phi_1$
As ϕ is strictly guarded and strongly positive, this implies the following two cases:
1. ϕ_1 is equivalent to a disjunction of some formulae of the form $x_1 \wedge \cdots x_n \wedge \chi_1 \wedge \cdots \wedge \chi_\ell$, where $n,\ell \geq 0$, $x_1,\cdots,x_n \in Var$, and for each $1 \leq i \leq \ell$, $\chi_i \in \mathcal{L}_\mu(Act)$ which is strictly guarded and strongly positive;
2. ϕ_1 is strictly guarded and strongly positive.
In either of the two cases, by the induction hypothesis, it is easy to construct a formula ϕ' in which no $+$ occurs such that ϕ' is strictly guarded and $\phi' \overset{bc}{\Leftrightarrow} \phi$.

. $\phi = \phi_1 + \phi_2$
Since ϕ is strictly guarded and strongly positive, so are ϕ_1 and ϕ_2. By the induction hypothesis, there exist ϕ_i' such that ϕ_i' is strongly positive and strictly guarded, $\phi_i' \overset{bc}{\Leftrightarrow} \phi_i$ and no $+$ occurs in ϕ_i' for $i = 1,2$.
We consider the following two cases:

1. $\phi_1' \overset{bc}{\Leftrightarrow} ff$ or $\phi_2' \overset{bc}{\Leftrightarrow} ff$. If so, let $\phi' \widehat{=} ff$. By Lemma 5.(1), we have that $\phi_1' + \phi_2' \overset{bc}{\Leftrightarrow} ff$. On the other hand, by Proposition 1, it follows that $\phi \overset{bc}{\Leftrightarrow} ff$. Hence, ϕ' is what we want.
2. $\phi_1' \overset{bc}{\not\Leftrightarrow} ff$ and $\phi_2' \overset{bc}{\not\Leftrightarrow} ff$. Using the laws of Boolean Algebra, Lemma 5.9–12 and Lemma 6, we can transform ϕ_1' and ϕ_2' equivalently as follows:

$$\phi_1' \Leftrightarrow \bigvee_{i=1}^{m_1}(\bigwedge_{j=1}^{m_{1,i}} \bigwedge_{h=1}^{m_{1,i,j}} \langle A_{1,i,j}\rangle\phi_{1,i,j,h} \wedge \bigwedge_{j=1}^{m'_{1,i}} [B_{1,i,j}]\psi_{1,i,j}), \tag{4}$$

$$\phi_2' \Leftrightarrow \bigvee_{i=1}^{m_2}(\bigwedge_{j=1}^{m_{2,i}} \bigwedge_{h=1}^{m_{2,i,j}} \langle A_{2,i,j}\rangle\phi_{2,i,j,h} \wedge \bigwedge_{j=1}^{m'_{2,i}} [B_{2,i,j}]\psi_{2,i,j}), \tag{5}$$

where

. $\forall 1 \leq i \leq 2, \forall 1 \leq j \leq m_i.(\forall 1 \leq k_1,k_2 \leq m_{i,j}.k_1 \neq k_2 \Rightarrow A_{i,j,k_1} \cap A_{i,j,k_2} = \emptyset) \wedge (\forall 1 \leq k_1,k_2 \leq m'_{i,j}.k_1 \neq k_2 \Rightarrow B_{i,j,k_1} \cap B_{i,j,k_2} = \emptyset) \wedge (\forall 1 \leq k_1 \leq m_{i,j}, \forall 1 \leq k_2 \leq m'_{i,j}.A_{i,j,k_1} \subseteq B_{i,j,k_2} \vee A_{i,j,k_1} \cap B_{i,j,k_2} = \emptyset))$;

. $B_{1,i_1,j_1} = B_{2,i_2,j_2}$ or $B_{1,i_1,j_1} \cap B_{2,i_2,j_2} = \emptyset$ for all $1 \leq i_1 \leq m_1$, $1 \leq j_1 \leq m'_{1,i_1}$, $1 \leq i_2 \leq m_2$, $1 \leq j_2 \leq m'_{1,i_2}$;

- for all $i = 1, 2$, $1 \le j_1 \le m_i$, $1 \le k_1 \le m_{i,j_1}$ $1 \le j_2 \le m_{3-i}$, $1 \le k_2 \le m'_{3-i,j_2}$, $A_{i,j_1,j_2} \subseteq B_{3-i,j_2,k_2}$ or $A_{i,j_1,j_2} \cap B_{3-i,j_2,k_2} = \emptyset$.

By Lemma 5.5–8, we have

$$\phi'_1 + \phi'_2 \overset{bc}{\Leftrightarrow} \bigvee_{i_1=1}^{m_1} \bigvee_{i_2=1}^{m_2} \big(\bigwedge_{j=1}^{m_{1,i_1}} \bigwedge_{h=1}^{m_{1,i_1,j}} \langle A_{1,i_1,j} \rangle \phi_{1,i_1,j,h} \wedge \bigwedge_{j=1}^{m'_{1,i_1}} [B_{1,i_1,j}] \psi_{1,i_1,j} \big) + \big(\bigwedge_{j=1}^{m_{2,i_2}} \bigwedge_{h=1}^{m_{2,i_2,j}} \langle A_{2,i_2,j} \rangle \phi_{2,i_2,j,h} \wedge \bigwedge_{j=1}^{m'_{2,i_2}} [B_{2,i_2,j}] \psi_{2,i_2,j} \big) \quad (6)$$

Thus, according to Lemma 5 and Lemma 7, for each disjunct of the right hand of (6), there is a formula $\varphi_{i,j}$ that is equivalent to the disjunct w.r.t. BCV, strictly guarded, strongly positive and no + occurs in it, where $1 \le i \le m_1$ and $1 \le j \le m_2$. So, let $\phi' \widehat{=} \bigvee_{i=1}^{m_1} \bigvee_{j=1}^{m_2} \varphi_{i,j}$. It is easy to see that ϕ' meets the requirement.

- $\phi = \mu x.\phi_1$

Since ϕ is strictly guarded and strongly positive, so is ϕ_1. Therefore, by the induction hypothesis, there exists ϕ'_1 in which no + occurs such that ϕ'_1 is strictly guarded and strongly positive and $\phi'_1 \overset{bc}{\Leftrightarrow} \phi_1$. By Lemma 4, it is easy to see that $\mu x.\phi_1 \overset{bc}{\Leftrightarrow} \mu x.\phi'_1$. Thus, let $\phi' \widehat{=} \mu x.\phi'_1$. ⊣

Finally, in order to encode $c\mathcal{L}^+_\mu(Act)$ into $c\mathcal{L}_\mu(Act)$, we need to show the following lemma:

Lemma 9. *For any $\phi \in c\mathcal{L}^+_\mu(Act)$, there exists $\phi' \in c\mathcal{L}^+_\mu(Act)$ such that ϕ' is strictly guarded and $\phi \Leftrightarrow \phi'$.*

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

$$\mu x.\phi_1[\langle A \rangle \phi_2[(x \odot \phi_3) + \phi_4]] \Leftrightarrow \mu x.\phi_1[\langle A \rangle \phi_2[\mu y.(\phi_1[\langle A \rangle \phi_2[y]] \odot \phi_3) + \phi_4]] \quad (7)$$
$$\nu x.\phi_1[\langle A \rangle \phi_2[(x \odot \phi_3) + \phi_4]] \Leftrightarrow \nu x.\phi_1[\langle A \rangle \phi_2[\nu y.(\phi_1[\langle A \rangle \phi_2[y]] \odot \phi_3) + \phi_4]] \quad (8)$$
$$\mu x.\phi_1[[A] \phi_2[(x \odot \phi_3) + \phi_4]] \Leftrightarrow \mu x.\phi_1[[A] \phi_2[\mu y.(\phi_1[[A] \phi_2[y]] \odot \phi_3) + \phi_4]] \quad (9)$$
$$\nu x.\phi_1[[A] \phi_2[(x \odot \phi_3) + \phi_4]] \Leftrightarrow \nu x.\phi_1[[A] \phi_2[\nu y.(\phi_1[[A] \phi_2[y]] \odot \phi_3) + \phi_4]] \quad (10)$$
$$\mu x.\phi_1[\langle A \rangle \phi_2[(\neg x \odot \phi_3) + \phi_4]] \Leftrightarrow \mu x.\phi_1[\langle A \rangle \phi_2[\nu y.(\neg \phi_1[\langle A \rangle \phi_2[y]] \odot \phi_3) + \phi_4]] \quad (11)$$
$$\nu x.\phi_1[\langle A \rangle \phi_2[(\neg x \odot \phi_3) + \phi_4]] \Leftrightarrow \nu x.\phi_1[\langle A \rangle \phi_2[\mu y.(\neg \phi_1[\langle A \rangle \phi_2[y]] \odot \phi_3) + \phi_4]] \quad (12)$$
$$\mu x.\phi_1[[A] \phi_2[(\neg x \odot \phi_3) + \phi_4]] \Leftrightarrow \mu x.\phi_1[[A] \phi_2[\nu y.(\neg \phi_1[[A] \phi_2[y]] \odot \phi_3) + \phi_4]] \quad (13)$$
$$\nu x.\phi_1[[A] \phi_2[(\neg x \odot \phi_3) + \phi_4]] \Leftrightarrow \nu x.\phi_1[[A] \phi_2[\mu y.(\neg \phi_1[[A] \phi_2[y]] \odot \phi_3) + \phi_4]] \quad (14)$$

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 only prove (9) as an example, the others can be proved similarly.[1]

Since $\phi_1[[A]\phi_2[(x \odot \phi_3) + \phi_4]]$ is guarded, by Knaster-Tarski Theorem, it is clear that $\mu x.\phi_1[[A]\phi_2[(x \odot \phi_3) + \phi_4]]$ is the unique least solution of the equation

$$x = \phi_1[[A]\phi_2[(x \odot \phi_3) + \phi_4]] \quad (15)$$

[1] Note that in the proofs for (11)–(14), we need to let $\neg y = (\neg x \odot \phi_3) + \phi_4$ in order to guarantee the resulted formulae are still positive.

Let y be a fresh variable and $y = (x \odot \phi_3) + \phi_4$. It is easy to see the least solution of (11) is equivalent to the x-component of the least solution of the following equation system:

$$x = \phi_1[\,[A]\phi_2[(x \odot \phi_3) + \phi_4]]$$
$$y = (x \odot \phi_3) + \phi_4$$

Meanwhile, it is easy to rewrite the above equation system to the following one

$$x = \phi_1[\,[A]\phi_2[y]]$$
$$y = (\phi_1[\,[A]\phi_2[y]] \odot \phi_3) + \phi_4$$

It is not hard to derive the least solution of the above equation system as

$$(\mu x.\phi_1[\,[A]\phi_2[\mu y.(\phi_1[\,[A]\phi_2[y]] \odot \phi_3) + \phi_4]], \mu y.(\phi_1[\,[A]\phi_2[y]] \odot \phi_3) + \phi_4.$$

Therefore, (9) follows.

Repeatedly applying (7)–(14), for any given formula $\phi \in c\mathcal{L}^+_\mu(Act)$, we can rewrite it to ϕ' which is strictly guarded such that $\phi \Leftrightarrow \phi'$. ⊣

Example 1. Let $\phi = \mu x.\langle A\rangle x + \mu y.[C]\neg(\langle B\rangle\neg y + \neg x) \vee \langle C\rangle tt$, where $A \cap B = B \cap C = A \cap C = \emptyset$. Applying the rewriting rule (13), it results that

$$\begin{aligned}
\phi \Leftrightarrow\ & \mu x.\langle A\rangle x + \\
& \quad \mu y.([C]\neg(\nu z.\neg(\langle A\rangle x + \mu y'.([C]\neg z \vee \langle C\rangle tt)) + \langle B\rangle\neg y) \vee \langle C\rangle tt) \\
\Leftrightarrow\ & \mu x.\langle A\rangle x + \mu y.([C]\neg(\nu z.\neg(\langle A\rangle x + ([C]\neg z \vee \langle C\rangle tt)) + \langle B\rangle\neg y) \vee \langle C\rangle tt) \\
\Leftrightarrow\ & \mu x.\langle A\rangle x + \mu y.([C]\neg(\nu z.\neg\langle A\rangle x + \langle B\rangle\neg y) \vee \langle C\rangle tt) \\
\Leftrightarrow\ & \mu x.\langle A\rangle x + \mu y.([C]\neg(\neg\langle A\rangle x + \langle B\rangle\neg y) \vee \langle C\rangle tt) \\
\Leftrightarrow\ & \mu x.\langle A\rangle x + \mu y.([C][B]y \vee \langle C\rangle tt) \\
\Leftrightarrow\ & \mu x.\langle A\rangle x \vee (\langle A\rangle x \wedge \langle C\rangle tt) \\
\Leftrightarrow\ & \mu x.\langle A\rangle x,
\end{aligned}$$

where $\phi_1 = \langle A\rangle x + \mu y.([\ ] \vee \langle C\rangle tt), \phi_2 = \neg([\ ]), \phi_3 = tt, \phi_4 = \langle B\rangle\neg y$. ⊣

Note that in the above example, we can also unfold $\mu y.[C]\neg(\langle B\rangle\neg y + \neg x) \vee \langle C\rangle tt$ first, then apply Lemma 7 and obtain the same result.

Directly from Lemma 9 and Lemma 8, we can conclude:

Theorem 2. *$\forall\phi \in c\mathcal{L}^+_\mu(Act)$, $\exists\phi' \in c\mathcal{L}_\mu(Act).\phi \Leftrightarrow \phi'$.*

In the later, we will use En to denote the above implicit translating function from $c\mathcal{L}^+_\mu(Act)$ to $c\mathcal{L}_\mu(Act)$.

5 Synchronization Tree Logic

[2] proposed a logic, called *Synchronization Tree Logic* (STL) for the specification and proof of programs, described by $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$. Formulae of STL can

be obtained from the constants ϵ, $\top$ by using *logical connectives*, consistent extensions of the operators $a \in Act$, $+$ and *fixpoint operators*. Therefore, STL contains $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, i.e., terms of $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ are formulae of STL if we look recursive operators of $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ as greatest fixpoint operators. Its semantics is defined by associating with a formula a set of terms (synchronization trees) representing unions of congruence classes of the strong congruence relation.

Given a set Act of atomic actions and a set $\mathcal{X}$ of variables, formulae of STL are constructed by the rule:

$$\phi ::= \epsilon \mid \top \mid x \mid \neg\phi \mid B\phi \mid \phi + \phi' \mid \phi \vee \phi' \mid \mu x.\phi,$$

where $x \in \mathcal{X}$ and $B \subseteq Act$.

In what follows, we will use $\mathcal{L}_{\mathrm{STL}}(Act)$ to stand for the set of formulae of STL that are guarded and positive and $c\mathcal{L}_{\mathrm{STL}}(Act)$ for the subset of $\mathcal{L}_{\mathrm{STL}}(Act)$ in which all formulae are closed.

Definition 6. *Given a valuation $\rho \in BCV$, the semantics of $\mathcal{L}_{\mathrm{STL}}(Act)$ is given by a satisfaction relation between $\mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ and $\mathcal{L}_{\mathrm{STL}}(Act)$ relative to ρ, denoted by $\models^{\rho}_{\mathrm{STL}}$, inductively defined as follows:*

$$\begin{aligned}
&P \models^{\rho}_{\mathrm{STL}} \top,\\
&P \models^{\rho}_{\mathrm{STL}} \epsilon \textit{ iff } P \sim \epsilon,\\
&P \models^{\rho}_{\mathrm{STL}} \neg\phi \textit{ iff } P \not\models^{\rho}_{\mathrm{STL}} \phi,\\
&P \models^{\rho}_{\mathrm{STL}} B\phi \textit{ iff } \exists I \subseteq \mathbb{N}. I \neq \emptyset, I \textit{ is finite},\\
&\qquad\qquad \forall i \in I(\exists a_i \in B \textit{ and } \exists P_i.P_i \models^{\rho}_{\mathrm{STL}} \phi), P \sim \Sigma_{i\in I} a_i P_i,\\
&P \models^{\rho}_{\mathrm{STL}} \phi_1 \vee \phi_2 \textit{ iff } P \models^{\rho}_{\mathrm{STL}} \phi_1 \textit{ or } P \models^{\rho}_{\mathrm{STL}} \phi_2,\\
&P \models^{\rho}_{\mathrm{STL}} \phi_1 + \phi_2 \textit{ iff } \exists P_1, P_2.P_1 \models^{\rho}_{\mathrm{STL}} \phi_1, P_2 \models^{\rho}_{\mathrm{STL}} \phi_2 \textit{ and } P \sim P_1 + P_2,\\
&P \models^{\rho}_{\mathrm{STL}} \mu x.\phi \textit{ iff } P \in \bigcap\{\mathcal{A} \mid \mathcal{A} \textit{ is } \mathrm{B.C.} \textit{ and } [\![\, \phi \,]\!]_{\rho[x \rightsquigarrow \mathcal{A}]} \subseteq \mathcal{A}\},
\end{aligned}$$

where $\mathcal{A} \subseteq \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}], B \subseteq Act$.

Some notions and derived operators can be defined similarly as in μM and μM$^+$. In what follows we will use $\bot$ to denote $\neg\top$. *Note that* in STL all valuations are restricted to be in BCV.

[2] proved the following results:

Proposition 4. $[\![\, \phi \,]\!]_\rho$ *is* B.C.*, for any $\rho \in BCV$ and $\phi \in \mathcal{L}_{\mathrm{STL}}(Act)$.*

Proposition 5. *For each $P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$,*

$$[\![\, \phi_P \,]\!] = \{P' \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}] \mid P \sim P'\}.$$

More results on STL can be found in [2].

6 Reducing $c\mathcal{L}_{\mathrm{STL}}(Act)$ to $c\mathcal{L}_{\mu}(Act)$

In this section, we define a function $Tr : \mathcal{L}_{\mathrm{STL}}(Act) \rightarrow \mathcal{L}_{\mu}^{+}(Act)$ such that for any $\phi \in \mathcal{L}_{\mathrm{STL}}(Act)$, $P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ and $\rho \in BCV$, $P \models_{\mathrm{STL}}^{\rho} \phi$ iff $P \models_{\mu\mathrm{M}^{+}}^{\rho} Tr(\phi)$. Moreover, according to Theorem 2, for each $\phi \in c\mathcal{L}_{\mathrm{STL}}(Act)$ and $P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, $P \models_{\mu\mathrm{M}^{+}} Tr(\phi)$ iff $P \models_{\mu\mathrm{M}} En(Tr(\phi))$. Thus, this completes the reduction from $c\mathcal{L}_{\mathrm{STL}}(Act)$ to $c\mathcal{L}_{\mu}(Act)$.

Definition 7. *The function Tr is inductively defined as follows: $Tr(\bot) \widehat{=} ff$, $Tr(\top) \widehat{=} tt$, $Tr(x) \widehat{=} x$, $Tr(\epsilon) \widehat{=} [Act]ff$, $Tr(\neg\phi) \widehat{=} \neg Tr(\phi)$, $Tr(B\phi) \widehat{=} [B]Tr(\phi) \wedge [\bar{B}]ff \wedge \langle B\rangle Tr(\phi)$, $Tr(\phi_1 \vee \phi_2) \widehat{=} Tr(\phi_1) \vee Tr(\phi_2)$, $Tr(\phi_1 + \phi_2) \widehat{=} Tr(\phi_1) + Tr(\phi_2)$, $Tr(\mu x.\phi) \widehat{=} \mu x.Tr(\phi)$.*

Theorem 3. *For any $P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$ and $\phi \in \mathcal{L}_{\mathrm{STL}}(Act)$, $Tr(\phi) \in \mathcal{L}_{\mu}^{+}(Act)$ and $P \models_{\mathrm{STL}}^{\rho} \phi$ iff $P \models_{\mu\mathrm{M}^{+}}^{\rho} Tr(\phi)$. Where $\rho \in BCV$.*

Proof. $Tr(\phi) \in \mathcal{L}_{\mu}^{+}(Act)$ is obvious by Definition 7, the proof for the second part can proceed by induction on the structure of ϕ. ⊣

The following theorem that follows directly from Theorem 3 and Theorem 2 indicates that applying Tr and En, STL can be translated into μM.

Theorem 4. *For all $P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, $\phi \in c\mathcal{L}_{\mathrm{STL}}(Act)$, $En(Tr(\phi)) \in c\mathcal{L}_{\mu}(Act)$ and $P \models_{\mathrm{STL}} \phi$ iff $P \models_{\mu\mathrm{M}} En(Tr(\phi))$.*

Corollary 1. *For any $P, Q \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, $Q \models_{\mu\mathrm{M}} En(Tr(\phi_P))$ iff $P \sim Q$.*

Below we present an example to show how to translate a formula $\phi \in c\mathcal{L}_{\mathrm{STL}}(Act)$ into $c\mathcal{L}_{\mu}(Act)$, and indicate that for any $P \in \mathcal{T}[\{\epsilon\}, \{+\}, Act, \mathcal{X}]$, $En(Tr(\phi_P))$ is exactly the characteristic formula of P up to $\sim$. Given an equivalence or preorder $\preceq$ over processes, the characteristic formula for a process P up to it is a formula ϕ_P such that given a process Q, $Q \models \phi_P$ if and only if $Q \preceq P$.

Example 2. Suppose $Act = \{a, b, c\}$, $P \widehat{=} rec\, x.(a\, b\, x + a\, c\, \epsilon)$ and $Q \widehat{=} rec\, x.[a\, (b\, x + c\, \epsilon)]$. Thus, by Definition 7,

$$\begin{aligned} Tr(\phi_P) \Leftrightarrow \nu x.[&\langle a\rangle(\langle b\rangle x \wedge [\{\bar{b}\}]ff \wedge [b]x) \wedge [\{\bar{a}\}]ff \wedge [a](\langle b\rangle x \wedge [\{\bar{b}\}]ff \wedge [b]x)] \\ &+ [\langle a\rangle(\langle c\rangle[Act]ff \wedge [\{\bar{c}\}]ff \wedge [c][Act]ff) \wedge [\{\bar{a}\}]ff \wedge [a](\langle c\rangle[Act]ff \\ &\wedge [\{\bar{c}\}]ff \wedge [c][Act]ff)] \end{aligned}$$

Moreover, we can get

$$\begin{aligned} En(Tr(\phi_P)) \Leftrightarrow \nu x.&\langle a\rangle(\langle b\rangle x \wedge [\{\bar{b}\}]ff \wedge [b]x) \wedge \langle a\rangle(\langle c\rangle[Act]ff \wedge [\{\bar{c}\}]ff \\ &\wedge [c][Act]ff) \wedge [\{\bar{a}\}]ff \wedge [a]((\langle b\rangle x \wedge [\{\bar{b}\}]ff \wedge [b]x) \\ &\vee (\langle c\rangle[Act]ff \wedge [\{\bar{c}\}]ff \wedge [c][Act]ff)) \end{aligned}$$

It is easy to see that $En(Tr(\phi_P))$ is exactly the characteristic formula of P and $Q \not\models_{\mu\mathrm{M}} En(Tr(\phi_P))$ since $P \not\sim Q$. ⊣

7 Concluding Remarks

In this paper, we investigated the definability of the non-deterministic operator $+$ introduced in STL as a primitive in the μ-calculus. This was captured via extending the μ-calculus with the non-deterministic operator $+$ to $\mu\mathrm{M}^+$ first and then showing that $\mu\mathrm{M}^+$ can be encoded into the modal μ-calculus.

Furthermore, we proved that STL can be translated into the modal μ-calculus by encoding it into $\mu\mathrm{M}^+$. Thus, if *Act* is finite, we can get the decidability of STL by the decidability of the μ-calculus [5]. In fact, we could translate other STL-like modal logics into the μ-calculus, for example, it is easy to encode the modal process logic presented in [6] into the μ-calculus according to the results shown in this paper.

The converse procedure to translate $\mathcal{L}_\mu(Act)$ into $\mathcal{L}_{\mathrm{STL}}(Act)$ can be obtained easily. Thus, we see that the μ-calculus is as expressive as STL.

In summary, the significance of this work lies in:

- We proved that the non-deterministic choice $+$ is definable in the μ-calculus, so that we can compare the expressiveness between the μ-calculus with process algebra-like modal logics such as STL, for example, it was shown in this paper that the μ-calculus is as expressive as STL.
- A connection between the connectives of the μ-calculus and the operators of $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$ has been established in this paper. This thus makes it possible that syntax-directed proofs for programs defined in terms of $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$ can be done in the μ-calculus;
- We indirectly presented an algorithm to construct the characteristic formula up to $\sim$ for a given finite-state process specified by $\mathcal{T}[\{\epsilon\},\{+\},Act,\mathcal{X}]$ syntactically and compositionally.

References

1. L. Aceto and M. Hennessy. Termination, deadlock, and divergence. *Journal of ACM, Vol. 39, No.1:147-187.* January, 1992.
2. S. Graf and J. Sifakis. A logic for the description of non-deterministic programs and their properties. *Information and Control,* 68:254-270. 1986.
3. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
4. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of ACM,* 32:137-161. Jan., 1985.
5. D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333-354. 1983.
6. K.G. Larsen and B. Thomsen. A modal process logic. In the proc. of LICS'88, pp.203-210. IEEE Computer Science Society, 1988.
7. R. Milner. A complete inference system for a class of regular behaviours. *Journal of Computer and System Sciences*, 28:439-466. 1984.
8. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.
9. C. Stirling. *Modal and Temporal Properties of Processes.* Springer-Verlag, 2001.
10. A. Tarski. A lattice-theoretical fixpoint theorem and its application. *Pacific J. Math.,* 5:285-309. 1955.
11. I. Walukiewicz. Completeness of Kozen's axiomatisation of the propositional Mu-calculus. *Information and Computation*, 157:142–182. 2000.