More on Weak Bisimilarity of Normed Basic Parallel Processes*

Haiyan Chen[1,2]

[1] State Key Laboratory Computer Science of Institute of Software, Chinese Academy of Sciences
[2] Graduate School of the Chinese Academy of Sciences
chy@ios.ac.cn

Abstract. Deciding strong and weak bisimilarity of BPP are challenging because of the infinite nature of the state space of such processes. Deciding weak bisimilarity is harder since the usual decomposition property which holds for strong bisimilarity fails. Hirshfeld proposed the notion of bisimulation tree to prove that weak bisimulation is decidable for totally normed BPA and BPP processes. In this paper, we present a tableau method to decide weak bisimilarity of totally normed BPP. Compared with Hirshfeld's bisimulation tree method, our method is more intuitive and more direct. Moreover from the decidability proof we can derive a complete axiomatisation for the weak bisimulation of totally normed BPP.

1 Introduction

A lot of attention has been devoted to the study of decidability and complexity of verification problems for infinite-state systems [1,15,16]. In [2], Baeten, Bergstra, and Klop proved the remarkable result that bisimulation equivalence was decidable for irredundant context-free grammars (without the empty product). Subsequently, many algorithms in this domain were proposed. In [7], Hans Hüttel and Colin Stirling proved the decidability of normed BPA by using a tableau method, which can also be used as a decision procedure. Decidability of strong bisimilarity for BPP processes has been established in [13]. Furthermore, [14] proved that deciding strong bisimilarity of BPP is PSPACE-complete.

For weak bisimilarity, much less is known. Semidecidability of weak bisimilarity for BPP has been shown in [5]. In [6] it is shown that weak bisimilarity is decidable for those BPA and BPP processes which are "totally normed". P.Jančar conjectured that the method in [14] might be used to show the decidability of weak bisimilarity for general BPP. However, the problem of decidability of weak bisimilarity for general BPP is open.

Our work is inspired by Hirshfeld's idea. In [6] Hirshfeld proposed the notion of bisimulation tree to prove the decidability of weak bisimulation of totally normed

* Supported by the National Natural Science Foundation of China under Grant Nos. 60673045, 60496321.

M. Agrawal et al. (Eds.): TAMC 2008, LNCS 4978, pp. 192–203, 2008.

BPP. Based on the idea, we show that weak bisimulation for totally normed BPP is decidable by a tableau method. In [13], S. Christensen, Y. Hirshfield and F. Moller proposed a tableau decision procedure for deciding strong bisimilarity of normed BPP. The key for tableau method to work is a nice decomposition property which holds for strong bisimulation, but fails for weak bisimulation. In our work, instead of using decomposition property, we apply Hirshfeld's idea to control the size of the tableaux to make the tableau method work correctly. This approach not only provides us a more direct decision method, but also has the advantage of providing a completeness proof of an equational theory for weak bisimulation of totally normed BPP processes, similar to the tableau method of [13] provides such a completeness proof for strong bisimulation of normed BPP processes. Moreover, the termination proof for tableau is greatly simplified.

The paper is organized as follows. Section 2 introduces the notion of BPP processes and weak bisimulation and describes weak bisimulation equivalence. Section 3 gives the tableau decision method and presents the soundness and completeness results. In Section 4 we prove the completeness of the equational theory. Finally, Section 5 sums up conclusions and gives suggestions for further work.

2 BPP Processes and Weak Bisimulation Equivalence

Assuming a set of variables $\mathcal{V}$, $\mathcal{V}$=$\{X, Y, Z, \cdots\}$ and a set of actions Act_τ, $Act_\tau = \{\tau, a, b, c, \cdots\}$ which contains a special element τ, we consider the set of BPP expressions $\mathcal{E}$ given by the following syntax; we shall use $E, F, \ldots$ as metavariables over $\mathcal{E}$.

$$\begin{array}{lll} E ::= & 0 & \text{(inaction)} \\ & \mid X & \text{(variables, } X \in \mathcal{V}) \\ & \mid E_1 + E_2 & \text{(summation)} \\ & \mid \mu E & (\mu \in Act_\tau) \\ & \mid E_1 | E_2 & \text{(merge)} \end{array}$$

A BPP process is defined by a finite family of recursive process equations

$$\Delta = \{X_i \stackrel{def}{=} E_i | 1 \leq i \leq n\}$$

where the $X_i \in \mathcal{V}$ are distinct variables and each E_i is BPP expressions, and free variables in each E_i range over set $\{X_1, \ldots, X_n\}$. In this paper, we concentrate on guarded BPP systems.

Definition 1. *A BPP expression E is guarded if each occurrence of variable is within the scope of an atomic action, and a BPP system is guarded if each E_i is guarded for $1 \leq i \leq n$.*

Definition 2. *The operational semantics of a guarded BPP system can be simply given by a labeled transition system $(\mathcal{S}, Act_\tau, \longrightarrow)$ where the transition relation $\longrightarrow$ is generated by the rules in Table 1.*

Table 1. Transition rules

act	$aE \xrightarrow{a} E$	rec	$\dfrac{E \xrightarrow{a} E'}{X \xrightarrow{a} E'} \quad (X = E \in \Delta)$
sum1	$\dfrac{E \xrightarrow{a} \alpha}{E + F \xrightarrow{a} \alpha}$	sum2	$\dfrac{F \xrightarrow{a} \beta}{E + F \xrightarrow{a} \beta}$
par1	$\dfrac{E \xrightarrow{a} \alpha}{E\|F \xrightarrow{a} \alpha\|F}$	par2	$\dfrac{F \xrightarrow{a} \beta}{E\|F \xrightarrow{a} E\|\beta}$

where the state space $\mathcal{S}$ consists of finite parallel of BPP processes, and the transition relation $\longrightarrow \subseteq \mathcal{S} \times Act_\tau \times \mathcal{S}$ is generated by the rules in Table 1, in which (as also later) we use Greek letters $\alpha, \beta, \cdots$ as meta variables ranging over elements of $\mathcal{S}$, Each such α denotes a BPP process by forming the product of the elements of α, i.e. by combining the elements of α in parallel using the merge operator. We write ϵ for empty sequence. We shall write X^n to represent the term $X|\cdots|X$ consisting of n copies of X combined in parallel. By length(α) we denote the cardinality of α.

It is shown in [3] that any guarded system can be effectively transformed into a 3-GNF normal form

$$\{X_i \stackrel{def}{=} \Sigma_{j=1}^{n_i} a_{ij}\alpha_{ij} | 1 \leq i \leq m\}$$

where for all i, j such that $1 \leq i \leq m$, $1 \leq j \leq n_i$, $length(\alpha_{ij}) < 3$. So we only considered BPP processes given in 3-GNF in this paper.

Moreover, we write $\alpha \stackrel{\epsilon}{\Longrightarrow} \beta$ for $\alpha(\stackrel{\tau}{\longrightarrow})^*\beta$, and write $\alpha \stackrel{a}{\Longrightarrow} \beta$ for $\alpha \stackrel{\epsilon}{\Longrightarrow} \stackrel{a}{\longrightarrow} \stackrel{\epsilon}{\Longrightarrow} \beta$. Let $\hat{}: Act_\tau \to \{Act_\tau - \tau\} \cup \epsilon$ be the function such that $\hat{a} = a$ when $a \neq \tau$ and $\hat{\tau} = \epsilon$, then the following general definition of weak bisimulation on $\mathcal{S}$ is standard.

Definition 3. *A binary relation $R \subseteq \mathcal{S} \times \mathcal{S}$ is a weak bisimulation if for all $(\alpha, \beta) \in R$ the following conditions hold:*

1. *whenever $\alpha \stackrel{a}{\longrightarrow} \alpha'$, then $\beta \stackrel{\hat{a}}{\Longrightarrow} \beta'$ for some β' with $(\alpha', \beta') \in R$;*
2. *whenever $\beta \stackrel{a}{\longrightarrow} \beta'$, then $\alpha \stackrel{\hat{a}}{\Longrightarrow} \alpha'$ for some α' with $(\alpha', \beta') \in R$.*

Two states α and β are said to be weak bisimulation equivalent, written $\alpha \approx \beta$, if there is a weak bisimulation R such that $(\alpha, \beta) \in R$.

It is standard to prove that $\approx$ is an equivalence relation between processes. Moreover it is a congruence with respect to composition on $\mathcal{S}$:

Proposition 1. *If $\alpha_1 \approx \beta_1$ and $\alpha_2 \approx \beta_2$ then $\alpha_1|\alpha_2 \approx \beta_1|\beta_2$.*

Proposition 2. *$\alpha|\beta \approx \beta|\alpha$.*

Proposition 3. $(\alpha|\beta)|\gamma \approx \alpha|(\beta|\gamma)$.

Definition 4. *A process α is said to be* normed *if there exists a finite sequence $\alpha \longrightarrow \ldots \longrightarrow \alpha_n \longrightarrow \epsilon$ transitions from α to ϵ, and un-normed otherwise. The weak norm of a normed α is the length of the shortest transition sequence of the form $\alpha \stackrel{a_1}{\Longrightarrow} \ldots \stackrel{a_n}{\Longrightarrow} \epsilon$, where each $a_i \neq \tau$ and $\stackrel{a_i}{\Longrightarrow}$ is counted as 1. We denote by $||\alpha||$ the weak norm of α. Also, for unnormed α, we follow the convention that $||\alpha|| = \infty$ and $\infty > n$ for any number n. A BPP system Δ is totally normed if for every variable X appears Δ, $0 < \|X\| < \infty$.*

With this definition, it is obvious that weak norm is additive: for normed $\alpha, \beta \in \mathcal{S}$, $||\alpha|\beta|| = ||\alpha|| + ||\beta||$. Moreover, the following proposition says that weak norm is respected by $\approx$.

Proposition 4. *If $\alpha \approx \beta$, then either both α, β are un-normed, or both are normed and $||\alpha|| = ||\beta||$.*

3 The Tableau Decision Method

From now on, we restrict our attention to the totally normed BPP processes in 3-GNF, i.e. processes of a parallel labeled rewrite system $\langle \mathcal{S}, Act_\tau, \longrightarrow \rangle$ where $\infty > ||X|| > 0$ for all $X \in \mathcal{V}$. And throughout the rest of the paper, we assume that all the processes considered are totally normed unless stated otherwise.

With the preparation of the previous section, in this section we can devise a tableau decision method. The rules of the tableau system are built around equations $\alpha = \beta$, where $\alpha, \beta \in \mathcal{S}$. Each rule of the tableau system has the form

$$\textsf{name} \quad \frac{\alpha = \beta}{\alpha_1 = \beta_1 \ldots \alpha_n = \beta_n} \quad \text{side condition.}$$

The premise of a rule represents the goal to be achieved while the consequents are the subgoals. There are three rules altogether. One for unfolding. Two rules for substituting the states. We now explain the three rules in turn.

Table 2. Tableau rules

$$\textsf{subl} \quad \frac{\alpha_1|\beta_1 = \alpha_2|\beta_2}{\alpha_1|\beta_1 = \alpha_1|\beta_2} \quad (\textit{if there is } \alpha_1 \prec \alpha_2 \textit{ and a dominated node labeled } \alpha_1 = \alpha_2 \textit{ or } \alpha_2 = \alpha_1\,)$$

$$\textsf{subr} \quad \frac{\alpha_1|\beta_1 = \alpha_2|\beta_2}{\alpha_2|\beta_1 = \alpha_2|\beta_2} \quad (\textit{if there is } \alpha_2 \prec \alpha_1 \textit{ and a dominated node labeled } \alpha_1 = \alpha_2 \textit{ or } \alpha_2 = \alpha_1\,)$$

$$\textsf{unfold} \quad \frac{\alpha = \beta}{\{\alpha' = \beta' \mid (\alpha', \beta') \in M\}} \quad M \textit{ is a match for } (\alpha, \beta)$$

3.1 Substituting the States

The next two rules can be used to substitute the expressions in the goal. The rules are based on the following observation.

Definition 5. (dominate and improve)[6]

1. *The pair* $(\alpha_1|\alpha_2, \beta_1|\beta_2)$ *dominates the pair* (α_1, β_1).
2. $X_1^{k_1}|\cdots|X_{i_0}^{k_{i_0}}|\cdots|X_n^{k_n}$ *improves* $X_1^{m_1}|\cdots|X_{i_0}^{m_{i_0}}|\cdots|X_n^{m_n}$ *iff there is some* i_0 *such that for* $i < i_0$ *the (total) number of occurrences of* X_i *is equal in both pairs, i.e.* $k_i = m_i$ *while the number of occurrences of* X_{i_0} *is smaller in* $(X_1^{k_1}|\cdots|X_{i_0}^{k_{i_0}}|\cdots|X_n^{k_n})$ *than in* $(X_1^{m_1}|\cdots|X_{i_0}^{m_{i_0}}|\cdots|X_n^{m_n})$ *i.e.* $k_{i_0} < m_{i_0}$.

Proposition 5. *Every sequence of pairs in which every pair improves the previous one is finite.*

Proposition 6. *Every sequence of pairs in which no pair dominates a previous one is finite.*

Definition 6. *By* $\prec$ *we denote the well-founded ordering on* $\mathcal{S}$ *given as follows:* $X_1^{k_1}|\cdots|X_n^{k_n} \prec X_1^{l_1}|\cdots|X_n^{l_n}$ *iff there exists* j *such that* $k_j < l_j$ *and for all* $i < j$ *we have* $k_i = l_i$.

It is easy to show that $\prec$ is well-founded. Moreover, We shall rely on the fact that $\prec$ is total in the sense that for any $\alpha, \beta \in \mathcal{S}$ such that $\alpha \neq \beta$ we have $\alpha \prec \beta$ or $\beta \prec \alpha$. Also we shall rely on the fact that $\alpha \prec \beta$ implies $\alpha|\alpha' \prec \beta|\alpha'$ as well as $\alpha \prec \beta|\alpha'$ for any $\alpha' \in \mathcal{S}$. All these properties are easily seen to hold for $\prec$.

When building tableaux basic nodes might dominate other basic nodes; we say a basic node $n : \alpha_1|\beta_1 = \alpha_2|\beta_2$ or $\alpha_2|\beta_2 = \alpha_1|\beta_1$ dominates any node $n' : \alpha_1 = \alpha_2$ or $n' : \alpha_2 = \alpha_1$ which appears above n in the tableau. There $n' : \alpha_1 = \alpha_2$ or $n' : \alpha_2 = \alpha_1$ is called the dominated node.

Definition 7. *We define a weight function* ω, *s.t. for* $\alpha = X_1^{k_1}|\cdots|X_n^{k_n}, \beta = Y_1^{m_1}|\cdots|Y_n^{m_n}$, $\omega(\alpha,\beta) = 1\times k_1 + 1\times k_2 + \cdots + 1\times k_n + 1\times m_1 + 1\times m_2 + \cdots + 1\times m_n$.

Proposition 7. *For every* α_1, β_1, *if* $\alpha \approx \beta$, *then* $\alpha|\alpha_1 \approx \beta|\beta_1$ *iff* $\alpha|\alpha_1 \approx \alpha|\beta_1$ *iff* $\beta|\alpha_1 \approx \beta|\beta_1$.

Proof. For the only if direction, suppose $\alpha|\alpha_1 \approx \beta|\beta_1$, since $\alpha \approx \beta$ and $\beta_1 \approx \beta_1$, then $\alpha|\beta_1 \approx \beta|\beta_1$ by Proposition 1, by $\alpha|\alpha_1 \approx \beta|\beta_1$, so $\alpha|\alpha_1 \approx \alpha|\beta_1$ since $\approx$ is an equivalence. For the if direction, suppose $\alpha|\alpha_1 \approx \alpha|\beta_1$, since $\alpha \approx \beta$ and $\beta_1 \approx \beta_1$, $\alpha|\beta_1 \approx \beta|\beta_1$ by Proposition 1, by $\alpha|\alpha_1 \approx \alpha|\beta_1$, so $\alpha|\alpha_1 \approx \beta|\beta_1$ since $\approx$ is an equivalence. For β it is similar to previous proof. □

Proposition 8. *One of the pairs* $(\alpha|\alpha_1, \alpha|\beta_1)$ *or* $(\beta|\alpha_1, \beta|\beta_1)$ *is an improvement of* $(\alpha|\alpha_1, \beta|\beta_1)$ *where* $\alpha \neq \beta$.

This proposition guarantees the soundness and backwards soundness of subl, subr rules.

In fact in section 2, from Proposition 5 we know that every sequence of pairs in which every pair improves the previous one is finite. So this means that there are only finitely many different ways to apply the rules.

3.2 Unfolding by Matching the Transitions

Definition 8. *Let $(\alpha, \beta) \in \mathcal{S} \times \mathcal{S}$. A binary relation $M \subseteq \mathcal{S} \times \mathcal{S}$ is a match for (α, β) if the following hold:*

1. *whenever $\alpha \xrightarrow{a} \alpha'$ then $\beta \overset{\hat{a}}{\Longrightarrow} \beta'$ for some $(\alpha', \beta') \in M$;*
2. *whenever $\beta \xrightarrow{a} \beta'$ then $\alpha \overset{\hat{a}}{\Longrightarrow} \alpha'$ for some $(\alpha', \beta') \in M$;*
3. *whenever $(\alpha', \beta') \in M$ then $||\alpha'|| = ||\beta'||$ and either $\alpha \xrightarrow{a} \alpha'$ or $\beta \xrightarrow{a} \beta'$ for some $a \in Act_\tau$.*

It is easy to see that for a given $(\alpha, \beta) \in \mathcal{S} \times \mathcal{S}$, there are finitely many possible $M \subseteq \mathcal{S} \times \mathcal{S}$ which satisfies 3. Above and moreover each of them must be finite. And for such M it is not difficult to see that it is decidable whether M is a match for (α, β).

The rule can be used to obtain subgoals by matching transitions, and it is based on the following observation.

Proposition 9. *Let $\alpha, \beta \in \mathcal{S}$. Then $\alpha \approx \beta$ if and only if there exists a match M for (α, β) such that $\alpha' \approx \beta'$ for all $(\alpha', \beta') \in M$.*

This proposition guarantees the soundness and backwards soundness of unfold rule.

As pointed out above there are finitely many matches for a given (α, β), so there are finitely many ways to apply this rule on (α, β).

3.3 Constructing Tableau

We determine whether $\alpha \approx \beta$ by constructing a tableau with root $\alpha = \beta$ using the three rules introduced above. A tableau is a finite tree with nodes labeled by equations of the form $\alpha = \beta$, where $\alpha, \beta \in \mathcal{S}$.

Moreover if $\alpha = \beta$ labels a non-leaf node, then the following are satisfied:

1. $||\alpha|| = ||\beta||$;
2. its sons are labeled by $\alpha_1 = \beta_1 \ldots \alpha_n = \beta_n$ obtained by applying rule subl, subr or unfold in Table 2 to $\alpha = \beta$, in that priority order;
3. no other non-leaf node is labelled by $\alpha = \beta$.

A tableau is a successful tableau if the labels of all its leaves have the forms:

1. $\alpha = \beta$ where there is a non-leaf node is also labeled $\alpha = \beta$;
2. $\alpha \equiv \beta$

3.4 Decidability, Soundness, and Completeness

Lemma 1. *Every tableau with root $\alpha = \beta$ is finite, Furthermore, there is only a finite number of tableaux with root $\alpha = \beta$.*

Theorem 1. *If $\alpha \approx \beta$ then there exists a successful tableau with root labeled $\alpha = \beta$.*

Proof. Suppose $\alpha \approx \beta$. If we can construct a tableau $T(\alpha = \beta)$ for $\alpha = \beta$ with the property that any node $n : \alpha = \beta$ of $T(\alpha = \beta)$ satisfies $\alpha \approx \beta$, then by Lemma 1 that construction must terminate and each terminal will be successful. Thus the tableau itself will be successful.

We can construct such a $T(\alpha = \beta)$ if we verify that each rule of the tableau system is forward sound in the sense that if the antecedent relates bisimilar processes then it is possible to find a set of consequents relating bisimilar processes. For the rule subl or subr we know from Proposition 7. For the rest of the tableau rules it is easily verified that they are forward sound in the above sense. □

Finally we must show soundness of the tableau system, namely that the existence of a successful tableau for $\alpha = \beta$ indicates that $\alpha \approx \beta$. This follows from the fact that the tableau system tries to construct a family of binary relations which are bisimilar.

Definition 9. *A sound tableau is a tableau such that if $\alpha = \beta$ is a label in it then $\alpha \approx \beta$.*

Theorem 2. *A successful tableau is a sound tableau.*

Proof. Let T be a successful tableau. We define $W = \{B \subseteq \mathcal{S} \times \mathcal{S}\}$ to be the smallest binary relations satisfies the following:

1. if $\alpha \equiv \beta$ labels a node in T then $(\alpha, \beta) \in W$;
2. if there is a node in T labeled $\alpha = \beta$ and on which rule unfold is applied then $(\alpha, \beta) \in W$;
3. if $(\alpha_1, \alpha_2) \in W$, $(\alpha_1|\beta_1, \alpha_1|\beta_2) \in W$ where $\alpha_1 \prec \alpha_2$ then $(\alpha_1|\beta_1, \alpha_2|\beta_2) \in W$;
4. if $(\alpha_1, \alpha_2) \in W$, $(\alpha_2|\beta_1, \alpha_2|\beta_2) \in W$ where $\alpha_2 \prec \alpha_1$then $(\alpha_1|\beta_1, \alpha_2|\beta_2) \in W$.

We will prove the following properties about W:

A. If $\alpha = \beta$ labels a node in T then $(\alpha, \beta) \in W$.
B. If $(\alpha, \beta) \in W$, then the following hold:
   (a) if $\alpha \xrightarrow{a} \alpha'$ then $\beta \xRightarrow{\hat{a}} \beta'$ for some β' such that $(\alpha', \beta') \in W$;
   (b) if $\beta \xrightarrow{a} \beta'$ then $\alpha \xRightarrow{\hat{a}} \alpha'$ for some α' such that $(\alpha', \beta') \in W$.

Clearly property B. implies that

$$B = \{(\alpha, \beta) \mid (\alpha, \beta) \in W\}$$

is a weak bisimulation. Then together with property A. it implies that T is a sound tableau.

We prove A. by induction on weight $\omega' = \omega(\alpha, \beta)$. If $\alpha = \beta$ is a label of an non-leaf node, there are three cases according to which rule is applied on this node. If unfold is applied, then by rule 2. of the construction of W clearly $(\alpha, \beta) \in W$. If subl is applied, in this case $\alpha = \beta$ is of the form $\alpha_1|\beta_1 = \alpha_2|\beta_2$, and the node has sons labeled by $\alpha_1|\beta_1 = \alpha_1|\beta_2$. Clearly $\omega(\alpha_1|\beta_1, \alpha_1|\beta_2) < \omega(\alpha_1|\beta_1, \alpha_2|\beta_2)$, then by the induction hypothesis $(\alpha_1|\beta_1, \alpha_1|\beta_2) \in W$. Then by rule 3. in the construction of W, $(\alpha_1|\beta_1, \alpha_2|\beta_2) \in W$. If subr is applied, it is similar to subl proof. If $\alpha = \beta$ is a label of a leaf node, then since T is a successful tableau either there is a non-leaf node also labeled by $\alpha = \beta$ and in this case we have proved that $(\alpha, \beta) \in W$, or $\alpha \equiv \beta$ must hold and in this case by rule 1. in the construction of W we also have $(\alpha, \beta) \in W$.

We prove B. by induction on the four rules define W. Suppose $(\alpha, \beta) \in W$, there are the following cases.

Case of rule 1. i.e. $\alpha \equiv \beta$. It is obvious B. holds.

Case of rule 2. i.e. there exists M which is a match for (α, β) such that $\alpha' = \beta'$ is a label of T for all $(\alpha', \beta') \in M$. Then by A. it holds that $(\alpha', \beta') \in W$ for all $(\alpha', \beta') \in M$, then by definition of a match, clearly B. holds.

Case of rule 3. i.e. there exist $(\alpha_1, \alpha_2) \in W$, $(\alpha_1|\beta_1, \alpha_1|\beta_2) \in W$ where $\alpha_1 \prec \alpha_2$ and $\alpha = \alpha_1|\beta_1$. If $\alpha_1|\beta_1 \xrightarrow{a} \alpha''$, we have to match this by looking for a β'' such that $\alpha_2|\beta_2 \stackrel{\hat{a}}{\Longrightarrow} \beta''$ and $(\alpha'', \beta'') \in W$. By transition rule for α'' has two cases: the first case $\alpha_1 \xrightarrow{a} \alpha_1'$ then $\alpha'' = \alpha_1'|\beta_1$. Now $(\alpha_1, \alpha_2) \in W$, by the induction hypothesis there exists $\alpha_2' \in \alpha$ such that $\alpha_2 \stackrel{\hat{a}}{\Longrightarrow} \alpha_2'$ and $(\alpha_1', \alpha_2') \in W$. since $(\alpha_1|\beta_1, \alpha_1|\beta_2) \in W$, by the induction hypothesis there exists $\alpha_1'|\beta_2 \in \alpha$ such that $\alpha_1|\beta_2 \stackrel{\hat{a}}{\Longrightarrow} \alpha_1'|\beta_2$ and $(\alpha_1'|\beta_1, \alpha_1'|\beta_2) \in W$. By rule 3 we have $(\alpha_1'|\beta_1, \alpha_2'|\beta_2) \in W$. The another direction can be proved in a similar way; the second case $\beta_1 \xrightarrow{a} \beta_1'$ then $\alpha'' = \alpha_1|\beta_1'$. since $(\alpha_1|\beta_1, \alpha_1|\beta_2) \in W$, by the induction hypothesis there exists $\alpha_1|\beta_1' \in \alpha$ such that $\alpha_1|\beta_2 \stackrel{\hat{a}}{\Longrightarrow} \alpha_1|\beta_2'$ and $(\alpha_1|\beta_1', \alpha_1|\beta_2') \in W$. Now $(\alpha_1, \alpha_2) \in W$, by rule 3 we have $(\alpha_1|\beta_1', \alpha_2|\beta_2') \in W$. The another direction can be proved in a similar way.

Case of rule 4. it is similar rule 3. proof. □

Theorem 3. *Let $\alpha, \beta \in \mathcal{S}$ be totally normed. Then $\alpha \approx \beta$ if and only if there exists a successful tableau with root $\alpha = \beta$.*

4 The Equational Theory

We will develop the equational theory proposed by Sϕren Chirstensen, Yoram Hirshfeld, Faron Moller in [13] for strong bisimulation on normed BPP processes given in 3-GNF. We now describe a sound and complete axiomatisation for totally normed BPP processes. We pay attention to BPP processes in 3-GNF. The axiomatisation shall be parameterised by Δ and consists of axioms and inference rules that enable one to derive the root of successful tableaux.

The axiomatisation is built around sequences of the form $\Gamma \vdash_{\Delta} E = F$ where Γ is a finite set of assumptions of the form $\alpha = \beta$ and E, F are BPP expressions. Let Δ be a finite family of BPP processes in 3-GNF. A sequent is interpreted as follows:

Definition 10. *We write $\Gamma \models_{\Delta} E = F$ when it is the case that if the relation $\{(\alpha, \beta) | \alpha = \beta \in \Gamma\} \cup \{(X_i, E_i) | X_i \stackrel{def}{=} E_i \in \Delta\}$ is part of a bisimulation then $E \approx F$.*

Thus, the special case $\emptyset \models_\Delta E = F$ states that $E \approx F$ (relative to the system of process equations Δ).

For the presentation of rule unfold we introduce notation $unf(\alpha)$ to mean the unfolding of α given as follows (assuming that $\alpha \equiv Y_1|Y_2|\cdots|Y_m$):

$$unf(\alpha) = \sum_{i=1}^{m}\{a_i\gamma_i : a_i\alpha_i \in Y_i\},$$

where $\gamma_i = \alpha_i|(\prod_{j=1,j\neq i}^{m} Y_j)$ and the notation $a\alpha \in Y$ means that $a\alpha$ is a summand of the defining equation for Y.

The proof system is presented in Table 3. Equivalence and congruence rules are R1-6. In [13] the rule R5 of the axiomatisation for normed BPP processes can not directly apply in our rules, since we know that weak bisimulation is not preserved by summation, i.e. if $E_1 \approx E_2$ and $F_1 \approx F_2$, but we can't get $E_1 + F_1 \approx E_2 + F_2$. So we increase two rules R5-6 to achieve summation. The rules R7-15 correspond to the BPP laws; notably we have associativity and commutativity for merge. Finally, we have two rules characteristic for this axiomatisation; R16 is an assumption introduction rule underpinning the role of the assumption list Γ and R17 is an assumption elimination rule and also a version of fixed point induction. The special form of R17 has been dictated by the rule unfold of the tableau system presented in Table 2.

Definition 11. *A proof of $\Gamma \vdash_\Delta E = F$ is a proof tree with root labeled $\Gamma \vdash_\Delta E = F$, instances of the axioms R1 and R7-R16 as leaves and where the father of a set of nodes is determined by an application of one of the inference rules R2-R6 or R17.*

Definition 12. *The relations $\approx_o$ for ordinals o are defined inductively as follows, where we assume that l is a limit ordinal*
$E \approx_0 F$ for all E, F
$E \approx_{o+1} F$ iff for $a \in (Act_\tau \cup \{\epsilon\})$

$E \stackrel{a}{\Longrightarrow} E'$, *then* $\exists F'.F \stackrel{\hat{a}}{\Longrightarrow} F'$ *and* $E' \approx_o F'$

$F \stackrel{a}{\Longrightarrow} F'$, *then* $\exists E'.E \stackrel{\hat{a}}{\Longrightarrow} E'$ *and* $E' \approx_o F'$.
$E \approx_l F$ iff $\forall o < l. E \approx_o F$

So we can get a fact that is $\approx = \bigcap_{n=0}^{\infty} \approx_n$.

Theorem 4. (Soundness) *If $\Gamma \vdash_\Delta E = F$ then we have $\Gamma \models_\Delta E = F$. In particular if $\vdash_\Delta E = F$ then $E \approx F$.*

The similar proof for soundness can be found in [13].

Lemma 2. *If $\Gamma \vdash_\Delta E = F$ then $\Gamma, \Gamma' \vdash_\Delta E = F$ for any Γ'.*

The completeness proof rests on a number of lemmas and definitions which tell us how to determine our sets of hypotheses throughout a proof of $E \approx F$ from a successful tableau for $E \approx F$. We prove completeness from [13] idea.

Table 3. The axiomatisation

Equivalence		Congruence	
R1	$\Gamma \vdash_\Delta E = E$	R4	$\dfrac{\Gamma \vdash_\Delta E_1 = F_1 \;\; \Gamma \vdash_\Delta E_2 = F_2}{\Gamma \vdash_\Delta E_1 \| E_2 = F_1 \| F_2}$
R2	$\dfrac{\Gamma \vdash_\Delta E = F}{\Gamma \vdash_\Delta F = E}$	R5	$\dfrac{\Gamma \vdash_\Delta E = F}{\Gamma \vdash_\Delta E = F + \tau E}$
R3	$\dfrac{\Gamma \vdash_\Delta E = F \;\; \Gamma \vdash_\Delta F = G}{\Gamma \vdash_\Delta E = G}$	R6	$\dfrac{\Gamma \vdash_\Delta E = F}{\Gamma \vdash_\Delta aE + R = aF + R}$
Axioms			
R7	$\Gamma \vdash_\Delta E + (F + G) = (E + F) + G$	R12	$\Gamma \vdash_\Delta E \| F = F \| E$
R8	$\Gamma \vdash_\Delta E + F = F + E$	R13	$\Gamma \vdash_\Delta E \| 0 = E$
R9	$\Gamma \vdash_\Delta E + E = E$	R14	$\Gamma \vdash_\Delta \tau E = E$
R10	$\Gamma \vdash_\Delta E + 0 = E$	R15	$\Gamma \vdash_\Delta a(E + \tau F) + aF = a(E + \tau F)$
R11	$\Gamma \vdash_\Delta E \| (F \| G) = (E \| F) \| G$		
Recursion			
R16	$\Gamma, \alpha = \beta \vdash_\Delta \alpha = \beta$	R17	$\dfrac{\Gamma, \alpha = \beta \vdash_\Delta unf(\alpha) = unf(\beta)}{\Gamma \vdash_\Delta \alpha = \beta}$

Definition 13. *For any node n of a tableau, $Rn(n)$ denotes the set of labels of the nodes above n to which the rule* unfold *is applied. In particular, $Rn(r)$=$\emptyset$ where r is the root of the tableau.*

Theorem 5. (Completeness)*If $\alpha \approx \beta$ then $\Gamma \vdash_\Delta \alpha = \beta$*

Proof. If $\alpha \approx \beta$, then there exists a finite successful tableau with root labeled $\alpha = \beta$. Let $T(\alpha = \beta)$ be such a tableau. We shall prove that for any node $n : E = F$ of $T(\alpha = \beta)$ we have $Rn(n) \vdash_\Delta E = F$. In particular, for the root $r : \alpha = \beta$, this reduces to $\vdash_\Delta \alpha = \beta$, so we shall have our result.

We prove $Rn(n) \vdash_\Delta E = F$ by induction on the depth of the subtableau rooted at n. As the tableau is built modulo associativity and commutativity of merge and by removing 0 components sitting in parallel or in sum we shall assume that the axioms R12-R14 are used whenever required to accomplish the proof.

Firstly, if $n : E = F$ is a terminal node then either E and F are identical terms α, so $Rn(n) \vdash E = F$ follows from R1.

Hence assume that $n : E = F$ is a internal nodes. We proceed to apply to n according to the tableau rule.

(i) Suppose unfold is applied. Then n is the label $E = F$ and the son n' of n is labeled $E_i = F_i (i \in \{1 \cdots n\}$, $E_i = F_i$ is match of $E = F)$, by induction hypothesis $Rn(E_i = F_i) \vdash E_i = F_i$, $Rn(E_i = F_i) - \{E = F\}, E = F \vdash E_i = F_i$, $Rn(E_i = F_i) - \{E = F\} \vdash E = F$ by R17, we know $Rn(E = F) = Rn(E_i = F_i) - \{E = F\}$, so $Rn(E = F) \vdash E = F$.

(ii) Suppose sub is applied wlog that is subl. Then the label $E = F$ is of the form $E_1 | F_1 = E_2 | F_2$ with the corresponding node n'' labeled $E_1 = E_2$ and the son n' of n is labeled $E_1|F_1 = E_1|F_2$, by induction hypothesis $Rn(E_1|F_1 = E_1|F_2) \vdash E_1|F_1 = E_1|F_2$ since $Rn(E_1|F_1 = E_1|F_2) = Rn(E_1|F_1 = E_2|F_2)$, and $E_1 = E_2 \in Rn(E_1|F_1 = E_2|F_2)$, so $Rn(E_1|F_1 = E_2|F_2) \vdash E_1 = E_2$ by R16. Hence from R1, R4, R3 we have $Rn(E_1|F_1 = E_2|F_2) \vdash E_1|F_1 = E_2|F_2$, last $Rn(E = F) \vdash E = F$ is required.

This completes the proof. □

5 Conclusions and Directions for Further Work

In this paper we proposed a tableau method to decide whether a pair of totally normed BPP processes is a weak bisimilar relation. The whole procedure is direct and easy to understand, while the termination proof is also very simple. This tableau method also helps us to show the completeness of Sϕren Chirstensen, Yoram Hirshfeld, Faron Moller's equational theory on totally normed BPP systems. Recent results by Richard Mayr show that weak bisimulation of Basic Parallel Processes is $\prod_2^P$-hard[18].

The study of bisimulation decision problems in the fields of BPA and BPP processes has been already rather sophisticated. All the results were recorded and updated by J.Srba[17], as well as open problems in this field. About algorithms, the things left should be concerned with lowing complexity and improving the efficiency. As the equational theory depends on assumptions, it is somewhat different from Milner's equational theory for regular processes[9]. One direction of interest is the construction of equational theory of $\approx$ since many decision results for weak bisimulation are already given.

References

1. Moller, F.: Infinite results. In: Sassone, V., Montanari, U. (eds.) CONCUR 1996. LNCS, vol. 1119, pp. 195–216. Springer, Heidelberg (1996)
2. Baeten, J.C.M., Bergstra, J.A., Klop, J.W.: Decidability of bisimulation equivalence for processes generating context-free languages. Journal of the Association for Computing Machinery 93(40), 653–682 (1993)
3. Christensen, S.: Forthcoming Ph.D. thesis. University of Edinburgh
4. Hirshfeld, Y., Jerrum, M., Moller, F.: Bisimulation equivalence is decidable for normed process algebra. In: Wiedermann, J., Van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 412–421. Springer, Heidelberg (1999)
5. Esparza, J.: Petri nets, commutative context-free grammars, and basic parallel processes. In: Reichel, H. (ed.) FCT 1995. LNCS, vol. 965, pp. 221–232. Springer, Heidelberg (1995)
6. Hirshfeld, Y.: Bisimulation trees and the decidability of weak bisimulations. In: INFINITY 1996. Proceedings of the 1st International Workshop on Verification of Infinite State Systems, Germany, vol. 5 (1996)
7. Hüttel, H., Stirling, C.: Actions speak louder than words. Proving bisimilarity for context-free processes. In: LICS 1991. Proceedings of 6th Annual symposium on Logic in Computer Science, Amsterdam, pp. 376–386 (1991)
8. R., Milner: A Calculus of Communicating Systems. LNCS, vol. 92. Springer, Heidelberg (1980)

9. Milner, R.: Communication and Concurrency. Prentice-Hall International, Englewood Cliffs (1989)
10. Bergstra, J.A., Klop, J.W.: Process theory based on bisimulation semantics. In: Prodeedings of REX Workshop, pp. 50–122. The Netherlands, Amsterdam (1988)
11. Caucal, D.: Graphes canoniques de graphes algébriques. Informatique théorique et Applications(RAIRO) 90-24(4), 339–352 (1990)
12. Dixon, L.E.: Finiteness of the odd perfct and primitive abundant numbers with distinct factors. American Journal of Mathematics 35, 413–422 (1913)
13. Christensen, S., Hirshfield, Y., Moller, F.: Decomposability, Decidability and Axiomatisability for Bisimulation Equivalence on Basic Parallel Processes. In: LICS 1993. Proceedings of the Eighth Annual Symposium on Logic in Computer Science, Montreal, Canada, pp. 386–396 (1993)
14. Jančar, P.: Strong bisimilarity on basic parallel processes is PSPACE-complete. In: LICS 2003. Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science, Ottawa, Canada, pp. 218-227 (2003)
15. Esparza, J.: Decidability of model checking for infinite-state concurrent systems. Acta Informatica 97-34, 85–107 (1997)
16. Burkart, O., Esparza, J.: More infinite results. Electronic Notes in Theoretical computer Science 5 (1997)
17. Srba, J.: Roadmap of Infinite Results. Bulletin of the European Association for Theoretical Computer Science 78, 163–175 (2002), columns:Concurrency
18. Mayr, R.: On the Complexity of Bisimulation Problems for Basic Parallel Processes. In: Welzl, E., Montanari, U., Rolim, J.D.P. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 329–341. Springer, Heidelberg (2000)
19. Mayr, R.: Weak Bisimulation and Model Checking for Basic Parallel Processes. In: Chandru, V., Vinay, V. (eds.) FSTTCS 1996. LNCS, vol. 1180, pp. 88–99. Springer, Heidelberg (1996)