Actions Speak Louder than Words: Proving Bisimilarity for Context-Free Processes

Hans Hüttel – hans@iesd.auc.dk*
Department of Mathematics and Computer Science
Aalborg University Centre
Fredrik Bajersvej 7E
9220 Aalborg Ø
DENMARK

Colin Stirling – cps@dcs.ed.ac.uk
Laboratory for Foundations of Computer Science
James Clerk Maxwell Building
University of Edinburgh
Edinburgh EH9 3JZ
SCOTLAND

June 1993

Abstract

Baeten, Bergstra, and Klop (and later Caucal) have proved the remarkable result that bisimulation equivalence is decidable for irredundant context-free grammars. In this paper we provide a much simpler and much more direct proof of this result using a tableau decision method involving goal-directed rules. The decision procedure also provides the essential part of the bisimulation relation between two processes which underlies their equivalence. We also show how to obtain a sound and complete sequent-based equational theory for such processes from the tableau system and how one can extract what Caucal calls a fundamental relation from a successful tableau.

*An earlier, shorter version of this paper appeared at the 6th Annual Symposium on Logic in Computer Science (LICS 91), Amsterdam 1991.

1 Introduction

In [1] (and [2]) Baeten, Bergstra, and Klop prove the remarkable result that bisimulation equivalence is *decidable* for irredundant context-free grammars (without the empty production). Within process calculus theory these grammars correspond to *normed* processes defined by a finite family of guarded recursion equations in the signature of BPA (Basic Process Algebra) [4]. These processes can have infinitely many states (even after quotienting by bisimulation equivalence). Consequently the process calculus approach (as exemplified in [24]) encompasses a much richer class of infinite-state systems that are open to automatic techniques normally associated with finite state systems than all those approaches based on trace, or language, equivalence. Recently, Huynh and Tian [20] have shown that failures and readiness equivalences are *undecidable* for this class of processes and Groote and the first author [15] have proved that in fact *all* known equivalences other than bisimulation are undecidable here, thus suggesting a new criterion for distinguishing between the computational qualities of behavioural equivalences.

However, the proof of decidability in [1, 2] is not easy as it relies on isolating a possibly complex periodicity from the transition graphs of these processes. An alternative, more elegant, proof utilizing rewrite techniques is presented by Caucal [6]; a simplified version of this proof is due to Groote [14]. The idea is to show that the maximal bisimulation on a transition graph is given as the least congruence of a canonical and strongly normalizing Thue system and that there are only finitely many candidates for such a system. However, the decision procedure consists of a linear search for the desired Thue system. Neither of the proofs reflects how one intuitively would show that two processes are bisimilar.

In this paper we first provide a simpler and much more direct proof of the decidability result using a *tableau* decision method involving goal-directed rules; our method is closely related to the branching algorithms introduced by Korenjak and Hopcroft for the study of equivalence problems in language theory and is also related to the tableau methods used by the second author in the different context of local model checking finite and infinite state transition systems [26, 5]. The decision procedure yields an upper bound on the depth of a tableau. Moreover, it provides the essential part of the bisimulation relation between two processes which underlies their equivalence, a self-bisimulation in the sense of [6].

An important by-product of the tableau system is a sound and complete sequent-based equational theory for normed BPA processes; the theory emanates from 'running the tableau method backwards'. We see this as an initial step in extending Milner's axiomatization of regular processes [23] to this class of 'context-free' processes. At the same time it offers an alternative method for designing equational theories which does not depend on appealing to normal forms.

Preliminaries are dealt with in Section 2. In Section 3 we present the tableau decision method and give an upper bound on the depth of a tableau. Section

4 gives a sound and complete equational theory for these normed context-free processes that follows from the tableau system and finally, in Section 5 we show how to extract what Caucal calls a fundamental relation from a successful tableau.

2 Preliminaries

2.1 Normed recursive BPA processes

We consider the class of guarded recursive normed BPA (Basic Process Algebra) processes (see e.g. [2, 4]). BPA process expressions are given by the abstract syntax

$$E ::= a \mid X \mid E_1 + E_2 \mid E_1 E_2$$

Here a ranges over a set of atomic actions, and X over a set of variables. The operator $+$ is nondeterministic choice while $E_1 E_2$ is the sequential composition of E_1 and E_2. A process is defined by a finite system of recursive process equations Δ

$$\Delta = \{X_i \stackrel{\mathrm{def}}{=} E_i \mid 1 \leq i \leq k\}$$

where the X_i are distinct, and the E_i are guarded BPA expressions with free variables in $Var = \{X_1, \ldots, X_m\}$. A process expression is *guarded* if every variable occurrence is within the scope of an atomic action. In the sequel we let $X, Y, \ldots$ range over variables in Var and $\alpha, \beta, \ldots$ over finite length sequences of variables.

Given a system of process equations Δ we define its transition behaviour by the following rules (ϵ denotes the empty process with the expected convention that ϵF is just F):

$$\frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'} \qquad \frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'}$$

$$\frac{E \xrightarrow{a} E'}{EF \xrightarrow{a} E'F} \qquad a \xrightarrow{a} \epsilon \quad a \in Act$$

$$\frac{E \xrightarrow{a} E'}{X \xrightarrow{a} E'} \qquad X \stackrel{\mathrm{def}}{=} E \in \Delta$$

A simple extension is the transitive closure of the relations $\{\xrightarrow{a} \mid a \in Act\}$: for $w \in Act^+$ we write $p \xrightarrow{aw} q$ if $p \xrightarrow{a} p'$ and $p' \xrightarrow{w} q$ for some p'.

The important extra restriction on a system of process equations Δ is *normedness.*

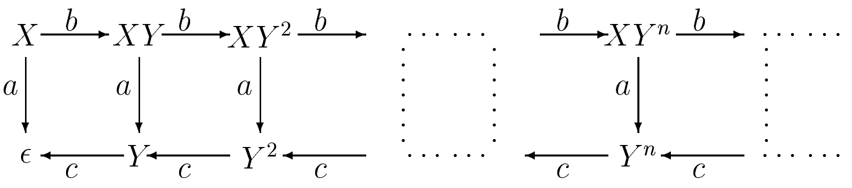

Figure 1: Transition graph for $X \stackrel{\text{def}}{=} a + bXY; \quad Y \stackrel{\text{def}}{=} c$ (Example 1)

Definition 1 [2] *The* norm *of a BPA expression E is*

$$|E| = \min\{length(w) \mid E \xrightarrow{w} \epsilon\}$$

E is said to be normed *if $E < \infty$. If for all variables X in Δ we have $|X| < \infty$ we say that Δ is normed.*

The norm is easily computed: we have $|a| = 1$, $|E + F| = \min(|E|, |F|)$, $|EF| = |E| + |F|$ and when $X \stackrel{\text{def}}{=} E$, $|X| = |E|$.

Example 1 Consider the pair $X \stackrel{\text{def}}{=} a + bXY; \quad Y \stackrel{\text{def}}{=} c$. Here $|X| = |Y| = 1$. By the transition rules above X generates the transition system in Figure 1. □

Because of the normedness restriction, this class of processes does not include the regular processes (such as $X \stackrel{\text{def}}{=} aX$). Nevertheless, it is a very rich class of processes and as illustrated in Figure 1, it contains processes that can have infinitely many states even after quotienting by bisimulation equivalence (or, in fact, any other equivalence).

The language $L(X_i)$ accepted by a variable X_i is the set $\{w \mid X_i \xrightarrow{w} \epsilon\}$. For instance, in Example 1 above $L(Y) = \{c\}$ whereas $L(X) = \{b^n a c^n \mid n \geq 0\}$.

When the variables of two families in 3-GNF Δ and Δ' are disjoint, the system of process equations $\Delta \cup \Delta'$ also defines a set of normed processes in normal form. Consequently in general the question of whether $L(X) = L(Y)$ when X, Y are variables in a system of BPA process equations Δ is *undecidable* since this is just a reformulation of the usual equivalence problem for context-free grammars [18].

2.2 Bisimulation equivalence

Within process calculus theory a variety of equivalences have been proposed in order to capture when two processes may be said to exhibit the same behaviour. The most notable is bisimulation equivalence [25], as utilized in [24] for example.

Definition 2 *A relation R between processes is a* bisimulation *if whenever pRq then for each $a \in Act$*

1. $p \xrightarrow{a} p' \Rightarrow \exists q' : q \xrightarrow{a} q' \wedge p'Rq'$
2. $q \xrightarrow{a} q' \Rightarrow \exists p' : p \xrightarrow{a} p' \wedge p'Rq'$

Two processes p and q are said to be bisimulation equivalent *or* bisimilar, *written $p \sim q$, if there is a bisimulation R such that pRq.*

Not only is $\sim$ an equivalence relation, but it is also a congruence relation w.r.t. the process constructs of BPA [4].

Example 2 For an example of bisimilar BPA process expressions take the system of process equations $\{X \stackrel{\text{def}}{=} aYX + b, Y \stackrel{\text{def}}{=} bX, A \stackrel{\text{def}}{=} aC + b, C \stackrel{\text{def}}{=} bAA\}$. We have that $X \sim A$; the reader may want to verify that the relation $\{(X^n, A^n) \mid n \geq 0\} \cup \{(YX^{n+1}, CA^n) | n \geq 0\}$ is a bisimulation (where X^n here denotes n successive Xs, $X \in V$). □

The following proposition, originally due to Caucal [7] is essential, providing us with a way of removing suffixes of bisimilar BPA expressions.

Proposition 1 *If G is a normed BPA process expression and E, F are arbitrary BPA process expressions and $EG \sim FG$ then $E \sim F$.*

Proof: We show that the relation

$$R = \{(E, F) \mid EG \sim FG \text{ for some } G\} \cup \{(\epsilon, \epsilon)\}$$

is a bisimulation. Suppose not. Then wlog $(E, F) \in R$ and for some E' which may be ϵ, $E \xrightarrow{a} E'$ and whenever $F \xrightarrow{a} F'$ we have that $(E', F') \notin R$. We know that there is a normed G such that $EG \sim FG$. We consider two cases. First $E' = \epsilon$. Then $EG \xrightarrow{a} G$. By assumption, if $F \xrightarrow{a} F'$ then $F' \neq \epsilon$. Hence $FG \xrightarrow{a} F'G$ for some $F' \neq \epsilon$ such that $G \sim F'G$. But this is impossible, as $|G| \neq |F'G|$ and therefore $G \not\sim F'G$. Otherwise $E' \neq \epsilon$ and so $EG \xrightarrow{a} E'G$. Consequently $FG \xrightarrow{a} F'G$ for some F' such that $E'G \sim F'G$. If $F' \neq \epsilon$ then after all $(E', F') \in R$. If $F' = \epsilon$ then $E'G \sim G$ which again is impossible. □

Note that the proof relies on the assumed normedness; a simple counterexample for the unnormed case is that the system of process equations $\{X \stackrel{\text{def}}{=} aX, Y \stackrel{\text{def}}{=} a\}$. Here $YYX \sim YX$ but clearly $YY \not\sim Y$.

2.3 Normed recursive BPA processes in Greibach Normal Form

Any system of process equations Δ has a unique solution up to bisimulation equivalence [3]. Moreover, in [2] it is shown that any guarded system of equations can be effectively presented in a normal form (2.3), using a system of process equations Δ' of the form

$$\{X_i \stackrel{\mathrm{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij} \mid 1 \leq i \leq m\}$$

where each variable sequence α_{ij} has length of at most two. This normal form preserves bisimulation equivalence. Moreover, when Δ only contains normed processes, so does Δ'.

This normal form is called *3-GNF* (3-Greibach Normal Form) by analogy with context-free grammars (without the empty production). There is an obvious correspondence between variables and non-terminals and actions and terminals, and each equation $X_i \stackrel{\mathrm{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij}$ can be viewed as the family of productions $\{X_i \rightarrow a_{ij}\alpha_{ij} \mid 1 \leq j \leq n_i\}$. Normedness corresponds to irredundancy in the grammar. Conversely, any context-free language (without the empty string) is generated by such a grammar (where some variable is designated as the start symbol).

An important advantage of using GNF is that the states in the transition graph for a process given in this way are elements of Var^*. Moreover, the restriction to variable sequences of length at most 2 guarantees limited growth of these sequences under single transitions. When applying a defining equation to the leftmost variable in a string α the length of the derivative increases by at most 1:

Proposition 2 *Suppose Δ is in 3-GNF. Then, for any $\alpha \in Var^*$, whenever $\alpha \xrightarrow{a} \alpha'$ we have $length(\alpha') \leq length(\alpha) + 1$.*

PROOF: Suppose $\alpha = X_i\alpha'''$. Then $\alpha \xrightarrow{a} \alpha'$ must be due to $X_i \xrightarrow{a} \alpha''$. This in turn is due to the defining equation $X_i \stackrel{\mathrm{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij}$ having a summand $a\alpha''$ with $length(\alpha'') \leq 2$. Since $\alpha' = \alpha''\alpha'''$, the result follows. □

Finally, there is a simple relationship between lengths and norms for variable sequences.

Proposition 3 *For $\alpha \in Var^*$ $length(\alpha) \leq |\alpha|$ and $|\alpha| \leq m_\Delta length(\alpha)$.*

2.4 Self-bisimulations

For finite-state processes a naive decision procedure for the bisimulation problem $p \sim q$ consists in enumerating all binary relations over the state space and determining if there is a relation among them which is a bisimulation containing (p,q). But since in general bisimulations over normed BPA processes may be infinite - for instance, the least non-empty bisimulation for the transition graph in Example 1 is the identity - a decision procedure for the bisimulation problem for normed BPA cannot rely on this. However, whenever $\alpha \sim \beta$, our tableau system will construct a *self-bisimulation*, a finite relation $R \subseteq Var^* \times Var^*$ whose closure under congruence w.r.t. sequential composition is a bisimulation containing (α, β). The notion of self-bisimulation was introduced by Didier Caucal in [7] (originally published as [6]). Here the notion of a least congruence is essential.

Definition 3 *For any binary relation R on Var^*, $\underset{R}{\rightarrow}$ is the least precongruence w.r.t. sequential composition that contains R, $\underset{R}{\longleftrightarrow}$ the symmetric closure of $\underset{R}{\rightarrow}$ and $\underset{R}{\longleftrightarrow}^*$ the reflexive and transitive closure of $\underset{R}{\longleftrightarrow}$ and thus the least congruence w.r.t. sequential composition containing R.*

A self-bisimulation is then simply a bisimulation up to congruence w.r.t. sequential composition.

Definition 4 *A relation $R \subseteq Var^* \times Var^*$ is called* a self-bisimulation *iff $\alpha R \beta$ implies that*

1. *$\alpha \xrightarrow{a} \alpha'$ implies $\beta \xrightarrow{a} \beta'$ for some β' with $\alpha' \underset{R}{\longleftrightarrow}^* \beta'$*

2. *$\beta \xrightarrow{a} \beta'$ implies $\alpha \xrightarrow{a} \alpha'$ for some β' with $\alpha' \underset{R}{\longleftrightarrow}^* \beta'$*

The following lemma, due to Caucal, shows that a self-bisimulation is a finite witness for bisimilarity.

Lemma 1 [7] *If R is self-bisimilar then $\underset{R}{\longleftrightarrow}^* \subseteq \sim$.*

Corollary 1 *$\alpha \sim \beta$ iff there is a self-bisimulation R such that $\alpha R \beta$.*

Proof: Clearly, since $\sim$ is a congruence it is also a self-bisimulation. Conversely, by the above lemma, if R is a self-bisimulation then $\underset{R}{\longleftrightarrow}^*$ is a bisimulation. $\square$

3 The tableau decision method

The bisimulation checker for normed BPA we now present is a *tableau system*, a goal-directed proof system. The proof technique is similar to the algorithm used in [22] to show that language equivalence is decidable for simple grammars.

We assume a fixed system of normed BPA process equations in 3-GNF, $\Delta = \{X_i \stackrel{\text{def}}{=} \sum_{j=1}^{n_i} a_{ij}\alpha_{ij} \mid 1 \leq i \leq m\}$. We determine whether $X\alpha \sim Y\beta$ (assuming of course that all occurring variables are defined in Δ) by constructing a tableau using the proof rules presented in Table 1. A *tableau* for $X\alpha = Y\beta$ is a maximal finite proof tree whose root is labelled $X\alpha = Y\beta$ such that the equations labelling the immediate successors of a node are determined by an application of one of the rules in accordance with the procedure described in this section.

The rules are built around equations $E\alpha = F\beta$ (where α, β could be the empty sequence of variables). Each rule has the form

$$\frac{E\alpha = F\beta}{E_1\alpha_1 = F_1\beta_1 \quad \cdots \quad E_n\alpha_n = F_n\beta_n}$$

(possibly with side conditions). The premise of a rule represents the goal to be achieved (that $E\alpha \sim F\beta$) while the consequents are the subgoals.

The rules are only applied to nodes that are not *terminal*. Terminal nodes are either *successful* or *unsuccessful*.

Definition 5 *A tableau node is called an* unsuccessful terminal *if it has one of the forms*

1. $\alpha = \beta$ *with* $|\alpha| \neq |\beta|$
2. $a\alpha = b\beta$ *with* $a \neq b$

Clearly, such nodes cannot relate bisimilar processes. In the following subsection we define the notion of successful termination.

3.1 Constructing subtableaux

A tableau consists of a number of *eliminating subtableaux* constructed using the rules REC, SUM, and PREFIX of Table 1. Each of these rules is *forwards sound* in the sense that if the antecedent is true (the equation relates bisimilar processes) then one can find a set of true consequents. This is expressed in the following proposition, whose proof is immediate:

Proposition 4 (Soundness of REC,SUM and PREFIX)

- *If* $X\alpha \sim Y\beta$ *and* $X \stackrel{\text{def}}{=} E$ *and* $Y \stackrel{\text{def}}{=} F$, *then* $E\alpha \sim F\beta$.
- *If* $(\sum_{i=1}^{m} a_i\alpha_i)\alpha \sim (\sum_{j=1}^{n} b_j\beta_j)\beta$ *then there exist functions* $f : \{1, \ldots, m\} \rightarrow \{1, \ldots, n\}$ *and* $g : \{1, \ldots, n\} \rightarrow \{1, \ldots, m\}$ *such that* $a_i\alpha_i\alpha \sim b_{f(i)}\beta_{f(i)}\beta$ *for* $1 \leq i \leq m$ *and* $a_{g(j)}\alpha_{g(j)}\alpha \sim b_j\beta_j\beta$ *for* $1 \leq j \leq n$.
- *If* $a\alpha \sim a\beta$ *then* $\alpha \sim \beta$.

Rules within subtableaux

REC $\dfrac{X\alpha = Y\beta}{E\alpha = F\beta}$ where $X \stackrel{\mathrm{def}}{=} E$ and $Y \stackrel{\mathrm{def}}{=} F$

PREFIX $\dfrac{a\alpha = a\beta}{\alpha = \beta}$

SUM $\dfrac{(\sum_{i=1}^{m} a_i\alpha_i)\alpha = (\sum_{j=1}^{n} b_j\beta_j)\beta}{\{a_i\alpha_i\alpha = b_{f(i)}\beta_{f(i)}\beta\}_{i=1}^{m} \ \{a_{g(j)}\alpha_{g(j)}\alpha = b_j\beta_j\beta\}_{j=1}^{n}}$

$$\textit{where} \quad \begin{array}{l} f : \{1,\ldots,m\} \rightarrow \{1,\ldots,n\} \\ g : \{1,\ldots,n\} \rightarrow \{1,\ldots,m\} \\ \textit{with } m,n \geq 1 \end{array}$$

Rules for new subtableaux

SUBL $\dfrac{\alpha_i\alpha = \beta_i\beta}{\alpha_i\gamma = \beta_i}$ where $\alpha = \gamma\beta$ is the residual

SUBR $\dfrac{\alpha_i\alpha = \beta_i\beta}{\alpha_i = \beta_i\gamma}$ where $\gamma\alpha = \beta$ is the residual

Table 1: The tableau rules

A subtableau is built from *basic steps*. See Figure 2.

Definition 6 *A* basic step *for $X\alpha = Y\beta$ consists of an application of* REC *followed by at most one application of* SUM *followed by an application of* PREFIX *to each of its consequents (assuming that no node encountered is an unsuccessful terminal). A* basic node *is any node of the form $\alpha' = \beta'$ where $\alpha', \beta' \in Var^*$.*

Corresponding to a basic step for $X\alpha = Y\beta$ is a set of single transition steps in the operational semantics, as $X\alpha \xrightarrow{a_i} \alpha_i$ and $Y\beta \xrightarrow{a_i} \beta_i$ for any consequent $\alpha_i = \beta_i$. By Proposition 2 we have that $length(\alpha_i) \leq 1 + length(X\alpha)$ and $length(\beta_i) \leq length(Y\beta) + 1$.

Definition 7 *Assume that $k = \min(|X|, |Y|)$. An* eliminating subtableau *for $X\alpha = Y\beta$ iterates the construction of basic steps to depth k.*

See Figure 3 for a sketch of an eliminating subtableau in the case when $|X| \leq |Y|$. Notice that if $\alpha' = \beta'$ is a leaf of an eliminating subtableau then $X\alpha \xrightarrow{w} \alpha'$ and $Y\beta \xrightarrow{w} \beta'$ for some w of length k.

In the case that $|X| \leq |Y|$ each leaf of an eliminating subtableau for $X\alpha = Y\beta$ is either labelled $\alpha = \gamma\beta$, which we call a *residual* of the subtableau, as X has been eliminated, or $\alpha_i\alpha = \beta_i\beta$ where α_i and β_i need not be empty. Since the number of iterations of basic steps is $|X|$ there must be at least one residual and α and β must persist as suffixes throughout the subtableau. For any such subtableau we pick one residual node and call it *the* residual. If instead $|Y| < |X|$ similar remarks would apply except that the residual then has the form $\gamma\alpha = \beta$.

The next step is to apply one of the SUB rules of Table 1 to each leaf *other than residuals* of an eliminating subtableau. If the residual is $\alpha = \gamma\beta$ we apply SUBL, and if it is $\gamma\alpha = \beta$ we apply SUBR. So assume $|X| \leq |Y|$; then for each leaf $\alpha_i\alpha = \beta_i\beta$ which is not a residual we obtain

$$\frac{\alpha_i\alpha = \beta_i\beta}{\alpha_i\gamma = \beta_i}\ \mathsf{SUBL} \qquad\qquad \text{where } \alpha = \gamma\beta \text{ is the residual}$$

If instead $|Y| < |X|$ so $\gamma\alpha = \beta$ is the residual, SUBR gives us the consequent $\alpha_i = \beta_i\gamma$. The SUB rules are also forwards sound in the following sense:

Proposition 5 (Soundness of SUBL and SUBR) *If $\alpha_i\alpha \sim \beta_i\beta$ and $\alpha \sim \gamma\beta$ then $\alpha_i\gamma \sim \beta_i$. If $\gamma\alpha_i \sim \beta_i$ then $\alpha_i \sim \beta_i\gamma$.*

PROOF: Since $\sim$ is a congruence, a substitution yields $\alpha_i\gamma\beta \sim \beta_i\beta$ and by Proposition 1 we get $\alpha_i\gamma = \beta$. The proof for the other half is entirely similar. □

From the above proof we see that the SUB rules should be thought of as two-step rules consisting of a *substitution* using the residual followed by a *reduction* of the length of the expressions involved according to Proposition 1. Notice that for any application of SUB we have that

Proposition 6 *For any eliminating subtableay with root $X\alpha = Y\beta$ and residual $\alpha = \gamma\beta$, for any leaf $\alpha_i\alpha = \beta_i\beta$ we have*

1. $|\alpha| < |X\alpha|$ *and* $|\gamma\beta| < |Y\beta|$
2. *length*(α_i) + *length*(γ) + *length*$(\beta_i) \leq 3m_\Delta + 1$ *for any application of* SUB.

The latter follows from Proposition 2. Also notice that the bound obtained here is completely independent of *length*(α) and *length*(β).

We can now define successful termination.

Definition 8 *A residual or a consequent of an application of a* SUB *rule is a* successful terminal *if it has one of the forms*

1. $\alpha = \beta$ *where there is a subtableau root above it also labelled* $\alpha = \beta$.
2. $\alpha = \alpha$

It should be obvious that a node obeying termination condition 2 in the above relates bisimilar processes. It turns out that this is also true of termination condition 1 in the context of a successful tableau.

When a consequent of SUB or the residual is not a terminal node we build a new eliminating subtableau with it as root as described above, and continue in this fashion. Therefore, a tableau is defined as successions of eliminating subtableaux as sketched in Figure 4.

Definition 9 *A* successful tableau *is a tableau all of whose leaves are successful terminals. If at any point in the construction of an eliminating subtableau we reach an unsuccessful terminal then the resulting tableau is* unsuccessful.

Example 3 (Example 2 continued) Consider again $\Delta = \{X \stackrel{\text{def}}{=} aYX + b, Y \stackrel{\text{def}}{=} bX, A \stackrel{\text{def}}{=} aC + b, C \stackrel{\text{def}}{=} bAA\}$. The tableau in Figure 5 is a successful tableau for $X = A$. □

3.2 Decidability, soundness, and completeness

We now give the proof of correctness of the tableau method and give a complexity measure in terms of an upper bound on the length of a tableau path.

Theorem 1 *Every tableau for $X\alpha = Y\beta$ is finite.*

Proof: If a tableau were infinite then it would have an infinite path. By definition such a path could not contain either successful or unsuccessful terminals. By Proposition 6(2) such a path can not pass through infinitely many nodes which are consequents of a SUB rule – for since there are only finitely many different equations with total norm $\leq 3m_\Delta + 1$ (by Proposition 3) the path would then contain some successful terminal infinitely often. Otherwise the path must almost always pass through a residual; but this also is impossible as the norm of a residual is strictly less than one directly above it by Proposition 6(1) and as the norms of the residuals are uniformly bounded by $\max(|\alpha|, |\beta|)$. □

We can give a complexity bound on the tableau method in terms of the longest possible path in any tableau for $X\alpha = Y\beta$. We measure the length of a path in terms of the total number of basic steps, since this gives a measure of the number of transition matches that we need to consider.

Theorem 2 *Any path in a tableau for $X\alpha = Y\beta$ has a length of at most*

$$m_\Delta \max(|\alpha|, |\beta|, m_\Delta \lceil \frac{3m_\Delta + 1}{2} \rceil \sum_{j=2}^{3m_\Delta+1} (j-1)v^j)$$

basic steps, where v is the cardinality of Var.

Proof: Any SUB consequent has a length of at most $3m_\Delta + 1$, so there can be at most $\sum_{j=2}^{3m_\Delta+1}(j-1)v^j$ distinct SUB consequents along a path. Between any two of these consequents there can be at most $\lceil \frac{3m_\Delta+1}{2} \rceil$ residuals, since the worst that can happen is that the norm on each side decreases by 1 between two consecutive residuals. Thus, any containing SUB consequents has at most $\lceil \frac{3m_\Delta+1}{2} \rceil \sum_{j=2}^{3m_\Delta+1}(j-1)v^j$ subtableau roots. The leftmost path in a tableau contains only residuals, and since their norms are strictly decreasing there can be at most $\max(|\alpha|, |\beta|)$ residuals along this path. Since any subtableau can have a depth of at most m_Δ basic steps, the result follows. □

Corollary 2 *There are only finitely many tableaux for any $X\alpha = Y\beta$.*

Proof: This follows from the above theorem and the fact that the branching at any basic step in any tableau is uniformly bounded by the maximal number of SUM consequents. This is bounded by $2B_\Delta$ where $B_\Delta = \max\{m \mid \exists X_i \in Var : X_i \stackrel{\text{def}}{=} \sum_{j=1}^m a_{ij}\alpha_{ij}\}$. □

The next theorem states soundness and completeness of the tableau method. The proof of soundness relies on the notion of self-bisimulation introduced in section 2.4.

Theorem 3 *$X\alpha \sim Y\beta$ iff there exists a successful tableau for $X\alpha = Y\beta$.*

PROOF: Suppose $X\alpha \sim Y\beta$. Then we can build a tableau for $X\alpha = Y\beta$ which has the property that for each node $\alpha' = \beta'$ we have $\alpha' \sim \beta'$. For by Propositions 4 we can at any point in the tableau construcion choose true consequents. By Theorem 1 this tableau construction must terminate and without unsuccessful terminals.

Now assume $\mathbf{T}$ is a successful tableau for $X\alpha = Y\beta$. We now show that $R_{\mathbf{T}} = \{(\alpha,\beta) \mid \alpha = \beta \text{ is a basic node in } \mathbf{T},\ \ \alpha,\beta \in Var^*\}$ is a self-bisimulation. By Lemma 1 this means that $X\alpha \sim Y\beta$.

So suppose $(\alpha',\beta') \in R_{\mathbf{T}}$. We must then show that $\alpha' \xrightarrow{a} \alpha''$ implies $\exists \beta'' :$ $\beta' \xrightarrow{a} \beta''$ with $\alpha'' \underset{R_{\mathbf{T}}}{\longleftrightarrow^*} \beta''$. $\alpha' = \beta'$ can either be a terminal or an internal node.

Suppose $\alpha' = \beta'$ is a terminal. If it is a terminal because of condition 2 we can certainly match within $\underset{R_{\mathbf{T}}}{\longleftrightarrow^*}$, since the least congruence of any relation contains the identity. Otherwise, if $\alpha' = \beta'$ is a terminal is due to condition 1, there is a previous occurrence of $\alpha' = \beta'$ as a subtableau root. Then we have the following basic step in $\mathbf{T}$:

$$\frac{\alpha' = \beta'}{\alpha_1'' = \beta_1'' \quad \cdots \quad \alpha_n = \beta_n''}$$

and $\alpha' \xrightarrow{a_i} \alpha_i''$ is matched by $\beta' \xrightarrow{a_i} \beta_i''$ because $\mathbf{T}$ is successful and $(\alpha_i'',\beta_i'') \in R_{\mathbf{T}}$ for $1 \leq i \leq n$ by definition of $R_{\mathbf{T}}$.

Otherwise $\alpha' = \beta'$ is an internal node. There are now two possibilities: either REC was applied to $\alpha' = \beta'$ or one of the SUB rules was.

Suppose REC was applied. Then we had the basic step

$$\frac{\alpha' = \beta'}{\alpha_1'' = \beta_1'' \quad \cdots \quad \alpha_n = \beta_n''}$$

and just as in the above case, we can match within $R_{\mathbf{T}}$.

Now suppose a SUB rule was applied. Suppose wlog that it was SUBL. Further assume that $\alpha' = \beta'$ is $X_1\alpha_1\alpha_0 = Y_1\beta_1\beta_0$, that $X_1 \stackrel{\text{def}}{=} \sum_{i=1}^m a_i\alpha_{2i}$ and $Y_1 \stackrel{\text{def}}{=} \sum_{j=1}^n b_j\beta_{2j}$ and that the residual was $\alpha_0 = \gamma_0\beta_0$:

$$\frac{X_1\alpha_1\alpha_0 = Y_1\beta_1\beta_0}{X_1\alpha_1\gamma_0 = Y_1\beta_1} \ \ \mathsf{SUBL}$$

By definition $(\alpha_0, \gamma_0\beta_0) \in R_{\mathbf{T}}$. Either $X_1\alpha_1\gamma_0 = Y_1\beta_1$ is a terminal or a subtableau root. In any case we must have that

$$\forall a_i : X_1\alpha_1\gamma_0 \xrightarrow{a_i} \alpha_{2i}\alpha_1\gamma_0 \;\; \exists b_j : b_j = a_i, Y_1\beta_1 \xrightarrow{b_j} \beta_{2j}\beta_1 \text{ with } \alpha_{2i}\alpha_1\gamma_0 \underset{R_{\mathbf{T}}}{\longleftrightarrow^*} \beta_{2j}\beta_1 \quad (1)$$

If $X_1\alpha_1\gamma_0 = Y_1\beta_1$ is a terminal, this follows from the same reasoning used in the case where $\alpha' = \beta'$ is a terminal. Otherwise, we have the basic step

$$\dfrac{\dfrac{X_1\alpha_1\gamma_0 = Y_1\beta_1}{(\sum_{i=1}^{m} a_i\alpha_{2i})\alpha_1\gamma_0 = (\sum_{j=1}^{n} b_j\beta_{2j})\beta_1}\ \mathsf{REC}}{\dfrac{a_1\alpha_{21}\alpha_1\gamma_0 = a_1\beta_{f(1)}\beta_1}{\alpha_{21}\alpha_1\gamma_0 = \beta_{f(1)}\beta_1}\ \mathsf{PREFIX} \quad \cdots \quad \dfrac{a_n\alpha_{g(n)}\alpha_1\gamma_0 = a_n\beta_{2n}\beta_1}{\alpha_{g(n)}\alpha_1\gamma_0 = \beta_{2n}\beta_1}\ \mathsf{PREFIX}}\ \mathsf{SUM}$$

Now, what are the transitions of $X_1\alpha_1\alpha_0$ and $Y_1\beta_1\beta_0$ and how do we match them ? For $1 \leq i \leq m$ we have that $X_1\alpha_1\alpha_0 \xrightarrow{a_i} \alpha_{2i}\alpha_1\alpha_0$ and for $1 \leq j \leq n$, $Y_1\beta_1\beta_0 \xrightarrow{b_j} \beta_{2j}\beta_1\beta_0$. For any $X_1\alpha_1\alpha_0 \xrightarrow{a_i} \alpha_{2i}\alpha_1\alpha_0$ there is (1) a $Y_1\beta_1\beta_0 \xrightarrow{b_{f(i)}} \beta_{2f(i)}\beta_1\beta_0$ such that $\alpha_{2i}\alpha_1\gamma_0 \underset{R_{\mathbf{T}}}{\longleftrightarrow^*} \beta_{2f(i)}\beta_1$. We now have the match, $\alpha_{2i}\alpha_1\alpha_0 \underset{R_{\mathbf{T}}}{\longleftrightarrow^*} \beta_{2f(i)}\beta_1\beta_0$. For since $\alpha_{2i}\alpha_1\gamma_0 \underset{R_{\mathbf{T}}}{\longleftrightarrow^*} \beta_{2f(i)}\beta_1$, also $\alpha_{2i}\alpha_1\gamma_0\beta_0 \underset{R_{\mathbf{T}}}{\longleftrightarrow^*} \beta_{2f(i)}\beta_1\beta_0$. Since $(\alpha_0, \gamma_0\beta_0) \in R_{\mathbf{T}}$ we then have $\alpha_{2i}\alpha_1\alpha_0 \underset{R_{\mathbf{T}}}{\longleftrightarrow^*} \beta_{2f(i)}\beta_1\beta_0$, as was to be shown. Finding a match for any $Y_1\beta_1\beta_0 \xrightarrow{b_j} \beta_{2j}\beta_1\beta_0$ is entirely similar. □

Corollary 3 *For any normed system of BPA equations in GNF Δ and $\alpha, \beta \in Var^*$ it is decidable whether $\alpha \sim \beta$.*

Proof: A decision procedure goes as follows: Enumerate all the finitely many tableaux for $\alpha = \beta$ (this is possible by Corollary 2). By Theorem 3 $\alpha \sim \beta$ iff we find a successful tableau. □

4 An equational theory

Besides yielding a straightforward decision procedure, the tableau technique can also be used to build a (weakly) sound and complete sequent-based equational theory for bisimulation equivalence of normed BPA processes given in 3-GNF. For all that is required is a family of sound rules that permit one to derive the roots of successful tableaux. The equational theory is somewhat non-standard in the arena of process algebras. As it depends on assumptions, it is different in style both from Milner's elegant equational theory for regular processes with an explicit fixed point operator μ [23] and the version in [4] without μ.

Since the theory is based on the tableau system from the previous section, we restrict our attention to normed systems of process equations in 3-GNF. Let Δ be such a system. The proof system appeals to *assumptions* of the form $X\alpha = Y\beta$. The basic sequent of the system has the form $\Gamma \vdash_\Delta E = F$ where Γ is a set of assumptions and E, F range over BPA expressions. A sequent is interpreted as follows:

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

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

The proof system is given in Table 2. Equivalence and congruence rules are R1-5. The rules R6-10 correspond to the BPA laws of [4]. R11 and R12 deal with recursion and have been dictated by the tableau method. R11 is an assumption *introduction* rule, justified by the interpretation of sequents described above. R12 is an assumption *elimination* or discharge rule, which at the same time is a version of fixed point induction. Notice that the rule is contextual in character, involving the BPA contexts $[\,]\alpha$ and $[\,]\beta$ where $[\,]$ is a 'hole'.

Definition 11 *A* proof *of $\Gamma \vdash_\Delta E = F$ is a finite proof tree with the root labelled by $\Gamma \vdash_\Delta E = F$, with leaves that are instances of the axioms* R1,R6-10 *or* R11 *and such that the parent of a set of nodes is determined by an application of one of the rules* R2-5 *or* R12. *If $\Gamma \vdash_\Delta E = F$ has a proof we simply write $\Gamma \vdash_\Delta E = F$.*

In our proof that the equational theory is weakly sound and complete it turns out to be easiest to prove that it is in fact *strongly* sound. We need to appeal to the standard characterization of the maximal strong bisimulation as a limit:

Definition 12 *For any transition graph $\mathcal{T} = (Pr, Act, \rightarrow)$ define the family of binary relations $\{\sim_n\}_{n=0}^{\omega}$ over Pr inductively as follows.*

- *$p \sim_0 q$ for all $p, q \in Pr$,*
- *$p \sim_{n+1} q$ iff*
  - *if $p \xrightarrow{a} p'$ then $\exists q'$ with $q \xrightarrow{a} q'$ and $p' \sim_n q'$ and*
  - *if $q \xrightarrow{a} q'$ then $\exists p'$ with $p \xrightarrow{a} p'$ and $p' \sim_n q'$.*

Theorem 4 [24] *For any image-finite transition graph we have*

$$\sim \; = \bigcap_{n=0}^{\omega} \sim_n$$

Equivalence

R1 $$\Gamma \vdash_\Delta E = E$$

R2 $$\frac{\Gamma \vdash_\Delta E = F}{\Gamma \vdash_\Delta F = E}$$

R3 $$\frac{\Gamma \vdash_\Delta E = F \quad \Gamma \vdash_\Delta F = G}{\Gamma \vdash_\Delta E = G}$$

Congruence

R4 $$\frac{\Gamma \vdash_\Delta E_1 = F_1 \quad \Gamma \vdash_\Delta E_2 = F_2}{\Gamma \vdash_\Delta E_1 + E_2 = F_1 + F_2}$$

R5 $$\frac{\Gamma \vdash_\Delta E_1 = F_1 \quad \Gamma \vdash_\Delta E_2 = F_2}{\Gamma \vdash_\Delta E_1 E_2 = F_1 F_2}$$

BPA axioms

R6 $$\Gamma \vdash_\Delta E + F = F + E$$

R7 $$\Gamma \vdash_\Delta (E + F) + G = E + (F + G)$$

R8 $$\Gamma \vdash_\Delta E + E = E$$

R9 $$\Gamma \vdash_\Delta (E + F)G = EG + FG$$

R10 $$\Gamma \vdash_\Delta E(FG) = (EF)G$$

Recursion

R11 $$\Gamma, X\alpha = Y\beta \vdash_\Delta X\alpha = Y\beta$$

R12 $$\frac{\Gamma, X\alpha = Y\beta \vdash_\Delta E\alpha = F\beta}{\Gamma \vdash_\Delta X\alpha = Y\beta} \qquad X \stackrel{\mathrm{def}}{=} E,\ Y \stackrel{\mathrm{def}}{=} F \in \Delta$$

Table 2: Rules of inference in the equational theory

Theorem 5 *If* $\Gamma \vdash_\Delta X\alpha = Y\beta$ *then* $\Gamma \models_\Delta X\alpha = Y\beta$

PROOF: We proceed by contraposition. Assume that we have a proof of $\Gamma \vdash_\Delta X\alpha = Y\beta$ but that $\Gamma \not\models_\Delta X\alpha = Y\beta$. Then $\{(X\alpha, Y\beta) \,|\, X\alpha = Y\beta \in \Gamma\} \cup \{(X_i, E_i) \,|\, X_i \stackrel{\mathrm{def}}{=} E_i \in \Delta\}$ is part of a bisimulation but $X\alpha \not\sim Y\beta$; consequently, as Δ defines an image-finite transition graph, by Theorem 4 we know that $X\alpha \not\sim_n Y\beta$ for some n.

Observe that if we ignore the hypotheses, then the rules preserve $\sim_n$ and all formulae that are instances of axioms other than R11 are true for all $\sim_n$. For instance, for R12 we have that if $E\alpha \sim_n F\beta$ then $X\alpha \sim_n Y\beta$ because $X \stackrel{\mathrm{def}}{=} E \in \Delta$ and $Y \stackrel{\mathrm{def}}{=} F \in \Delta$. Similarly, for R3 we have that $E \sim_n F$ and $F \sim_n G$ imply that $E \sim_n G$. Now significantly, in the case of R5 we can strengthen this to say that $E_1 \sim_n F_1$ and $E_2 \sim_{n-1} F_2$ imply that $E_1E_2 \sim_n F_1F_2$ (when $|E_1| > 0$.)

Now consider the proof tree for $\Gamma \vdash_\Delta X\alpha = Y\beta$. Since $X\alpha \not\sim_n Y\beta$, by the above observations, there is a path π to some leaf in the proof tree such that for every node $\Gamma_i \vdash_\Delta \alpha_i = \beta_i$ along π we have $\alpha_i \not\sim_{k_i} \beta_i$ for some k_i. For each i choose k_i such that it is the least number with this property.

What could the leaf of the path look like ? It cannot be an assumption in Γ, since Γ is part of a bisimulation. Nor can it be an identity. The only other possibility is that the leaf at the end of the path π is an instance of R11 of the form $\Gamma_m, X'_m\alpha'_m = Y'_m\beta'_m \vdash_\Delta X'_m\alpha'_m = Y'_m\beta'_m$ and such that $X'_m\alpha'_m \not\sim_{k_m} Y'_m\beta'_m$ where k_m is the least number with this property. Assume that $X'_m \stackrel{\mathrm{def}}{=} E' \in \Delta$ and $Y'_m \stackrel{\mathrm{def}}{=} F' \in \Delta$.

As $X'_m\alpha'_m = Y'_m\beta'_m$ has been eliminated as a hypothesis in the course of the proof, there must be an application of R12 on π with premise $\Gamma_i \vdash_\Delta E'\alpha'_m = F'\beta'_m$ somewhere on the path π. On the subpath between this premise and the leaf $\Gamma_m, X'_m\alpha'_m = Y'_m\beta'_m \vdash_\Delta X'_m\alpha'_m = Y'_m\beta'_m$ there must be at least one application of the congruence rule R5 in order to build up the expressions E' and F'. By the above observation on the soundness of R5 w.r.t. $\sim_n$ every node $\alpha_i = \beta_i$ on the subpath must have $\alpha_i \sim_k \beta_i$ for all $k < k_m$. Since E' and F' are guarded, in at least one application of R5 an equation derived from $X'_m\alpha'_m = Y'_m\beta'_m$ must be the right-hand premise (possibly simply to introduce action prefixes). Thus we must in fact have that $E'\alpha'_m \sim_k F'\beta'_m$ for some $k \geq k_m$. But this then implies that $X'_m\alpha'_m \sim_k Y'_m\beta'_m$ for some $k \geq k_m$, contradicting our assumption that $X'_m\alpha'_m \not\sim_{k_m} Y'_m\beta'_m$. □

The completeness proof depends on simulating the tableau construction using the proof rules. The thinning rule usually found in sequent-based proof systems is a derived rule in ours.

Lemma 2 (Thinning) *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 $X\alpha \sim Y\beta$ from a successful tableau for $X\alpha \sim Y\beta$.

Definition 13 *In a successful tableau* $\mathbf{T}$*, we define the set of* companion nodes $Com(E\alpha' = F\beta')$ *for a node* $E\alpha' = F\beta'$ *as the set of nodes along the path to the root of* $\mathbf{T}$ *that correspond to an instance of a successful terminal for termination condition 1.*

For any subtableau $\mathbf{T}'$ *of* $\mathbf{T}$ *the set* $Basic_{\mathbf{T}'}(E\alpha' = F\beta')$ *for a node* $E\alpha' = F\beta'$ *in* $\mathbf{T}'$ *is the set of basic nodes on the path starting above* $E\alpha' = F\beta'$ *and ending at the root of* $\mathbf{T}'$*.*

Proposition 7 *For any node* $E\alpha' = F\beta'$ *in a successful tableau* $\mathbf{T}$ *we have*

$$Com(E\alpha' = F\beta') \subseteq Basic_{\mathbf{T}}(E\alpha' = F\beta')$$

Lemma 3 *Let* $\mathbf{T}'$ *be a subtableau of a successful tableau* $\mathbf{T}$ *such that* $\mathbf{T}'$ *is built using only basic steps, has root* $X'\alpha' = Y'\beta'$ *and leaves* $\alpha_1 = \beta_1, \ldots, \alpha_n = \beta_n$*. If for some* Γ *we have* $\Gamma \vdash_\Delta \alpha_i = \beta_i$ *for* $1 \le i \le n$ *then* $\Gamma \vdash_\Delta X'\alpha' = Y'\beta'$ *with a proof tree with nodes of the form* $\Gamma, Basic_{\mathbf{T}'}(E''\alpha'' = F''\beta'') \vdash_\Delta E''\alpha'' = F''\beta''$ *for any node* $E''\alpha'' = F''\beta''$ *in* $\mathbf{T}'$*.*

Proof: Induction in d, the depth w.r.t. basic steps of $\mathbf{T}'$.

$d = 1$*:* $\mathbf{T}'$ consists of one basic step:

$$\dfrac{\dfrac{X'\alpha' = Y'\beta'}{(\sum_{i=1}^{m} a_i\alpha_i)\alpha' = (\sum_{j=1}^{n} b_j\beta_j)\beta'}\ \textsf{REC}}{\dfrac{a_1\alpha_1\alpha' = b_{f(1)}\beta_{f(1)}\beta'}{\alpha_1\alpha' = \beta_{f(1)}\beta'}\ \textsf{PREFIX} \quad \cdots \quad \dfrac{a_{g(n)}\alpha_{g(n)}\alpha' = b_n\beta_n\beta'}{\alpha_{g(n)}\alpha' = \beta_n\beta'}\ \textsf{PREFIX}}\ \textsf{SUM}$$

where $\Gamma \vdash_\Delta \alpha_i\alpha' = \beta_{f(i)}\beta'$ for $1 \le i \le m$ and $\Gamma \vdash_\Delta \alpha_{g(j)}\alpha' = \beta_j\beta'$ for $1 \le j \le n$. Since we have that

$$Basic_{\mathbf{T}'}(\alpha_i\alpha' = \beta_i\beta') = Basic_{\mathbf{T}'}(\alpha_j\alpha' = \beta_j\beta') = \{X'\alpha' = Y'\beta'\}$$

for any consequents, Lemma 2 tells us that

$$\Gamma, X'\alpha' = Y'\beta' \vdash_\Delta \alpha_i\alpha' = \beta_{f(i)}\beta' \text{ for } 1 \le i \le m$$

and

$$\Gamma, X'\alpha' = Y'\beta' \vdash_\Delta \alpha_{g(j)}\alpha' = \beta_j\beta' \text{ for } 1 \le j \le n$$

Repeated use of R5 followed by repeated use of R4 gives us

$$\Gamma, X'\alpha' = Y'\beta' \vdash_\Delta (\sum_{i=1}^{m} a_i\alpha_i)\alpha' = (\sum_{j=1}^{m} b_j\beta_j)\beta'$$

Finally, by R12 we get $\Gamma \vdash_\Delta X'\alpha' = Y'\beta$.

Step (assuming for d): The first basic step of the subtableau is as in the base case. By induction hypothesis we have that

$$\Gamma \vdash_\Delta \alpha_i\alpha' = \beta_{f(i)}\beta' \text{ for } 1 \leq i \leq m$$

and

$$\Gamma \vdash_\Delta \alpha_{g(j)}\alpha' = \beta_j\beta' \text{ for } 1 \leq j \leq n$$

And by Lemma 2 we get that

$$\Gamma, X'\alpha' = Y'\beta' \vdash_\Delta \alpha_i\alpha' = \beta_{f(i)}\beta' \text{ for } 1 \leq i \leq m$$

and

$$\Gamma, X'\alpha' = Y'\beta' \vdash_\Delta \alpha_{g(j)}\alpha' = \beta_j\beta' \text{ for } 1 \leq j \leq n$$

The proof now proceeds as for the base case. □

Lemma 4 *Given a successful tableau* $\mathbf{T}$*, for any* $X\alpha = Y\beta$ *that is a terminal or the root of an eliminating subtableau we have* $Com(X\alpha = Y\beta) \vdash_\Delta X\alpha = Y\beta$.

PROOF: Induction in the structure of $\mathbf{T}$.

Base case - $X\alpha = Y\beta$ *is a terminal:* $X\alpha = Y\beta$ is either a terminal due to termination condition 2 or termination condition 1. In the former case, R1 immediately gives us $Com(X\alpha = Y\beta) \vdash_\Delta X\alpha = Y\beta$. In the latter case, $X\alpha = Y\beta \in Com(X\alpha = Y\beta)$ so we get desired result by R11.

Step: Now $X\alpha = Y\beta$ is root of the subtableau $\mathbf{T}'$:

$$\begin{array}{c} X\alpha = Y\beta \\ \\ \alpha = \gamma\beta \cdots \quad \cdots \; \dfrac{\alpha_i\alpha = \beta_i\beta}{\alpha_i\gamma = \beta_i}\,\mathsf{SUBL} \; \cdots \end{array}$$

By induction hypothesis, we have $Com(\alpha = \gamma\beta) \vdash_\Delta \alpha = \gamma\beta$ and for any SUB consequent (assume wlog that it is SUBL) $Com(\alpha_i\gamma = \beta_i) \vdash_\Delta \alpha_i\gamma = \beta_i$. But since there are no terminals within $\mathbf{T}'$ we have that $Com(\alpha = \gamma\beta) = Com(\alpha_i\gamma = \beta_i)$. By $\mathsf{R1}$, we get $Com(\alpha = \gamma\beta) \vdash_\Delta \beta = \beta$ and by $\mathsf{R5}$ $Com(\alpha = \gamma\beta) \vdash_\Delta \alpha_i\gamma\beta = \beta_i\beta$. By $\mathsf{R3}$ this implies $Com(\alpha = \gamma\beta) \vdash_\Delta \alpha_i\alpha = \beta_i\beta$. By Lemma 3 we then get $Com(X\alpha = Y\beta) \vdash_\Delta X\alpha = Y\beta$ as desired. Note also that by Lemma 3 that for any node $E''\alpha'' = F''\beta''$ in $\mathbf{T}'$ we have $Basic_{\mathbf{T}'}(E''\alpha'' = F''\beta'') \vdash_\Delta E''\alpha'' = F''\beta''$. □

Theorem 6 *If $X\alpha \sim Y\beta$ (with respect to Δ) then $\emptyset \vdash_\Delta X\alpha = Y\beta$*

Proof: By Theorem 3 we know that $X\alpha = Y\beta$ has a successful tableau $\mathbf{T}$. For each node $E''\alpha'' = F''\beta$ in $\mathbf{T}$ we have that $Basic_{\mathbf{T}}(E''\alpha'' = F''\beta'') \vdash E''\alpha'' = F''\beta$. If $E''\alpha'' = F''\beta$ is a subtableau root or a terminal, this follows from Lemma 4, Proposition 7 and Lemma 2. If $E''\alpha'' = F''\beta$ is a node in an eliminating subtableau, it follows from the remarks at the end of the proof of Lemma 4 and Lemma 2. Since $Basic_{\mathbf{T}}(X\alpha = Y\beta) = \emptyset$, the result follows. □

5 Extracting fundamental relations

In Section 3 we have seen that the tableau system presented generates a self-bisimulation in case of successful termination. In this section we show another relationship with the work of Caucal [7] in that we give an *auxiliary tableau system* for extracting a *fundamental* relation R from a successful tableau for $X\alpha = Y\beta$ with the property that $X\alpha \underset{R}{\longleftrightarrow}^* Y\beta$.

Definition 14 [7] *A relation $R \subseteq Var^+ \times Var^+$ is called* fundamental *iff*

1. *$Dom(R) \subseteq Var$, $Im(R) \subseteq (Var \setminus Dom(R))^+$*
2. *R is a function: $\alpha R\beta$ and $\alpha R\gamma$ implies $\beta = \gamma$*
3. *$\alpha R\beta$ implies $|\alpha| = |\beta|$*

From the first condition above it is immediately seen that fundamental relations are finite and from the third condition one sees that there are finitely many fundamental relations for any normed BPA process (since there are only finitely many elements of Var^* with any given norm). Seen as a rewrite relation, if R is fundamental then it is also canonical, i.e. confluent and well-founded (this follows from the functionality of R and the finiteness of $Dom(R)$), and thus its least congruence is decidable.

One can think of the least congruence $\underset{R}{\longleftrightarrow}^*$ of a relation R as the set of equations provable within equational logic (with added congruence rules) using

R as axioms. Thus, we can we view a fundamental relation with the above property as constituting a 'local axiomatization' of $\sim$, relative to Δ and the root equation $X\alpha = Y\beta$.

Throughout the following we shall assume the existence of a successful tableau $\mathbf{T}$ for $X\alpha = Y\beta$.

The fundamental observation is that for the eliminating subtableau for $X\alpha = Y\beta$ we must, when $\alpha = \gamma\beta$ is the residual, have $Y\beta \sim X\gamma\beta$ and thus by Proposition 1 $Y \sim X\gamma$; assume now wlog that X and Y are not the same variable. Since $|Y| = |X\gamma|$ we know that Y does not occur in $X\gamma$, so $(Y, X\gamma)$ is a fundamental relation. Clearly, if we let $R = \{(\alpha, \gamma\beta), (Y, X\gamma)\}$ we have $X\alpha \underset{R}{\longleftrightarrow^*} Y\beta$. The auxiliary tableau system now gradually modifies and extends R until it becomes a fundamental relation with this property. While doing this we may need to introduce new goals.

The auxiliary tableau system is built around sequents of the form $R \vdash_{\mathbf{T}} \Gamma$ where R is a finite subset of $Var \times Var^+$ and Γ is a finite set of equations over Var^*. Since the relations R constructed are all fundamental (by Proposition 9 below), they are all confluent and strongly normalizing, so for any α its unique normal form $\alpha \downarrow R$ is known to exist.

At all times during the auxiliary tableau construction we rewrite as much of Γ as much as possible using R. We may then need to introduce new goals or extend R. There are in general three possible situations possibly at any point where this can happen:

- If an equation $X\alpha = Y\beta$ has the residual $\alpha = \gamma\beta$ and $R \cup \{(Y, X\gamma)\}$ is fundamental we simply extend R with the pair $(Y, X\gamma)$. This justifies the rule EXTEND.

- If an equation $X\alpha = Y\beta$ has the residual $\alpha = \gamma\beta$ but $R \cup \{(Y, X\gamma)\}$ is not fundamental because $Y \in Dom(R)$ with $(Y, X_1\gamma_1) \in R$ for some $X_1\gamma_1$. Then we must also compare $X_1\gamma_1$ and $X\gamma$. This gives rise to the rule COMPARE.

- If an equation $X\alpha = Y\beta$ has the residual $\alpha = \gamma\beta$ but $R \cup \{(Y, X\gamma)\}$ is not fundamental even though $Y \notin Dom(R)$ because some variable $Z \in Dom(R)$ occurs in $X\gamma$. We must rewrite $X\gamma$ and can then add $(Y, (X\gamma) \downarrow R)$ to R. This is the basis of the rule UPDATE.

The rule REWRITE tells us that we must rewrite using R whenever possible. And finally there are the rules CONGL and CONGR whose purpose is to remove identical heads and tails from equations. CONGR is strictly speaking not necessary but has been included for reasons of symmetry.

The rules have the following *priority*: CONGL must always be used whenever possible to remove identical leftmost variables. Next in priority is REWRITE.

EXTEND	$\dfrac{R \vdash_{\mathbf{T}} X\alpha = Y\beta, \Gamma}{R, Y = X\gamma \vdash_{\mathbf{T}} \alpha = \gamma\beta, \Gamma}$	if $\alpha = \gamma\beta$ is the residual of $X\alpha = Y\beta$ and $R, Y = X\gamma$ is fundamental
COMPARE	$\dfrac{R \vdash_{\mathbf{T}} X\alpha = Y\beta, \Gamma}{R \vdash_{\mathbf{T}} X_1\gamma_1 = X\gamma, \alpha = \gamma\beta, \Gamma}$	if $Y \in Dom(R)$ and $\alpha = \gamma\beta$ is the residual for $X\alpha = Y\beta$ but $(Y, X_1\gamma_1) \in R$
UPDATE	$\dfrac{R \vdash_{\mathbf{T}} X\alpha = Y\beta, \Gamma}{R, Y = (X\gamma) \downarrow R \vdash_{\mathbf{T}} (\alpha) \downarrow R = (\beta) \downarrow R, \Gamma}$	if $Y \notin Dom(R)$ and $\alpha = \gamma\beta$ is the residual for $X\alpha = Y\beta$ but some variable $Z \in Dom(R)$ occurs in $X\gamma$
REWRITE	$\dfrac{R \vdash_{\mathbf{T}} X\alpha = Y\beta, \Gamma}{R \vdash_{\mathbf{T}} (X\alpha) \downarrow R = (Y\beta) \downarrow R, \Gamma}$	if $(X\alpha) \downarrow R \neq X\alpha$ or $(Y\beta) \downarrow R \neq Y\beta$
CONGL	$\dfrac{R \vdash_{\mathbf{T}} \alpha\delta_1 = \alpha\delta_2, \Gamma}{R \vdash_{\mathbf{T}} \delta_1 = \delta_2, \Gamma}$	
CONGR	$\dfrac{R \vdash_{\mathbf{T}} \alpha_1\delta = \alpha_2\delta, \Gamma}{R \vdash_{\mathbf{T}} \alpha_1 = \alpha_2, \Gamma}$	

Table 3: Rules of the auxiliary tableau system

Finally, the other rules have equal priority. Thus we see that $\mathsf{REWRITE}$ will only be used between updatings of R, as desired.

The rules in Table 3 are sound w.r.t. $\sim$.

Definition 15 *A relation $R \subseteq Var \times Var^+$ is said to be* $\sim$ -consistent *if $R \subseteq \sim$. Similarly, a set of equations Γ is called* $\sim$ -consistent *if for all $\alpha' = \beta' \in \Gamma'$ we have $\alpha' \sim \beta'$.*

Proposition 8 (Soundness under $\sim$) *If $R' \vdash_{\mathbf{T}} \Gamma'$ has R' and Γ' $\sim$ -consistent then for the consequent $R'' \vdash_{\mathbf{T}} \Gamma''$ of any rule application to $R' \vdash_{\mathbf{T}} \Gamma'$, R'' and Γ'' are $\sim$ -consistent.*

The rules also preserve fundamentality:

Proposition 9 *If in $R' \vdash_{\mathbf{T}} \Gamma'$ we have that R' is fundamental and $\sim$ -consistent and Γ is $\sim$ -consistent, then for the consequent $R'' \vdash_{\mathbf{T}} \Gamma''$ of any rule application to $R' \vdash_{\mathbf{T}} \Gamma'$ we have that R'' is fundamental.*

We can now be precise about the notion of an auxiliary tableau.

Definition 16 *An* auxiliary tableau *for $R \vdash_{\mathbf{T}} \Gamma$ is a maximal sequence of sequents $R_0 \vdash_{\mathbf{T}} \Gamma_0, R_1 \vdash_{\mathbf{T}} \Gamma_1, \ldots, R_{n-1} \vdash_{\mathbf{T}} \Gamma_{n-1}, R_n \vdash_{\mathbf{T}} \Gamma_n$ where $R \vdash_{\mathbf{T}} \Gamma = R_0 \vdash_{\mathbf{T}} \Gamma_0$ and for all $i \geq 0$ $R_{i+1} \vdash_{\mathbf{T}} \Gamma_{i+1}$ is the consequent of using a rule in Table 3 with $R_i \vdash_{\mathbf{T}} \Gamma_i$ as premise. An auxiliary tableau is finite if for some n all equations in Γ_n are identities (i.e. of the form $\alpha' = \alpha'$ for some α').*

Lemma 5 *If Γ is* $\sim$ -consistent *all auxiliary tableaux for $\emptyset \vdash_{\mathbf{T}} \Gamma$ are finite.*

PROOF: Every time an equation $\alpha' = \beta'$ is replaced in a rule application, it is replaced by equations whose norms are all $\leq |\alpha'| = |\beta'|$. At least one of these new equations has norm $< |\alpha'|$. Thus we must eventually reach a situation where all equations in a sequent are identities, possibly of the form $\epsilon = \epsilon$. □

Theorem 7 *If R_0 is fundamental and $\sim$ -consistent and Γ_0 is $\sim$ -consistent, then for any finite auxiliary tableau $R_0 \vdash_{\mathbf{T}} \Gamma_0, \ldots, R_n \vdash_{\mathbf{T}} \Gamma_n$ we have $\alpha \underset{R}{\longleftrightarrow}^* \beta$ with $R = R_n$ for any $\alpha = \beta \in \Gamma_0$.*

PROOF: We proceed by induction in n.
$n = 1$: The auxiliary tableau is $R_0 \vdash_{\mathbf{T}} \Gamma, R \vdash_{\mathbf{T}} \Gamma'$, and every equation $\alpha = \beta$ in Γ' is an identity. Thus, obviously $\alpha \underset{R}{\longleftrightarrow}^* \beta$. We now proceed by case analysis, looking at the rule used.

If the rule was EXTEND, we have $\alpha \underset{R}{\longleftrightarrow}^* \gamma\beta$ and $\alpha' \underset{R}{\longleftrightarrow}^* \beta'$ for any $\alpha' = \beta' \in \Gamma'$. Therefore, $X\alpha \underset{R}{\longleftrightarrow}^* X\gamma\beta$ and $X\alpha \underset{R}{\longleftrightarrow}^* Y\beta$ as desired.

Had the rule been UPDATE, we would have $\alpha \downarrow R \underset{R}{\longleftrightarrow}^* (\gamma\beta) \downarrow R$ so $\alpha \underset{R}{\longleftrightarrow}^* \gamma\beta$ and $Y \underset{R}{\longleftrightarrow}^* X\gamma$, so again $X\alpha \underset{R}{\longleftrightarrow}^* X\gamma\beta$ implying $X\alpha \underset{R}{\longleftrightarrow}^* Y\beta$.

If the rule was REWRITE, we would have $(X\alpha) \downarrow R \underset{R}{\longleftrightarrow}^* (Y\beta) \downarrow R$ implying $X\alpha \underset{R}{\longleftrightarrow}^* Y\beta$.

And if the rule used had been COMPARE, we had $X_1\gamma_1 \underset{R}{\longleftrightarrow}^* X\gamma$ and $\alpha \underset{R}{\longleftrightarrow}^* \gamma\beta$ and $Y \underset{R}{\longleftrightarrow}^* X_1\gamma_1$. But then $Y\beta \underset{R}{\longleftrightarrow}^* X_1\gamma_1\beta$ and $X_1\gamma_1\beta \underset{R}{\longleftrightarrow}^* X\gamma\beta$, which implies that $Y\beta \underset{R}{\longleftrightarrow}^* X\gamma\beta$ and again $X\alpha \underset{R}{\longleftrightarrow}^* Y\beta$.

CONGL and CONGR are immediate.

Step: The auxiliary tableau is now $R_0 \vdash_{\mathbf{T}} \Gamma_0, R_1 \vdash_{\mathrm{T}} \Gamma_1 \ldots, R_n \vdash_{\mathbf{T}} \Gamma_n$ and $R_1 \vdash_{\mathbf{T}} \Gamma_1, \ldots, R_n \vdash_{\mathbf{T}} \Gamma_n$ is an auxiliary tableau for $R_1 \vdash_{\mathbf{T}} \Gamma_1$, so for every equation $\alpha' = \beta' \in \Gamma_1$ by induction hypothesis $\alpha' \underset{R}{\longleftrightarrow}^* \beta'$. The proof proceeds exactly as in the base case. □

From the above theorem and Lemma 5 we now get the desired result:

Corollary 4 *If $X\alpha \sim Y\beta$ by the successful tableau $\mathbf{T}$, for any auxiliary tableau $\emptyset \vdash_{\mathbf{T}} X\alpha = Y\beta, \ldots, R_n \vdash_{\mathbf{T}} \Gamma_n$ we have $X\alpha \underset{R}{\longleftrightarrow}^* Y\beta$ where $R = R_n$.*

6 Conclusions and perspectives

In this paper we have given an alternative and much simpler proof of the decidability of bisimulation equivalence for normed BPA processes, first proved by Baeten, Bergstra and Klop [1, 2] and later by Caucal [6, 7]. Our decidability proof uses a tableau system closely related to the branching algorithms employed in the study of equivalence problems in language theory [22, 13]. If a successful tableau for an equation $\alpha = \beta$ exists, the tableau provides us with a finite witness for a bisimulation containing (α, β), a self-bisimulation in the sense of [6, 7]. The tableau method allows us to extract a sound and complete sequent-based equational theory for bisimilarity over normed BPA processes in 3-GNF. Finally, we have shown how to extract a fundamental relation R (as in the work of [7]) from a successful tableau for $\alpha = \beta$ such that $\alpha \underset{R}{\longleftrightarrow}^* \beta$.

The present paper only considers the normed case. Does the tableau method extend to the full BPA calculus? In [12], it is shown that bisimulation equivalence is indeed decidable for all of BPA. The decidability proof relies on showing that the maximal bisimulation of any BPA transition graph is generated by a finite self-bisimulation; the decision algorithm consists of two semi-decision procedures: one searches for such a self-bisimulation, the other tries to find a bisimulation error. The method does not yield an equational theory, nor does it give us any clues as to the complexity of the bisimulation problem.

A complexity analysis in given in [21] by Huynh and Tian who show that the complexity of the bisimilarity problem for normed BPA is in Σ_2^p in the polynomialtime hierarchy. It would be interesting to see whether going to the full BPA will lead to any modifications of the complexity bound.

Recently, attention has been focused on the process calculus BPP where a non-communicating parallel (full merge) operator takes the place of sequencing. Christensen, Hirshfeld and Moller have proved [11, 10] that bisimulation equivalence by using a tableau technique similar to the one in this paper. Again, a by-product of the decision procedure is a sound and complete sequent-style equational theory for bisimilarity. Recent results indicate [17] that all other known equivalences are undecidable for BPP; bisimulation equivalence seems to have a very special status as far as decidability is concerned. However, even when a slight extensions of BPP is considered, namely that of restriction, bisimilarity also becomes undecidable [9]. An interesting question is what happens if instead BPP and BPA are combined. The resulting calculus is similar to the PA calculus considered in e.g. [4].

In Section 2.3 we noted the correspondence between BPA and context-free grammars. There is a well-known and fundamental correspondence between context-free grammars and pushdown automata as regards language definability but this breaks down when one considers the transition graphs under the notion of bisimulation equivalence. In [8], Caucal and Monfort show that pushdown automata are more expressive than BPA in the sense that there exist pushdown automata whose transition graphs are not bisimilar to any BPA transition graph. The question of whether bisimilarity is decidable for general pushdown automata is therefore nontrivial; it remains open at the time of writing.

Weak versions of behavioural equivalences arise when unobservable actions are considered. As the strong behavioural equivalences are special cases of their weak counterparts, it is clear by the results of [16] that all weak equivalences other than bisimilarity are undecidable for normed BPA with unobservable actions. There exist a number of weak versions of bisimilarity. In [19] it was shown by means of a tableau decision procedure that van Glabbeek's branching bisimulation equivalence [27] is decidable for normed BPA processes that cannot terminate silently. However, the question remains open for weak bisimulation equivalence even in the normed case; Proposition 1 does not hold for weak bisimilarity so a different approach must be used.

References

[1] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. In *LNCS 259*, pages 93–114. Springer-Verlag, 1987.

[2] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Decidability of bisimulation equivalence for processes generating context-free languages. Technical

Report CS-R8632, CWI, September 1987.

[3] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Control*, 60:109–137, 1984.

[4] J.A. Bergstra and J.W. Klop. Process theory based on bisimulation semantics. In J.W. de Bakker, W.P de Roever, and G. Rozenberg, editors, *LNCS 354*, pages 50–122. Springer-Verlag, 1988.

[5] J. Bradfield and C. Stirling. Verifying temporal properties of processes. In *LNCS 458*, pages 115–125. Springer-Verlag, 1990.

[6] D. Caucal. Graphes canoniques de graphes algébriques. Rapport de Recherche 872, INRIA, Juillet 1988.

[7] D. Caucal. Graphes canoniques de graphes algébriques. *Informatique théorique et Applications (RAIRO)*, 24(4):339–352, 1990.

[8] D. Caucal and R. Monfort. On the transition graphs of automata and grammars. Technical report, IRISA, 1990.

[9] S. Christensen. *Decidability and Decomposition in Process Algebras*. PhD thesis, University of Edinburgh, 1993.

[10] S. Christensen, Y. Hirshfeld, and F. Moller. Bisimulation is decidable for all basic parallel processes. In *Proceedings of CONCUR '93*. Springer-Verlag, 1993.

[11] S. Christensen, Y. Hirshfeld, and F. Moller. Decomposability, decidability and axiomatisability for bisimulation equivalence on basic parallel processes. In *Proceedings of LICS '93*. IEEE, Computer Society Press, June 1993.

[12] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. In W.R. Cleaveland, editor, *Proceedings of CONCUR '92, LNCS 630*, pages 138–147. Springer-Verlag, August 1992.

[13] B. Courcelle. An axiomatic approach to the Korenjak-Hopcroft algorithms. *Mathematical Systems Theory*, 16:191–231, 1983.

[14] J.F. Groote. A short proof of the decidability of bisimulationf or normed bpa-processes. Technical report, CWI, 1991.

[15] J.F. Groote and Hüttel. Undecidable equivalences for basic process algebra. Technical Report ECS-LFCS-91-169, Department of Computer Science, University of Edinburgh, August 1991.

[16] J.F. Groote and H. Hüttel. Undecidable equivalences for basic process algebra. *Information and Computation*, 1993. to appear.

[17] Y. Hirshfeld. Finitely generated processes, petri nets and the equivalence problem. Unpublished note, May 1993.

[18] J. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[19] H. Hüttel. Silence is golden: Branching bisimilarity is decidable for context-free processes. In *Proceedings of CAV91*. Springer-Verlag, 1991. To appear. The full version is available as Report ECS-LFCS-91-173, Department of Computer Science, University of Edinburgh.

[20] Dung T. Huynh and Lu Tian. On deciding readiness and failure equivalences for processes. Technical Report UTDCS-31-90, University of Texas at Dallas, September 1990.

[21] Dung T. Huynh and Lu Tian. Deciding bisimilarity of normed context-free processes is in σ_2^p. Technical Report UTDCS-1-92, University of Texas at Dallas, January 1992. To appear in *Theoretical Computer Science*.

[22] A.J. Korenjak and J.E. Hopcroft. Simple deterministic languages. In *Proceedings of Seventh Annual IEEE Symposium on Switching and Automata Theory*, pages 36–46, 1966.

[23] R. Milner. A complete inference system for a class of regular behaviours. *Journal of Computer and System Sciences*, 28:439–466, 1984.

[24] R. Milner. *Communication and Concurrency*. Prentice-Hall International, 1989.

[25] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings of 5th GI Conference LNCS 104*, pages 167–183. Springer-Verlag, 1981.

[26] C. Stirling and D. Walker. Local model checking in the modal mu-calculus. In *LNCS 351*, pages 369–383. Springer-Verlag, 1989.

[27] R.J. van Glabbeek. The linear time – branching time spectrum. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings of CONCUR 90,* Amsterdam, *LNCS 458*, pages 278–297. Springer-Verlag, 1990.

$$\dfrac{\dfrac{X\alpha = Y\beta}{E\alpha = F\beta}\ \mathsf{REC}}{\dfrac{a_1\alpha_1 = a_1\beta_1}{\alpha_1 = \beta_1}\ \mathsf{PREFIX}\quad \cdots \quad \dfrac{a_n\alpha_n = a_n\beta_n}{\alpha_n = \beta_n}\ \mathsf{PREFIX}}\ \mathsf{SUM}$$

Figure 2: A basic step in the tableau system

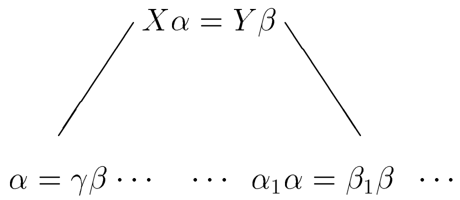

Figure 3: An eliminating subtableau for $X\alpha = Y\beta$.

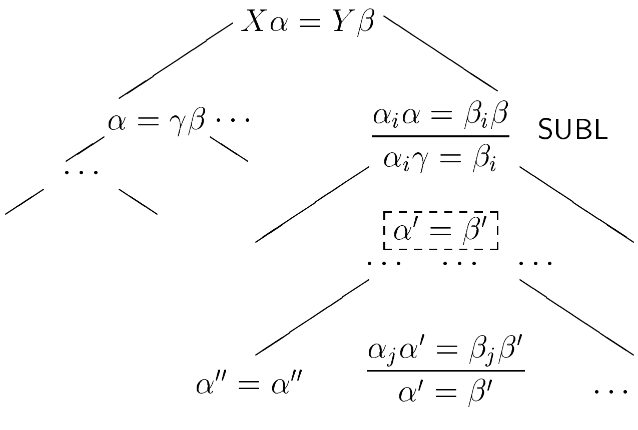

Figure 4: A tableau for $X\alpha = Y\beta$; some successful leaves are shown

$$
\dfrac{\dfrac{X = A}{aYX + b = aC + b}\ \textsf{REC}}{\dfrac{aYX = aC}{\dfrac{YX = C}{\dfrac{YX = C}{\dfrac{bXX = bAA}{\dfrac{XX = AA}{\dfrac{(aYX + b)X = (aC + b)A}{\dfrac{aYXX = aCA}{\dfrac{YXX = CA}{YX = C}\ \textsf{SUB}}\ \textsf{PREFIX} \quad \dfrac{bX = bA}{X = A}\ \textsf{PREFIX}}\ \textsf{SUM}}\ \textsf{REC}}\ \textsf{PREFIX}}\ \textsf{REC}}\ \textsf{SUB}}\ \textsf{PREFIX} \qquad \dfrac{b = b}{\epsilon = \epsilon}\ \textsf{PREFIX}}\ \textsf{SUM}
$$

Figure 5: A successful tableau for $X = A$ of Example 2