On the Axiomatizability of Priority ⋆

Luca Aceto[1,2], Taolue Chen[3,5], Wan Fokkink[3,4], and Anna Ingolfsdottir[1,2]

[1] Reykjavík University, School of Science and Engineering
Ofanleiti 2, 103 Reykjavík, Iceland
[2] BRICS, Aalborg University, Department of Computer Science
Fr. Bajersvej 7E, 9220 Aalborg Ø, Denmark
[3] CWI, Embedded Systems Group
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
[4] Vrije Universiteit, Section Theoretical Computer Science
Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
[5] Nanjing University, State Key Laboratory of Novel Software Technology
Nanjing, Jiangsu, P.R.China, 210093
luca@ru.is,chen@cwi.nl,wanf@cs.vu.nl,annai@ru.is

Abstract. This paper studies the equational theory of bisimulation equivalence over the process algebra BCCSP extended with the priority operator of Baeten, Bergstra and Klop. It is proven that, in the presence of an infinite set of actions, bisimulation equivalence has no finite, sound, ground-complete equational axiomatization over that language. This negative result applies even if the syntax is extended with an arbitrary collection of auxiliary operators, and motivates the study of axiomatizations using conditional equations. In the presence of an infinite set of actions, it is shown that, in general, bisimulation equivalence has no finite, sound, ground-complete axiomatization consisting of conditional equations over BCCSP. Sufficient conditions on the priority structure over actions are identified that lead to a finite, ground-complete axiomatization of bisimulation equivalence using conditional equations.

1 Introduction

Programming and specification languages often include constructs to specify mode switches (see, e.g., [17, 19]). Indeed, some form of mode transfer in computation appears in operating systems in the guise of interrupts, in programming languages as exceptions, and in the behaviour of control programs and embedded systems as discrete "mode switches" triggered by changes in the state of their environment. Such mode changes are often used to encode different levels

⋆ The first and fourth author were partly supported by the project "The Equational Logic of Parallel Processes" (nr. 060013021) of The Icelandic Research Fund. The second and third author were partly supported by the Dutch Bsik project BRICKS (Basic Research in Informatics for Creating the Knowledge Society). The second author was partly supported by 973 Program of China (No. 2002CB312002), NNSFC (No. 60233010, No. 60273034, No. 60403014).

of urgency amongst the actions that can be performed by a system as it computes, and implement variations on the notion of pre-emption. Classic process description languages include primitive operators to describe mode changes—for example, LOTOS [9] offers the so-called disruption operator—or have been extended with variations on mode transfer operators. Examples of such operators for the process algebra CCS are discussed by Milner in [18, pp. 192–193].

One of the most widely studied, and natural, notions used to implement different levels of urgency between system actions is priority. (A thorough and clear discussion of the different approaches to the study of priority in process description languages may be found in [12].) In this paper, we consider the well-known priority operator Θ studied by Baeten, Bergstra and Klop [5] in the context of process algebra. (See [10–13] for later accounts of this operator in the setting of process description languages.) The priority operator Θ gives certain actions priority over others based on an irreflexive partial ordering relation $<$ over the set of actions. Intuitively, $a < b$ is interpreted as "b has priority over a". This means that, in the context of the priority operator Θ, action a is pre-empted by action b. For example, if p is some process that can initially perform both a and b, then $\Theta(p)$ will initially only be able to execute the action b.

In their classic paper [5], Baeten, Bergstra and Klop provided a sound and ground-complete axiomatization for this operator modulo bisimulation equivalence. Their axiomatization uses predicates on actions (to express priorities between actions) and one extra auxiliary operator. Bergstra showed in the earlier paper [6] that, in case of a finite alphabet of actions, there exists a finite equational axiomatization for Θ, without action predicates and help operators. So, if the set of actions is finite, neither conditional equations nor auxiliary operators, as used in [5], are actually necessary to obtain a finite axiomatization of bisimulation equivalence over basic process description languages enriched with the priority operator. But, can Bergstra's positive result be extended to a setting with a countably infinite collection of actions? Or are conditional equations and auxiliary operators necessary to obtain a finite axiomatization of bisimulation equivalence in the presence of an infinite collection of actions? (Note that infinite sets of actions are common in process calculi, and arise, for instance, in the setting of value- or name-passing calculi.) The aim of this paper is to provide a thorough answer to these questions in the setting of the process algebra BCCSP enriched with the priority operator Θ. In case of an infinite alphabet, we permit the occurrence of action variables in axioms.

The process algebra BCCSP contains only basic process algebraic operators from CCS and CSP, but is sufficiently powerful to express all finite synchronization trees. This paper considers the equational theory of BCCSP with the priority operator Θ from [5] modulo bisimulation equivalence. Our first main result is a theorem indicating that the use of conditional equations is indeed inevitable in order to offer a finite axiomatization of bisimulation equivalence over the basic process language we consider in this study. To this end, we prove that, in case of an infinite alphabet and in the presence of at least one priority relation $a < b$ between a pair of actions, there is no finite equational axiomatization for BCCSP enriched with the priority operator (Theorem 2). This result even applies if one is allowed to add an arbitrary collection of help operators to the syntax. Theorem 2 offers a very strong indication that the use of conditional equations, where the conditions consist of action predicates, is essential for axiomatizing Θ, and cannot be circumvented by introducing auxiliary operators. (This is in contrast to the classic positive and negative results on the existence of finite equational axiomatizations for parallel composition offered in [7, 20, 21].)

Having established that conditional equations are necessary in order to obtain a finite, ground-complete equational axiomatization of bisimulation equivalence, we then proceed to investigate whether, in the presence of an infinite set of actions, this equivalence can be finitely axiomatized using conditional equations, but without auxiliary operators like the unless operator used in [5]. We show that, in general, the answer to this question is negative. This we do by exhibiting a priority structure with respect to which bisimulation equivalence affords no finite, sound and ground-complete axiomatization in terms of conditional equations (Theorem 3). This shows that, in general, the use of auxiliary operators is indeed necessary to axiomatize bisimulation equivalence finitely, even using conditional equations and over the simple language considered in this study.

In contrast to the aforementioned negative results, we exhibit a countably infinite, ground-complete axiomatization for bisimulation equivalence over BCCSP with the priority operator in terms of conditional equations (Theorem 4). This axiomatization suggests that infinite collections of pairwise incomparable actions with respect to the priority relation $<$ are the source of our negative result presented in Theorem 3.

Our results add the priority operator to the list of operators whose addition to a process algebra spoils finite axiomatizability modulo bisimulation equivalence; see, e.g., [2, 4, 20–22] for other examples of non-finite axiomatizability results over process algebras.

Most of the proofs have been omitted from this extended abstract; they can be found in the full version of the paper [1]. Only for the negative result in Section 5.1 do we provide a proof sketch.

2 Preliminaries

We begin by introducing the basic definitions and results on which the technical developments to follow are based.

2.1 The Language BCCSP$_\Theta$

Act denotes a non-empty alphabet of atomic actions, with typical elements a, b, c, d, e. Over *Act* we assume an irreflexive, transitive partial ordering $<$ to express priorities between actions. Intuitively, $a < b$ expresses that the action b has priority over the action a. We say that actions $a_1, \ldots, a_n$ are *incomparable* if they are distinct and $a_i < a_j$ does not hold for all $1 \leq i, j \leq n$.

The language of processes we shall consider in this paper, henceforth referred to as BCCSP$_\Theta$, is obtained by adding the unary priority operator Θ from [5] to the basic process algebra BCCSP [14, 15]. The language is given by the following grammar:

$$t ::= \mathbf{0} \mid a.t \mid t+t \mid \Theta(t) \mid x \mid \alpha.t \ ,$$

where a ranges over Act, x is a process variable and α is an action variable. Process and action variables range over given, disjoint countably infinite sets. We use x, y, z to range over the collection of process variables, and α, β as typical action variables. We use t, u, v to range over the collection of *open* process terms. A process term is *closed* if it does not contain any variables, and p, q, r, range over the set of closed terms T(BCCSP$_\Theta$). The *size* of a term is its length in function symbols.

A substitution maps each process variable to a process term, and each action variable to an action or action variable. A substitution is *closed* if it maps process variables to closed process terms and action variables to actions. For every term t and substitution σ, the term obtained by replacing occurrences of process variables x and action variables α in t with $\sigma(x)$ and $\sigma(\alpha)$ is written $\sigma(t)$.

The semantics of the operators is captured by the transition rules below, which give rise to Act-labelled transitions between closed terms:

$$\frac{}{a.x \xrightarrow{a} x} \qquad \frac{x_1 \xrightarrow{a} y}{x_1 + x_2 \xrightarrow{a} y} \qquad \frac{x_2 \xrightarrow{a} y}{x_1 + x_2 \xrightarrow{a} y} \qquad \frac{x \xrightarrow{a} y \quad x \stackrel{b}{\nrightarrow} \text{ for all } b \text{ such that } a < b}{\Theta(x) \xrightarrow{a} \Theta(y)}$$

where a ranges over Act. Intuitively, closed terms in the language BCCSP$_\Theta$ represent finite process behaviours, where $\mathbf{0}$ does not exhibit any behaviour, $p+q$ is the nondeterministic choice between the behaviours of p and q, and $a.p$ executes action a to transform into p. Furthermore, the process graph of $\Theta(p)$ is obtained by eliminating all transitions $q \xrightarrow{a} q'$ from the process graph of p for which there is a transition $q \xrightarrow{b} q''$ with $a < b$.

We consider the language BCCSP$_\Theta$ modulo bisimulation equivalence.

Definition 1. *A binary symmetric relation $\mathcal{R}$ over* T(BCCSP$_\Theta$) *is a* bisimulation *if $p \mathrel{\mathcal{R}} q$ together with $p \xrightarrow{a} p'$ imply $q \xrightarrow{a} q'$ for some q' with $p' \mathrel{\mathcal{R}} q'$. We write $p \leftrightarroweq q$ if there is a bisimulation relating p and q. The relation $\leftrightarroweq$ will be referred to as* bisimulation equivalence *or* bisimilarity.

It is well-known that $\leftrightarroweq$ is an equivalence relation. Moreover, the transition rules are in the GSOS format of [8]. Hence, bisimulation equivalence is a congruence with respect to all the operators in the signature of BCCSP$_\Theta$, meaning that $p \leftrightarroweq q$ implies $C[p] \leftrightarroweq C[q]$ for each BCCSP$_\Theta$-context $C[]$.

We can therefore consider the algebra of the closed terms in T(BCCSP$_\Theta$) modulo $\leftrightarroweq$. In Section 4, we shall offer results that apply to any signature Σ that extends the one of BCCSP$_\Theta$. To this end, we shall tacitly assume that all of the new operators in Σ also preserve bisimulation equivalence, and are semantically interpreted as operations over finite synchronization trees [18].

2.2 Equational Logic

An *axiom system* is a collection of equations $t \approx u$ over the language BCCSP_Θ. An equation $t \approx u$ is derivable from an axiom system E, notation $E \vdash t \approx u$, if it can be proven from the axioms in E using the rules of equational logic (viz. reflexivity, symmetry, transitivity, substitution and closure under BCCSP_Θ contexts). Without loss of generality one may assume that substitutions happen first in equational proofs, i.e., that the rule $\frac{t\approx u}{\sigma(t)\approx\sigma(u)}$ may only be used when $t \approx u \in E$. Moreover, by postulating that for each axiom in E also its symmetric counterpart is present in E, we can disregard applications of symmetry in equational proofs. In the remainder of this paper, we shall tacitly assume that our equational axiom systems are closed with respect to symmetry. Furthermore, it is well-known (cf., e.g., Section 2 in [16]) that if an equation relating two closed terms can be proven from an axiom system E, then there is a closed proof for it. (A proof is *closed* if it only mentions closed terms.) We shall only consider questions related to the provability of closed equations from an axiom system. Therefore, in light of the previous observation, we can restrict ourselves to considering closed proofs.

An equation $t \approx u$ is *sound* with respect to $\underline{\leftrightarrow}$ if $\sigma(t) \underline{\leftrightarrow} \sigma(u)$ holds for each closed substitution σ. An axiom system E is called *sound* over some language modulo $\underline{\leftrightarrow}$ if $E \vdash t \approx u$ implies $t \underline{\leftrightarrow} u$, for all terms t, u in the language. Conversely, E is called *ground-complete* if $p \underline{\leftrightarrow} q$ implies $E \vdash p \approx q$, for all *closed* terms p, q in the language.

Our order of business in the remainder of this paper will be to offer a thorough study of the equational theory of the language BCCSP_Θ modulo bisimulation equivalence. We begin our investigation by considering the case in which the set of actions *Act* is finite. We then move on to investigate the equational properties of bisimulation equivalence over BCCSP_Θ when the set of actions is infinite.

3 $|Act| < \infty$

In this section, we assume that the action set is finite. The axiom system in Table 1 was put forward by Jan Bergstra in [6]. Note that, in the case of a finite action set, this axiom system is finite, since then the axiom schemas PR2–4 give rise to finitely many equations.

Theorem 1 (Bergstra [6]). *The axiom system (A1)–(A4) and (PR1)–(PR4) is sound and ground-complete for* BCCSP_Θ *modulo* $\underline{\leftrightarrow}$.

In the remainder of this paper, process terms are considered modulo associativity and commutativity of $+$. We use $\sum_{i=1}^{n} t_i$ to denote $t_1 + \cdots + t_n$, where the empty sum represents $\mathbf{0}$. Modulo the axioms (A1) and (A2), every term t in the language BCCSP_Θ has the form $\sum_{i=1}^{n} t_i$, where the terms t_i do not have the form $t' + t''$. The terms t_i are called the *summands* of t.

A1	$x + y$	$\approx$	$y + x$	
A2	$x + (y + z)$	$\approx$	$(x + y) + z$	
A3	$x + x$	$\approx$	x	
A4	$x + \mathbf{0}$	$\approx$	x	
PR1	$\Theta(\mathbf{0})$	$\approx$	$\mathbf{0}$	
PR2	$\Theta(a.x + a.y + z)$	$\approx$	$\Theta(a.x + z) + \Theta(a.y + z)$	
PR3	$\Theta(a.x + b.y + z)$	$\approx$	$\Theta(b.y + z)$	$(a < b)$
PR4	$\Theta(a_1.x_1 + \cdots + a_n.x_n)$	$\approx$	$a_1.\Theta(x_1) + \cdots + a_n.\Theta(x_n)$	$(a_1, \ldots, a_n$ incomparable)

Table 1. Axiomatization in case of $|Act| < \infty$

4 $|Act| = \infty$

In this section, we deal with the case that the action set is infinite. Our main result is that bisimulation equivalence does *not* afford a finite equational axiomatization over the language BCCSP$_\Theta$, provided that *Act* contains at least two actions a, b with $a < b$. (Otherwise, the equation $\Theta(x) \approx x$ would be sound, and the priority operator could be eliminated from all terms.) This negative result even applies if BCCSP$_\Theta$ is extended with an arbitrary collection of operators (over finite synchronization trees) for which bisimulation is a congruence.

The idea behind the proof of our main result of this section is that a finite axiom system E can mention only finitely many action names. So, since *Act* is infinite, we can find a pair c, d of distinct actions that do not occur in E. If c and d are incomparable, then the equation $\Theta(c.\mathbf{0} + d.\mathbf{0}) \approx c.\mathbf{0} + d.\mathbf{0}$ is sound; if $c < d$, then $\Theta(c.\mathbf{0} + d.\mathbf{0}) \approx d.\mathbf{0}$ is sound. In the first case, we show that an equational proof of $\Theta(c.\mathbf{0} + d.\mathbf{0}) \approx c.\mathbf{0} + d.\mathbf{0}$ from E would give rise to a proof of the unsound equation $\Theta(a.\mathbf{0} + b.\mathbf{0}) \approx a.\mathbf{0} + b.\mathbf{0}$ from E. This follows by a simple renaming argument, using that c and d do not occur in E. Likewise, in the second case, a proof of $\Theta(c.\mathbf{0} + d.\mathbf{0}) \approx d.\mathbf{0}$ from E would give rise to a proof of the unsound equation $\Theta(d.\mathbf{0} + c.\mathbf{0}) \approx c.\mathbf{0}$ from E.

Theorem 2. *Let $|Act| = \infty$, and $a < b$ for some $a, b \in Act$. Let Σ be a signature consisting of the operators in* BCCSP$_\Theta$*, together with auxiliary operators for which bisimulation equivalence is a congruence. Then bisimulation equivalence has no finite, sound and ground-complete axiomatization over* $\mathrm{T}(\Sigma)$*.*

5 Axiomatizing Priority Conditionally

Theorem 2 offers very strong evidence that, in the presence of an infinite set of actions, equational logic is inherently not sufficiently powerful to achieve a finite axiomatization of bisimilarity over closed terms in the language BCCSP$_\Theta$. Indeed, that result holds true even in the presence of an arbitrary number of auxiliary operators.

In the presence of action variables, it is natural to view our language as consisting of two sorts: one for actions and the other for processes. This is all the more true because the set of actions has the structure of a partial order, and we should like to express axioms over processes that reflect the influence that this poset structure on actions has on the behaviour of processes. In case our set of actions is finite, this can be done by means of a finite number of equations that are instances of (PR3) and (PR4) in Table 1.

In the presence of an infinite action set, however, the axiom schemas (PR3) and (PR4), as well as (PR2), have infinitely many instances. One way to capture their effects finitely is, in the presence of action variables, to phrase the equation schemas (PR3) and (PR4) as conditional equations thus:

$$\text{(CPR3)} \quad (\alpha < \beta) \ \Rightarrow \ \Theta(\alpha.x + \beta.y + z) \ \approx \ \Theta(\beta.y + z)$$

$$\text{(CPR4)}_n \quad \Big(\bigwedge_{1 \leq i,j \leq n} \neg(\alpha_i < \alpha_j)\Big) \ \Rightarrow$$
$$\Theta(\alpha_1.x_1 + \cdots + \alpha_n.x_n) \ \approx \ \alpha_1.\Theta(x_1) + \cdots + \alpha_n.\Theta(x_n) \quad (n \geq 0) \ .$$

In both of the above conditional equations, we use predicates over actions to restrict the applicability of the equation on the right-hand side of the implication. In general, henceforth in this study we shall consider conditional equations of the form $P \Rightarrow t \approx u$, where P is a predicate over actions, and $t \approx u$ is an equation over the language BCCSP_Θ. In what follows, we shall assume that predicates over actions are expressed using formulae in first-order logic with equality and the binary relation symbol $<$.

The semantics of a predicate P is given by the collection of closed substitutions that satisfy it. If P is a tautology, then we simply write $t \approx u$. For instance, a version of of equation (PR2) with action variables will be written thus:

$$\text{(CPR2)} \quad \Theta(\alpha.x + \alpha.y + z) \ \approx \ \Theta(\alpha.x + z) + \Theta(\alpha.y + z) \ .$$

Note that equation (PR1) in Table 1 is just (CPR4)_0. Moreover, since $<$ is irreflexive, the conditional equation (CPR4)_1 reduces to $\Theta(\alpha.x) \approx \alpha.\Theta(x)$. (Note that this equation can be derived from each of the (CPR4)_n with $n \geq 1$ and (A3).)

A conditional equation $P \Rightarrow t \approx u$ is *sound* with respect to bisimilarity, if $\sigma(t) \leftrightarroweq \sigma(u)$ holds for each closed substitution σ that satisfies predicate P. It is not hard to see that for each partial order of actions $(Act, <)$, the conditional equations (CPR2), (CPR3) and (CPR4)_n $(n \geq 0)$ are sound modulo bisimilarity over the language BCCSP_Θ.

A natural question to ask at this point, and one that we shall address in the remainder of this study, is whether, unlike standard equational logic, conditional equations suffice to obtain a finite, ground-complete axiomatization of bisimulation equivalence over the language BCCSP_Θ.

In their classic paper [5], Baeten, Bergstra and Klop offered a finite, conditional, ground-complete axiomatization of bisimilarity over the language BPA_δ with the priority operator. Their axiomatization, however, relied upon the introduction of a binary auxiliary operator, the so-called *unless* operator $\triangleleft$, whose

$$
\begin{array}{rcl}
\Theta(\alpha.x) & \approx & \alpha.x \\
\Theta(\mathbf{0}) & \approx & \mathbf{0} \\
\Theta(x+y) & \approx & (\Theta(x) \lhd y) + (\Theta(y) \lhd x) \\
\neg(\alpha < \beta) \ \Rightarrow (\alpha.x) \lhd (\beta.y) & \approx & \alpha.x \\
(\alpha < \beta) \ \Rightarrow (\alpha.x) \lhd (\beta.y) & \approx & \mathbf{0} \\
(\alpha.x) \lhd \mathbf{0} & \approx & \alpha.x \\
\mathbf{0} \lhd (\alpha.x) & \approx & \mathbf{0} \\
(x+y) \lhd z & \approx & (x \lhd z) + (y \lhd z) \\
x \lhd (y+z) & \approx & (x \lhd y) \lhd z
\end{array}
$$

Table 2. Axioms for Θ in the presence of $\lhd$

transition rules are:

$$\frac{x \xrightarrow{a} x' \quad y \stackrel{b}{\nrightarrow} \text{ for all } b \text{ such that } a < b}{x \lhd y \xrightarrow{a} x'} \text{ , where } a \in \mathit{Act} \ .$$

In the setting of BCCSP$_\Theta$, and using action variables in lieu of concrete action names, the relation between the priority operator and the unless operator is expressed by the conditional equations in Table 2. It is not too hard to see that those conditional equations, together with (A1)–(A4) in Table 1, yield a ground-complete, finite, conditional equational axiomatization of bisimulation equivalence. Therefore, even in the presence of an infinite set of actions, bisimulation equivalence affords a finite, ground-complete axiomatization using conditional equations at the price of introducing a single auxiliary operator. But, if the set of actions is infinite, is the use of an auxiliary operator like the unless operator necessary to obtain a finite axiomatizability result for bisimulation equivalence over BCCSP$_\Theta$ using conditional equations?

5.1 A Negative Result

Our order of business will now be to prove that, in the presence of an infinite set of actions, in general auxiliary operators are indeed necessary in order to obtain a finite ground-complete axiomatization of bisimulation equivalence over the language BCCSP$_\Theta$. In this section, $\mathit{Act} = \{a_i, b_i \mid i \geq 1\} \cup \{c\}$, where $a_i < b_i < c$ for each $i \geq 1$, and these are the only inequalities. For convenience, we consider terms not only modulo associativity and commutativity of $+$, but also modulo the sound equations $x + \mathbf{0} \approx x$ and $\Theta(\Theta(x) + y) \approx \Theta(x + y)$.

The following lemma is the crux in the proof of Theorem 3. It states a property of closed terms that holds for all of the closed instantiations of axioms in any sound collection of conditional equations. In [3, Section 2.3] this is referred to as a proof-theoretic technique to prove that there is no finite basis for the equational theory. We use Φ_n to abbreviate $\sum_{i=1}^{n} b_i.\mathbf{0}$.

Lemma 1. *Let $P \Rightarrow t \approx u$ be a conditional equation that is sound modulo $\underline{\leftrightarrow}$. Let σ be a closed substitution with $\sigma(P) = \mathit{true}$. Assume that:*

– *n is larger than the size of t, where $n \geq 2$; and*
– *the summands of $\sigma(t)$ are all bisimilar to either Φ_n or $\mathbf{0}$.*

Then the summands of $\sigma(u)$ are all bisimilar to either Φ_n or $\mathbf{0}$.

Proof. The claim is easily seen to hold if $\sigma(t) \leftrightarrow \mathbf{0}$. Assume therefore that some summand of $\sigma(t)$ is bisimilar to Φ_n. Then $\sigma(t) \leftrightarrow \sigma(u) \leftrightarrow \Phi_n$.

Write $t = \sum_{i \in I} t_i$ and $u = \sum_{j \in J} u_j$ for some non-empty, finite index sets I and J, where the terms t_i and u_j are of the form x, $a.v$, $\alpha.v$ or $\Theta(v)$. By the proviso of the lemma, for each $i \in I$, the summands of $\sigma(t_i)$ are all bisimilar to Φ_n or $\mathbf{0}$. Since $n \geq 2$, for each $i \in I$, the term t_i is not of the form $a.v$ or $\alpha.v$. Hence either it is a process variable x, or it is of the form $\Theta(\sum_{\ell \in L_i} d_{i\ell}.t'_{i\ell} + \sum_{m \in M_i} \alpha_m.t''_{im} + \sum_{k \in K_i} z_{ik})$ (modulo $x + \mathbf{0} \approx x$ and $\Theta(\Theta(x) + y) \approx \Theta(x + y)$). Let $I' \subseteq I$ be the set of indices of summands of t that have the above form. Observe that $K_i \neq \emptyset$ for each $i \in I'$ such that $\sigma(t_i)$ is bisimilar to Φ_n (because n is larger than the size of t). Note moreover that summands t_i of t having the above form such that $\sigma(t_i) \leftrightarrow \mathbf{0}$ must have $L_i = M_i = \emptyset$, and for such summands $\sigma(z_{ik}) \leftrightarrow \mathbf{0}$ for each $k \in K_i$.

Let us assume, towards a contradiction, that there is an index $j \in J$ such that $\sigma(u_j)$ has a summand that is bisimilar neither to Φ_n nor to $\mathbf{0}$. We proceed by a case analysis on the form of u_j. The cases where u_j is of the form x, $a.u'_j$ or $\alpha.u'_j$ are easy and are omitted here. We focus on the case where $u_j = \Theta(u')$. Then u_j consists of a single summand, so by assumption, $\sigma(u_j) \not\leftrightarrow \Phi_n$ and $\sigma(u_j) \not\leftrightarrow \mathbf{0}$.

Since $\sigma(u) \leftrightarrow \Phi_n$, u' is of the form $\sum_{\ell \in L} e_\ell.u'_\ell + \sum_{m \in M} \beta_m.u''_m + \sum_{k \in K} y_k$. We distinguish two cases.

1. For each $i \in I'$ with $\sigma(t_i) \not\leftrightarrow \mathbf{0}$ there is a $k_i \in K_i$ such that z_{ik_i} is not a summand of u'.
   Define the substitution σ' as $\sigma'(y) = c.\mathbf{0}$ if either $y = z_{ik_i}$ for some $i \in I'$ with $\sigma(t_i) \not\leftrightarrow \mathbf{0}$ or if y is a summand of t with $\sigma(y) \not\leftrightarrow \mathbf{0}$, and let σ' agree with σ on other process variables and on action variables. It is not hard to see that $\sigma'(t) \stackrel{b_i}{\nrightarrow}$ for $i = 1, \ldots, n$ (because $c > b_i$ and t has no summand of the form $a.v$ or $\alpha.v$). On the other hand, since $\sigma(u_j) \not\leftrightarrow \mathbf{0}$ and $\sigma(u) \leftrightarrow \Phi_n$, there is an h with $1 \leq h \leq n$ such that $\sigma(u') \stackrel{b_h}{\rightarrow}$. Furthermore, $\sigma(u') \stackrel{c}{\nrightarrow}$. By assumption, z_{ik_i} is not a summand of u' for each $i \in I'$ with $\sigma(t_i) \not\leftrightarrow \mathbf{0}$. Moreover, for any variable summand y of t with $\sigma(y) \not\leftrightarrow \mathbf{0}$, y is not a summand of u', because by assumption $\sigma(y) \leftrightarrow \Phi_n$ while $\sigma(u') \not\leftrightarrow \Phi_n$. So $\sigma(u') \stackrel{b_h}{\rightarrow}$ and $\sigma(u') \stackrel{c}{\nrightarrow}$ imply $\sigma'(u') \stackrel{b_h}{\rightarrow}$ and $\sigma'(u') \stackrel{c}{\nrightarrow}$. It follows that $\sigma'(u_j) \stackrel{b_h}{\rightarrow}$, and so $\sigma'(u) \stackrel{b_h}{\rightarrow}$. Hence $\sigma'(t) \not\leftrightarrow \sigma'(u)$. Since $\sigma'(P) = \sigma(P) = \mathit{true}$, this contradicts the fact that $P \Rightarrow t \approx u$ is sound modulo $\leftrightarrow$.
2. $\{z_{i_0k} \mid k \in K_{i_0}\} \subseteq \{y_k \mid k \in K\}$, for some $i_0 \in I'$ with $\sigma(t_{i_0}) \not\leftrightarrow \mathbf{0}$. In this case, K is non-empty since, as previously observed, K_{i_0} is non-empty. By the proviso of the lemma, $\sigma(t_{i_0}) \leftrightarrow \Phi_n$, so (since n is larger than the size of t_{i_0}) there is a $k_0 \in K_{i_0}$ with $\sigma(z_{i_0k_0}) \not\leftrightarrow \mathbf{0}$. Furthermore, by assumption, $\sigma(u_j) \not\leftrightarrow \mathbf{0}$ and $\sigma(u_j) \not\leftrightarrow \Phi_n$. Therefore, there is an h with $1 \leq h \leq n$ such

that $\sigma(\Theta(u')) \stackrel{b_h}{\nrightarrow}$. Define the substitution σ' as $\sigma'(y) = a_h.\mathbf{0}$ if $y = z_{i_0k_0}$, and let σ' agree with σ on other process variables and on action variables. We argue that $\sigma'(t) \stackrel{a_h}{\nrightarrow}$. To this end, observe, first of all, that, since $\sigma(\Theta(u')) \stackrel{b_h}{\nrightarrow}$, we have $\sigma(\sum_{k\in K} y_k) \stackrel{b_h}{\nrightarrow}$, and so $\sigma(z_{i_0k_0}) \stackrel{b_h}{\nrightarrow}$. We are now ready to show that no summand of $\sigma'(t)$ affords an a_h-labelled transition. We consider three exhaustive possibilities:

(a) Let $i \in I'$ with $z_{i_0k_0} \notin \{z_{ik} \mid k \in K_i\}$. Then clearly $\sigma'(t_i) \stackrel{a_h}{\nrightarrow}$.

(b) Let $i \in I'$ with $z_{i_0k_0} \in \{z_{ik} \mid k \in K_i\}$. Then $\sigma(t_i) \not\underline{\leftrightarrow}\ \mathbf{0}$ because $\sigma(z_{i_0k_0}) \not\underline{\leftrightarrow}\ \mathbf{0}$, so by assumption $\sigma(t_i) \underline{\leftrightarrow}\ \Phi_n$. This implies $\sigma(t_i) \stackrel{b_h}{\rightarrow}$, so since $\sigma(z_{i_0k_0}) \stackrel{b_h}{\nrightarrow}$, it follows that $\sigma'(t_i) \stackrel{b_h}{\rightarrow}$. Since the outermost function symbol of t_i is Θ, we can conclude that $\sigma'(t_i) \stackrel{a_h}{\nrightarrow}$.

(c) Finally, since $\sigma(z_{i_0k_0}) \not\underline{\leftrightarrow}\ \mathbf{0}$ and $\sigma(z_{i_0k_0}) \stackrel{b_h}{\nrightarrow}$, the proviso of the lemma yields that $z_{i_0k_0}$ cannot be a summand of t.

From the three cases above we can conclude that $\sigma'(t) \stackrel{a_h}{\nrightarrow}$. On the other hand, $\sigma'(\Theta(u')) \stackrel{a_h}{\rightarrow}$, because $\sigma(\Theta(u')) \stackrel{b_h}{\nrightarrow}$ and $z_{i_0k_0} \in \{y_k \mid k \in K\}$. Hence $\sigma'(u) \stackrel{a_h}{\rightarrow}$, and so $\sigma'(t) \not\underline{\leftrightarrow}\ \sigma'(u)$. Since $\sigma'(P) = \sigma(P) = \mathit{true}$, this contradicts the fact that $P \Rightarrow t \approx u$ is sound modulo $\underline{\leftrightarrow}$.

In summary, the assumption that some $\sigma(u_j)$ has a summand that is bisimilar neither to Φ_n nor to $\mathbf{0}$, leads to a contradiction. This completes the proof. □

The following proposition states that the property of closed instantiations of sound conditional equations mentioned in the above lemma is preserved under equational derivations from a finite collection of sound equations.

Proposition 1. *Let E be a finite collection of conditional equations that is sound modulo $\underline{\leftrightarrow}$. Let $n \geq 2$ be larger than the size of any term in the equations of E. Assume, furthermore, that*

- *$E \vdash p \approx q$; and*
- *the summands of p are all bisimilar to Φ_n or $\mathbf{0}$.*

Then the summands of q are all bisimilar to Φ_n or $\mathbf{0}$.

Proof. By induction on the depth of the closed proof of the equation $p \approx q$ from E, using Lemma 1. □

Theorem 3. *Let $Act = \{a_i, b_i \mid i \geq 1\} \cup \{c\}$, where $a_i < b_i < c$ for each $i \geq 1$, and these are the only inequalities. Then bisimulation equivalence has no ground-complete axiomatization over* BCCSP$_\Theta$ *consisting of a finite set of sound conditional equations.*

Proof. Let E be a finite collection of conditional equations that is sound modulo $\underline{\leftrightarrow}$. Let $n \geq 2$ be larger than the size of any term in the equations of E. According to Proposition 1, from E we cannot derive $\Theta(\Phi_n) \approx \Phi_n$. This equation is sound modulo $\underline{\leftrightarrow}$, and therefore E is not ground-complete. □

5.2 A Positive Result

In the previous section, we offered an example of a priority structure $(Act, <)$ with respect to which it is impossible to give a finite, ground-complete axiomatization of bisimulation equivalence over BCCSP_Θ in terms of conditional equations without auxiliary operators. That result, however, does not imply that auxiliary operators are always necessary to achieve a finite basis of conditional equations for bisimulation equivalence. Our aim in this section is to substantiate this claim by providing some general conditions over the priority structure $(Act, <)$ that are sufficient to guarantee the existence of a finite, ground-complete conditional axiomatization of bisimulation equivalence over BCCSP_Θ.

Definition 2. *An anti-chain in a poset $(Act, <)$ is a subset of Act consisting of pairwise incomparable actions. The* width *of a poset $(Act, <)$ is the least upper bound of the cardinalities of its anti-chains.*

We now offer a countably infinite, ground-complete, conditional axiomatization of bisimulation equivalence over BCCSP_Θ. Such an axiomatization reduces to a finite one if the poset of actions has finite width.

Theorem 4. *Let $(Act, <)$ be an infinite poset of actions.*

1. *The axiom system consisting of (CPR2), (CPR3), $(CPR4)_n$ $(n \geq 0)$ and (A1)–(A4) is ground-complete for bisimilarity over BCCSP_Θ.*
2. *Assume that the width of $(Act, <)$ is k. Then the axiom system consisting of (CPR2), (CPR3), $(CPR4)_k$, (A1)–(A4) and (PR1) is ground-complete for bisimilarity over BCCSP_Θ.*

The sufficient condition over $(Act, <)$ stated in the above theorem applies, for instance, to any poset that has infinitely many finite anti-chains of bounded size. For example, it can be used to show that bisimilarity affords a finite, ground-complete axiomatization consisting of conditional equations over BCCSP_Θ if, for some k, the poset $(Act, <)$ has elements a_{ij} $(i \geq 1, 1 \leq j \leq k)$ ordered thus: $a_{hk} < a_{ij}$ if, and only if, $h < k$. That poset has countably many finite, maximal anti-chains of size k.

A more general sufficient condition over $(Act, <)$ that applies to some posets containing infinite anti-chains, and still guarantees the existence of a finite conditional basis of equations for bisimilarity over BCCSP_Θ may be found in the full version of the paper [1, Section 5.2]. That condition applies, for instance, to the flat priority structure $(\{\bot, a_0, a_1, \ldots\}, <)$, where the only ordering relations are given by $\bot < a_i$ for each $i \geq 0$. Membership of the countably infinite anti-chain $\{a_0, a_1, \ldots\}$ can be characterized syntactically by the predicate $P(\alpha) = \forall \beta.\ \neg(\alpha < \beta)$. We can therefore write the following, sound conditional equation that allows us to reduce the number of summands within the scope of a Θ operator:

$$P(\alpha) \wedge P(\beta) \Rightarrow \Theta(\alpha.x + \beta.y + z) \ \approx \ \Theta(\alpha.x + z) + \Theta(\beta.y + z) \ .$$

The generalization of Theorem 4(2) in the full version of this paper relies on the isolation of conditions on the priority structure that ensure the soundness of the above conditional equation over infinite, maximal anti-chains.

References

1. L. Aceto, T. Chen, W. Fokkink, and A. Ingolfsdottir. On the axiomatizability of priority. BRICS Report Series RS-06-1, BRICS, 2006.
2. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. CCS with Hennessy's merge has no finite equational axiomatization. *Theoretical Computer Science*, 330(3):377–405, 2005.
3. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. Finite equational bases in process algebra: Results and open questions. In *Processes, Terms and Cycles: Steps on the Road to Infinity*, LNCS 3838, pp. 338–367. Springer, 2005.
4. L. Aceto, W. Fokkink, A. Ingolfsdottir, and S. Nain. Bisimilarity is not finitely based over BPA with interrupt. In *Proc. CALCO'05*, LNCS 3629, pp. 52–66. Springer, 2005.
5. J. Baeten, J. Bergstra, and J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundam. Informat.*, IX(2):127–168, 1986.
6. J. Bergstra. *Put and Get, Primitives for Synchronous Unreliable Message Passing.* Logic Group Preprint Series 3, Utrecht University, Department of Philosophy, 1985.
7. J. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information & Control*, 60(1/3):109–137, 1984.
8. B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42(1):232–268, 1995.
9. E. Brinksma. A tutorial on LOTOS. In *Proc. PSTV'85*, pp. 171–194. North-Holland, 1985.
10. J. Camilleri and G. Winskel. CCS with priority choice. *Information and Computation*, 116(1):26–37, 1995.
11. R. Cleaveland and M. Hennessy. Priorities in process algebras. *Information and Computation*, 87(1-2):58–77, 1990.
12. R. Cleaveland, G. Lüttgen, and V. Natarajan. Priorities in process algebra. In *Handbook of Process Algebra*, pp. 711–765. Elsevier, 2001.
13. R. Cleaveland, G. Lüttgen, V. Natarajan, and S. Sims. Priorities for modeling and verifying distributed systems. In *Proc. TACAS'96,* LNCS 1055, pp. 278–297. Springer, 1996.
14. R. van Glabbeek. The linear time-branching time spectrum. In *Proc. CONCUR'90*, LNCS 458, pp. 278–297. Springer, 1990.
15. R. van Glabbeek. The linear time-branching time spectrum I. The semantics of concrete, sequential processes. In *Handbook of Process Algebra*, pp. 3–99. Elsevier, 2001.
16. J.F. Groote. A new strategy for proving ω-completeness with applications in process algebra. In *Proc. CONCUR'90*, LNCS 458, pp. 314–331. Springer, 1990.
17. S. Mauw, PSF – A Process Specification Formalism, PhD thesis, University of Amsterdam, 1991.
18. R. Milner. *Communication and Concurrency.* Prentice-Hall, 1989.
19. R. Milner, M. Tofte, R. Harper, and D. MacQueen. *The Definition of Standard ML (Revised).* MIT Press, 1997.
20. F. Moller. The importance of the left merge operator in process algebras. In *Proc. ICALP'90*, LNCS 443, pp. 752–764. Springer, 1990.
21. F. Moller. The nonexistence of finite axiomatisations for CCS congruences. In *Proc. LICS'90*, pp. 142–153. IEEE, 1990.
22. P. Sewell. Nonaxiomatisability of equivalences over finite state processes. *Annals of Pure and Applied Logic*, 90(1–3), 163–191, 1997.