A New Strategy for Proving ω-Completeness applied to Process Algebra

Jan Friso Groote

Centre for Mathematics and Computer Science

P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Email jfg@cwi.nl

Abstract

A new technique for proving ω-completeness based on proof transformations is presented. This technique is applied to axiom systems for finite, concrete, sequential processes. It turns out that the number of actions is important for these sets to be ω-complete. For the axiom systems for bisimulation and completed trace semantics one action suffices and for traces 2 actions are enough. The ready, failure, ready trace and failure trace axioms are only ω-complete if an infinite number of actions is available. We also consider process algebra with parallelism and show several axiom sets containing the axioms of standard concurrency ω-complete.

1 Introduction

An equational theory E over a signature Σ is called ω-complete iff for all open terms t_1, t_2:

$$\text{for all closed substitutions } \sigma: \; E \vdash \sigma(t_1) = \sigma(t_2) \quad \Leftrightarrow \quad E \vdash t_1 = t_2.$$

Not all equational theories are ω-complete: a well known example is the commutativity of the $+$ in Peano arithmetic. Another example is the three-element groupoid of MURSKIĬ [13], who showed that for an ω-complete specification of the groupoid an infinite number of equations is necessary.

Also in process algebra several theories are not ω-complete, and up till now this was more or less ignored (exceptions are MILNER [11] and MOLLER [12]). But there are several reasons why ω-completeness should not be neglected. In the first place equations between open terms play an important role in process algebra. For instance, processes are often described with sets of (open) equations. A complete set of axioms (not necessarily ω-complete) gives no guarantee that such sets of equations can be dealt with in a satisfactory manner. An example of this situation are the so-called 'axioms of standard concurrency' [2] in ACP, which had to be introduced in addition to the 'complete' set of axioms in order to prove the expansion theorem [3]. The status of these axioms became clear only

[*]The author is supported by the European Communities under RACE project no. 1046, Specification and Programming Environment for Communication Software (SPECS). This article also appears as report CS-R90XX, Centrum voor Wiskunde en Informatica, Amsterdam

after MOLLER [12] showed that in CCS with interleaving, but without communication, some of the axioms of standard concurrency are required for ω-completeness.

Furthermore, ω-completeness is also useful for theorem provers [8, 9, 15]. In [14] the so-called method 'proof by consistency' is introduced which can be applied to show inductive theorems equationally provable if ω-completeness of the axioms has been shown. In HEERING [6] it is argued that ω-completeness is desirable for the partial evaluation of programs. If $P(x, y)$ is a program with parameters x and y, and x has fixed value c, then the program $P_c(y)$ $(=P(c, y))$ should be evaluated as far as possible. In general this can only be achieved if the evaluation rules are ω-complete.

A more or less standard technique for proving ω-completeness is the following: given a set of axioms E over a signature Σ, find 'normal forms' and show that every open term is provably equal to a normal form. Then prove that for all pairs of different normal forms, closed instantiations can be found that differ in a model $\mathcal{M}$ for E. E does not necessarily have to be complete with respect to $\mathcal{M}$. This last step shows that the equivalence of these instantiations cannot be derived from E. From this ω-completeness of E follows directly. We prove the ω-completeness of the trace and completed trace axioms in this way. This technique has some disadvantages. The proofs are in general quite long and it is often difficult to find a suitable normal form.

In this paper we present an alternative technique that employs transformations of proofs. It is explained in section 3. With this method proofs of ω-completeness turn out to be shorter and for the major part straightforward. Moreover, no reference to a model is necessary. Unfortunately, this new technique cannot always be used. We apply our method to five sets of axioms, which are taken from [4], for finite, concrete, sequential processes. Among the proofs we give there is an ω-completeness proof of bisimulation semantics of which an earlier and longer version is given in [12]. It turns out that the number of actions is important for the axiom sets to be ω-complete. We need an infinite number of actions for the ready trace, failure trace, ready and failure axioms. For the bisimulation and the completed trace axioms at least one action is required whereas for the trace axioms two actions are necessary. Then we study axiom sets for finite, concrete process algebra with interleaving without communication (also done in [12]) and interleaving with communication. We give straightforward proofs of the ω-completeness of these sets.

Acknowledgements. I thank Rob van Glabbeek for several fruitful discussions and Alban Ponse for his detailed and constructive comments.

2 Preliminaries

Throughout this text we assume the existence of a countably infinite set V of variables with typical elements x, y, z. A (one sorted) *signature* $\Sigma = (F, rank)$ consists of a set of *function names* F, disjunct with V, and a rank function $rank : F \rightarrow \mathbb{N}$, denoting the arity of each function name in F. $T(\Sigma)$ is the set of *closed terms* over signature Σ and $\mathbb{T}(\Sigma)$ is the set of *open terms* terms over Σ and V. We use the symbol $\equiv$ for syntactic equality between terms. Furthermore, we have *substitutions* $\sigma, \rho : V \rightarrow \mathbb{T}(\Sigma)$ mapping variables to terms. Substitutions are in the standard way extended to functions from terms to terms. An expression of the form $t = u$ $(t, u \in \mathbb{T}(\Sigma))$ is called an *equation* over

$x = x$ (reflexivity)	$\frac{x = y}{y = x}$ (symmetry)	$\frac{x = y \quad y = z}{x = z}$ (transitivity)
$\frac{x_i = y_i \quad 1 \leq i \leq rank(f)}{f(x_1, ..., x_{rank(f)}) = f(y_1, ..., y_{rank(f)})}$ for all $f \in F$ (congruence)		

Table 1: The inference rules of equational logic

Σ. The letter e is used to range over equations. An expression of the form

$$\frac{e_1, ..., e_n}{e}$$

is called an *inference rule*. We call $e_1, ..., e_n$ the *premises* and e the *conclusion* of the inference rule. Substitutions are extended to equations and inference rules as expected.

An *equational theory* over a signature Σ is a set E containing equations over Σ. These equations are called *axioms*. An equation e can be *proved* from a theory E, notation $E \vdash e$, if e is an instantiation of an axiom in E or if e is the conclusion of an instantiation of an inference rule r in table 1 of which all (instantiated) premises can be proved. If it is clear from the context what E is, we sometimes write only e instead of $E \vdash e$. We write $E_1 \vdash E_2$ if $E_1 \vdash e$ for all $e \in E_2$. Note that if $E \vdash t = u$ for $t, u \in T(\Sigma)$, then $t = u$ can be proved using closed instantiated axioms and inference rules only.

An equational theory E is *ω-complete* if for all equations e: $E \vdash e$ iff $E \vdash \sigma(e)$ for all substitutions $\sigma : V \rightarrow T(\Sigma)$. Note that the implication from left to right is trivial. So, in general we only prove the implication from the right-hand side to the left-hand side.

3 The general proof strategy

Let $\Sigma = (F, rank)$ be a signature and let E be an equational theory over Σ. We present a technique to show that E is ω-complete. Assume $t = t'$ is an equation between open terms that can be proved for all its closed instantiations by the axioms of E. We transform $t = t'$ to a closed equation by a substitution $\rho : V \rightarrow T(\Sigma)$ that maps each variable in t and t' to a unique closed (sub)term representing this variable. By assumption $E \vdash \rho(t) = \rho(t')$. We transform the proof of this fact to a proof for $E \vdash t = t'$ by a translation R which replaces each subterm representing a variable by the variable itself. This transformation yields the desired proof if requirements (1), (2) and (3) below are satisfied. (1) says that the translation of $\rho(t) = \rho(t')$ must yield $t = t'$ (or something provably equivalent). In general this only works properly if each subterm representing a variable is unique for that variable and cannot be confused with other subterms. Requirements (2) and (3) guarantee that the transformed proof is indeed a proof. This is most clearly stated in equation (5), which is a consequence of (2) and (3).

- For $u \equiv t$ or $u \equiv t'$:

$$E \vdash R(\rho(u)) = u. \tag{1}$$

- For each $f \in F$ with $rank(f) > 0$ and $u_1, ..., u_{rank(f)}, u'_1, ..., u'_{rank(f)} \in T(\Sigma)$:

$$E \cup \{u_i = u'_i,\ R(u_i) = R(u'_i) | 1 \leq i \leq rank(f)\} \vdash \qquad (2)$$
$$R(f(u_1, ..., u_{rank(f)})) = R(f(u'_1, ..., u'_{rank(f)})).$$

- For each axiom $e \in E$ and closed substitution $\sigma : V \to T(\Sigma)$:

$$E \vdash R(\sigma(e)). \qquad (3)$$

Theorem 3.1. *Let E be an equational theory over signature Σ. If for each pair of terms $t, t' \in \mathbb{T}(\Sigma)$ that are provably equal for all closed instantiations, there exist a substitution $\rho : V \to T(\Sigma)$ and a mapping $R : T(\Sigma) \to \mathbb{T}(\Sigma)$ satisfying (1),(2) and (3), then E is ω-complete.*

Proof. Let $t, t' \in \mathbb{T}(\Sigma)$ such that for each substitution $\sigma : V \to T(\Sigma)$:

$$E \vdash \sigma(t) = \sigma(t'). \qquad (4)$$

We must prove that $E \vdash t = t'$. This is an immediate corollary of the following statement:

$$E \vdash u = u' \text{ for } u, u' \in T(\Sigma) \quad \Rightarrow \quad E \vdash R(u) = R(u'). \qquad (5)$$

It follows from (4) that $E \vdash \rho(t) = \rho(t')$. Using (5) this implies $E \vdash R(\rho(t)) = R(\rho(t'))$. By (1) it follows that $E \vdash t = t'$.

Statement (5) is shown by induction on the proof of $E \vdash u = u'$. As u and u' are closed terms, we may assume that the whole proof of $E \vdash u = u'$ consists of closed terms. First we consider the inference rules without premises. There are two possibilities. In the first case $u = u'$ has been shown by the inference rule $x = x$, i.e. $u \equiv \sigma(x) \equiv u'$ for some substitution $\sigma : V \to T(\Sigma)$. Clearly, $E \vdash R(u) = R(u')$ using the same inference rule and a substitution $\sigma' : V \to \mathbb{T}(\Sigma)$ defined by $\sigma'(x) = R(\sigma(x))$. Otherwise, $u = u'$ is an instantiation $\sigma(e)$ of an axiom $e \in E$. Using (3) it follows immediately that $E \vdash R(\sigma(e))$.

We check here the inference rules with premises. First we deal with the rule for transitivity. So assume $E \vdash u = u'$ has been proved using $E \vdash u = u''$ and $E \vdash u'' = u'$. By induction we know that there are proofs for $E \vdash R(u) = R(u'')$ and $E \vdash R(u'') = R(u')$. Applying the inference rule for transitivity again we have that $E \vdash R(u) = R(u')$. The rule for symmetry can be dealt with in the same way. Now suppose that $E \vdash f(u_1, ..., u_{rank(f)}) = f(u'_1, ..., u'_{rank(f)})$ has been proved using $E \vdash u_i = u'_i$ $(1 \leq i \leq rank(f))$. By induction we know that $E \vdash R(u_i) = R(u'_i)$. Using (2), it follows immediately that $E \vdash R(f(u_1, ..., u_{rank(f)})) = R(f(u'_1, ..., u'_{rank(f)}))$. □

This new proof strategy cannot always be applied. This is illustrated by the following example.

Example 3.2. Suppose we have an axiomatization for the natural numbers with a function *max* giving the maximum of any pair of numbers. In the signature we have a 0, a successor function S and *max*. The following set E_{max} of axioms is easily seen to be complete with respect to the standard interpretation.

$$max(x, 0) = x,$$
$$max(0, x) = x,$$
$$max(S(x), S(y)) = S(max(x, y)).$$

Clearly, E_{max} is not ω-complete as for instance the general associativity and commutativity of max is not derivable although each closed instance of them is.

It is impossible to use our technique to prove any extension of E_{max} ω-complete. This can be seen by considering the following two terms:

$$t_1 = max(S(0), x) \text{ and}$$
$$t_2 = x.$$

We can see that these terms are not provably equal because with $x = 0$, the first term is equal to $S(0)$ and the second is equal to 0. Note that this is the only way to see the difference. If any term that is not equal to 0 is substituted for x then both terms are equivalent.

Suppose we would like to apply our technique in this case. If we define ρ such that $\rho(x) = 0$ then we must define the translation R such that $R(0) = x$. But then $R(\rho(t_1)) = max(S(x), x)$ which cannot be shown equal to $max(S(0), x)$. If ρ would be chosen such that $\rho(x) \neq 0$ and R could be defined such that $E_{max} \vdash R(t_i) = t_i$ $(i = 1, 2)$ then equation (5), which follows from (2) and (3), cannot hold because it implies that $E_{max} \vdash t_1 = t_2$.

So, this example shows that the new technique is not generally applicable, but as will be shown in the next sections, there are enough cases where the application of this technique leads to attractive proofs.

4 Applications in finite, concrete, sequential process algebra

In the remainder of this paper we apply our technique to prove completeness of several axiom systems. In this section sets given for BCCSP in [4] are studied. BCCSP is a basic CCS and CSP-like language for finite, concrete, sequential processes. It is parameterized by a set *Act* of actions representing the elementary activities that can be performed by processes. We write $|Act|$ for the number of elements in *Act* ($|Act| = \infty$ if *Act* has an infinite number of elements). The language BCCSP contains a constant δ, which is comparable to **0** or NIL in CCS and to STOP in CSP. We call δ *inaction* or sometimes *deadlock*. There is an *alternative composition* operator $+$ with its usual meaning and, furthermore, there is an *action prefix* operator $a :$ for each action a in *Act*.

In the sequel we will often use sums of arbitrary finite size. It is convenient to have a notation for these. Therefore we introduce the abbreviation:

$$\sum_{i \in I} t_i = t_{i_1} + \dots + t_{i_n}$$

where $I = \{i_1, \dots, i_n\}$ is a finite index set and $t_i \in \mathbb{T}(\text{BCCSP})$ $(i \in I)$. We take $\sum_{i \in \emptyset} t_i = \delta$. Note that this notation is only justified if $+$ is commutative, associative. We only use this notation when this is the case.

The depth $|t|$ of a term $t \in \mathbb{T}(\text{BCCSP})$ is inductively defined as follows:

$$|\delta| = 0, \qquad |x| = 0 \text{ for all } x \in V,$$
$$|a : t| = 1 + |t| \text{ for all } a \in Act, \qquad |t_1 + t_2| = \max(|t_1|, |t_2|).$$

In table 2 we find several axiom systems corresponding to several semantics given in [4]. We will investigate the ω-completeness of these sets. On the top line of this table we

	B	RT	FT	R	F	CT	T
$x+y=y+x$	+	+	+	+	+	+	+
$(x+y)+z=x+(y+z)$	+	+	+	+	+	+	+
$x+x=x$	+	+	+	+	+	+	+
$x+\delta=x$	+	+	+	+	+	+	+
(see (6) in text)		+	+	v	v	v	v
$a:x+a:y=a:x+a:y+a:(x+y)$			+		v	v	v
$a(b:x+u)+a:(b:y+v)=$ $a:(b:x+b:y+u)+a:(b:x+b:y+v)$				+	+	v	v
$a:x+a:(y+z)=a:x+a:(x+y)+a:(y+z)$					+	ω	v
$a:(b:x+u)+a:(c:y+v)=a:(b:x+c:y+u+v)$						+	v
$a:x+a:y=a:(x+y)$							+

Table 2: Axioms for several process algebra semantics

find their abbreviations: B stands for *Bisimulation*, RT for *Ready Trace*, FT for *Failure Trace*, R for *Ready* and F for *Failure semantics*, CT for *Completed Traces* and finally T represents *Trace* semantics. The axioms that are necessary for ready trace semantics (besides the axioms for bisimulation) are given by the following scheme:

$$a:(\sum_{i\in I} a_i:x_i+y)+a:(\sum_{i\in J} a_i:x_i+y)=a:(\sum_{i\in I\cup J} a_i:x_i+y) \tag{6}$$

where $\{a_i|i\in I\}=\{a_i|i\in J\}$, and $x_i,y\in V$ $(i\in I\cup J)$. This scheme differs from the axiomatization given in [4], where an additional function name I and a conditional axiom were used to axiomatize ready trace semantics. We do not want to introduce these concepts here. Both axiomatizations prove exactly the same open equations.

Let X stand for any of the semantics B,RT,... The symbol 'v' in a column of semantics X indicates that an axiom is derivable from the other axioms valid for X. The symbol '+' means that the axiom is required for a complete axiomatization of the models given in [4] and 'ω' means that the axiom is only necessary for an ω-complete axiomatization. It follows immediately that:

$$\begin{tikzcd}
 & & R \arrow[dr] & & & \\
B \arrow[r] & RT \arrow[ur] \arrow[dr] & & F \arrow[r] & CT \arrow[r] & T \\
 & & FT \arrow[ur] & & &
\end{tikzcd}$$

where the semantics to the left are finer than the semantics to the right. The semantics FT and R are incomparable [4]. The abbreviation for a semantics will also be used to denote the set of axioms necessary for its ω-complete axiomatization.

Lemma 4.1. *Let $t,u\in\mathbb{T}(BCCSP)$. If $T\vdash t=u$, then $|t|=|u|$.*
Proof. Direct with induction on the proof of $t=u$. □

As T $\vdash$ B, T $\vdash$ RT etc. it immediately follows from the last lemma that 'X $\vdash t = u \Rightarrow |t| = |u|$', where X is any of the sets B, RT, etc.

4.1 The semantics B

We start considering the axioms for bisimulation semantics. If *Act* contains at least one element, then B is ω-complete. This fact has already been shown in [12] where a traditional technique was used. Note that it makes no sense to investigate the situation where $Act = \emptyset$, because in that case all closed terms will have the form δ, $\delta+\delta$, $\delta+\delta+\ldots$ and therefore they are equal and we only require the axiom $x = y$ for an ω-complete axiomatization.

Theorem 4.1.1. *If $|Act| \geq 1$ then the axiom system B is ω-complete.*
Proof. As $|Act| \geq 1$, *Act* contains at least one action a. This action will play an important role in this proof. We follow the lines set out in theorem 3.1. So, assume we have two terms $t, t' \in \mathbb{T}(\mathrm{BCCSP})$. Select a natural number $m > \max(|t|, |t'|)$ and define $\rho : V \to T(\mathrm{BCCSP})$ by:

$$\rho(x) = a^{n(x)\cdot m} : \delta$$

where $a^k : \delta$ is an abbreviation of k applications of $a :$ to δ and $n : V \to \mathbb{N}\setminus\{0\}$ is a function assigning a unique natural number to each variable in x. Define $R : T(\mathrm{BCCSP}) \to \mathbb{T}(\mathrm{BCCSP})$ as follows:

$$\begin{aligned}
&R(\delta) = \delta,\\
&R(t+u) = R(t) + R(u),\\
&R(b:t) = b : R(t) \text{ if } b \neq a \text{ or } |b:t| \neq m\cdot n(x) \text{ for all } x \in V,\\
&R(a:t) = x \text{ if } |a:t| = m\cdot n(x) \text{ for some } x \in V.
\end{aligned}$$

We will now check conditions (1), (2) and (3) of theorem 3.1. We prove (1) with induction on a term $u \in \mathbb{T}(\mathrm{BCCSP})$ provided $|u| < m$. Note that this is sufficient as $|t| < m$ and $|t'| < m$.

$$\begin{aligned}
&R(\rho(\delta)) = \delta,\\
&R(\rho(x)) = R(a^{n(x)\cdot m} : \delta) = x,\\
&R(\rho(u_1+u_2)) = R(\rho(u_1)) + R(\rho(u_2)) = u_1 + u_2,\\
&R(\rho(b:u)) = b : R(\rho(u)) = b : u \text{ if } b \neq a,\\
&R(\rho(a:u)) = R(a:\rho(u)) =^* a : R(\rho(u)) = a : u.
\end{aligned}$$

$=^*$ follows directly from the observation that $|a:\rho(u)| \neq m\cdot n(x)$ for all $x \in V$. In order to see this, first note that $1 \leq |a:u| < m$. If u does not contain variables, it is clear that $1 \leq |a:\rho(u)| < m$ and hence, $|a:\rho(u)| \neq m\cdot n(x)$. So, suppose u contains variables. By applying ρ to u each variable x is replaced by $a^{n(x)\cdot m} : \delta$. So $|a:\rho(u)| = p + n(x)\cdot m$ where x is a variable in u such that there is no other variable y in u with $n(y) > n(x)$ and p $(1 \leq p < m)$ is the 'depth' of the deepest occurrence of x in u. As $1 \leq p < m$, $|a:\rho(u)| \neq n(x)\cdot m$ for each $x \in V$.

Now we check (2). Assume B $\vdash u_i = u_i'$ and B $\vdash R(u_i) = R(u_i')$ for $u_i, u_i' \in T(\mathrm{BCCSP})$ and $i = 1, 2$. We find that:

$$\begin{aligned}
&\mathrm{B} \vdash R(u_1+u_2) = R(u_1) + R(u_2) = R(u_1') + R(u_2') = R(u_1'+u_2').\\
&\mathrm{B} \vdash R(b:u_1) = b : R(u_1) = b : R(u_1') = R(b:u_1') \text{ if } b \neq a.
\end{aligned}$$

$B \vdash R(a : u_1) =^* a : R(u_1) = a : R(u_1') =^+ R(a : u_1')$
if $|a : u_1| \neq m \cdot n(x)$ for all $x \in V$.

$=^*$ follows directly from the condition. As $B \vdash u_1 = u_1'$ it follows that $|a : u_1| = |a : u_1'|$ (cf. lemma 4.1) and hence, $|a : u_1'| \neq m \cdot n(x)$ for all $x \in V$. This justifies $=^+$.

$B \vdash R(a : u_1) = x =^* R(a : u_1')$ if $|a : u_1| = m \cdot n(x)$ for some $x \in V$.

It follows that $|a : u_1'| = m \cdot n(x)$ explaining $=^*$.

Finally, we must check (3). This is trivial as the axioms do not contain actions. We only check the axiom $x + y = y + x$. The other axioms can be dealt with in the same way. Let $\sigma : V \to T(\text{BCCSP})$ be a substitution, then:

$$B \vdash R(\sigma(x + y)) = R(\sigma(x)) + R(\sigma(y)) = R(\sigma(y)) + R(\sigma(x)) = R(\sigma(y + x)).$$

□

4.2 The semantics RT,FT,R and F

We will show that the sets of axioms RT,FT,R and F are all ω-complete in case *Act* is infinite. If *Act* is finite, we have the following identity:

$$a : \sum_{i \in J} a_i : \delta + a : (x + \sum_{i \in J} a_i : \delta) = a : (x + \sum_{i \in J} a_i : \delta) \tag{7}$$

where $\{a_i | i \in J\} = Act$. Each closed instance of this identity is derivable from the axioms of RT,FT,R or F. However, (7) is not derivable in its general form: if (7) were derivable, then it would also hold if *Act* would be extended by a 'fresh' action $b \notin \{a_i | i \in J\}$. Define a substitution σ satisfying $\sigma(x) = b : \delta$. Applying σ to (7) yields:

$$a : \sum_{i \in J} a_i : \delta + a : (b : \delta + \sum_{i \in J} a_i : \delta) = a : (b : \delta + \sum_{i \in J} a_i : \delta).$$

but this equation does not hold in the failure model [4]. Hence, it is not derivable from F and therefore it can certainly not be derived from RT,FT or R.

So, in order to prove RT,FT,R and F ω-complete, *Act* must at least be countably infinite. The following theorem shows that this condition is also sufficient.

Theorem 4.2.1. *If $|Act|$ is infinite, then the axiom sets RT,FT,R and F are ω-complete.*
Proof. Take two terms t, t'. Define a substitution $\rho : V \to T(\text{BCCSP})$ by:

$$\rho(x) = a_x : \delta$$

where a_x is a unique action for each $x \in V$ and a_x must not occur in either t or t'. Note that these actions can always be found as $|Act| = \infty$. Define $R : T(\text{BCCSP}) \to \mathbb{T}(\text{BCCSP})$ as follows:

$R(\delta) = \delta$,
$R(a : u) = a : R(u)$ if $a \neq a_x$ for each $x \in V$,
$R(a_x : u) = x$,
$R(u_1 + u_2) = R(u_1) + R(u_2)$.

Condition (1) of theorem 3.1 can be checked by induction on the structure of open terms not containing action prefix operators a_x :.

$$\begin{aligned}
&R(\rho(\delta)) = \delta,\\
&R(\rho(x)) = R(a_x : \delta) = x,\\
&R(\rho(a : u)) = R(a : \rho(u)) = a : R(\rho(u)) = a : u \text{ as } a \neq a_x \text{ for each } x \in V,\\
&R(\rho(u_1 + u_2)) = R(\rho(u_1)) + R(\rho(u_2)) = u_1 + u_2.
\end{aligned}$$

Condition (2) can be checked in the same straightforward manner. Suppose $\mathrm{X} \vdash R(u_i) = R(u_i')$ for $u_i, u_i' \in T(\mathrm{BCCSP})$ and $i = 1, 2$. X may be replaced by either RT,FT,R or F. Then:

$$\begin{aligned}
&\mathrm{X} \vdash R(a : u_1) = a : R(u_1) = a : R(u_1') = R(a : u_1') \text{ if } a \neq a_x \text{ for each } x \in V.\\
&\mathrm{X} \vdash R(a_x : u_1) = x = R(a_x : u_1').\\
&\mathrm{X} \vdash R(u_1 + u_2) = R(u_1) + R(u_2) = R(u_1') + R(u_2') = R(u_1' + u_2').
\end{aligned}$$

Finally, we check (3). We restrict ourselves to the ready trace axiom scheme. All other axioms can be dealt with in the same way. First we assume that $a = a_x$. Let $\sigma : V \to T(\mathrm{BCCSP})$ be a substitution. Then RT $\vdash$:

$$\begin{aligned}
&R(a_x : (\sum_{i\in I} a_i : \sigma(x_i) + \sigma(y)) + a_x : (\sum_{i\in J} a_i : \sigma(x_i) + \sigma(y))) =\\
&x + x = x =\\
&R(a_x : (\sum_{i\in I\cup J} a_i : \sigma(x_i) + \sigma(y))).
\end{aligned}$$

In case $a \neq a_x$ for each $x \in V$, we have that RT proves:

$$\begin{aligned}
&R(a : (\sum_{i\in I} a_i : \sigma(x_i) + \sigma(y)) + a : (\sum_{i\in J} a_i : \sigma(x_i) + \sigma(y))) =\\
&a : (\sum_{i\in I} R(a_i : \sigma(x_i)) + R(\sigma(y))) + a : (\sum_{i\in J} R(a_i : \sigma(x_i)) + R(\sigma(y))) =\\
&a : (\sum_{i\in I\setminus\{i\in I | a_i = a_x\}} a_i : R(\sigma(x_i)) + \sum_{x\in\{x | a_x = a_i \wedge i\in I\}} x + R(\sigma(y)))+\\
&a : (\sum_{i\in J\setminus\{i\in J | a_i = a_x\}} a_i : R(\sigma(x_i)) + \sum_{x\in\{x | a_x = a_i \wedge i\in J\}} x + R(\sigma(y))) =^*\\
&a : (\sum_{i\in (I\cup J)\setminus\{i\in I\cup J | a_i = a_x\}} a_i : R(\sigma(x_i)) + \sum_{x\in\{x | a_x = a_i \wedge i\in J\}} x + R(\sigma(y))) =\\
&R(a : (\sum_{i\in I\cup J} a_i : \sigma(x_i) + \sigma(y))).
\end{aligned}$$

$=^*$ follows from the observations that $\{a_i | i \in I,\ a_i \neq a_x \text{ for some } x \in V\} = \{a_i | i \in J,\ a_i \neq a_x \text{ for some } x \in V\}$ and $\{x | a_x = a_i \wedge i \in I\} = \{x | a_x = a_i \wedge i \in J\}$ which follow directly from the fact that $\{a_i | i \in I\} = \{a_i | i \in J\}$. □

4.3 The completed trace axioms

We now show the ω-completeness for the axiom set CT. However, it is not possible to use the technique presented in the beginning. This will be shown in example 4.3.4. Therefore, we will use a more traditional technique. Hence, it is necessary to explicitly define the completed trace semantics for BCCSP. In CT the meaning of a process is its set of traces that end in inaction.

Definition 4.3.1. The interpretation $[\![.]\!]_{\mathrm{CT}} : T(\mathrm{BCCSP}) \to 2^{Act^*}$ (the set of subsets of strings over Act) is defined as follows:

$$\begin{aligned}
&[\![\delta]\!]_{\mathrm{CT}} = \emptyset,\\
&[\![a : t]\!]_{\mathrm{CT}} = \{a \star s \mid s \in [\![t]\!]_{\mathrm{CT}}\} \cup \{a \mid [\![t]\!]_{\mathrm{CT}} = \emptyset\},\\
&[\![t_1 + t_2]\!]_{\mathrm{CT}} = [\![t_1]\!]_{\mathrm{CT}} \cup [\![t_2]\!]_{\mathrm{CT}}.
\end{aligned}$$

We say that $t_1, t_2 \in T(\mathrm{BCCSP})$ are *completed trace equivalent*, notation $t_1 =_{\mathrm{CT}} t_2$, iff $[\![t_1]\!]_{\mathrm{CT}} = [\![t_2]\!]_{\mathrm{CT}}$.

Lemma 4.3.2. *(Soundness) Let $t_1, t_2 \in T(BCCSP)$:*

$$\mathrm{CT} \vdash \mathrm{t}_1 = \mathrm{t}_2 \quad \Rightarrow \quad \mathrm{t}_1 =_{\mathrm{CT}} \mathrm{t}_2.$$

Proof. Straightforward using the definitions. □

For completed trace semantics the following theorem states the completeness of the axioms with respect to the given model. Moreover, as t_1 and t_2 may be open terms, ω-completeness is implied also.

Theorem 4.3.3. *If $|Act| \geq 1$ then for all $t_1, t_2 \in \mathbb{T}(BCCSP)$, we have that:*

$$\forall \sigma : V \to T(BCCSP)\ \sigma(t_1) =_{\mathrm{CT}} \sigma(t_2) \quad \Rightarrow \quad \mathrm{CT} \vdash \mathrm{t}_1 = \mathrm{t}_2.$$

Proof. We skip the long and tedious proof of this theorem in which we had to use the standard technique as shown by the next example. □

Example 4.3.4. Consider the following two BCCSP-terms.

$$\begin{aligned}
t_1 &= a : x + a : (a : \delta + x),\\
t_2 &= a : (a : \delta + x).
\end{aligned}$$

These two terms are clearly different in CT as for a substitution σ with $\sigma(x) = \delta$, $\sigma(t_1)$ has a completed trace a which is not available in $\sigma(t_2)$. For every substitution σ' with $\sigma'(x) \neq \delta$, $\sigma'(t_1) =_{CT} \sigma'(t_2)$. Hence, using the same arguments as in example 3.2, we cannot apply our new technique.

4.4 The trace axioms

Again we do not use the new technique as in this case the 'standard' technique is more convenient to use. We must give the trace semantics explicitly. In trace semantics each process is characterized by its set of prefix closed traces:

Definition 4.4.1. The interpretation $[\![.]\!]_{\mathrm{T}} : T(\mathrm{BCCSP}) \to 2^{Act^*}$ is defined as follows:

$$\begin{aligned}
&[\![\delta]\!]_{\mathrm{T}} = \emptyset,\\
&[\![a : t]\!]_{\mathrm{T}} = \{a \star \sigma \mid \sigma \in [\![t]\!]_{\mathrm{T}}\} \cup \{a\},\\
&[\![t_1 + t_2]\!]_{\mathrm{T}} = [\![t_1]\!]_{\mathrm{T}} \cup [\![t_2]\!]_{\mathrm{T}}.
\end{aligned}$$

We say that $t_1, t_2 \in T(\text{BCCSP})$ are *trace equivalent*, notation $t_1 =_{\mathrm{T}} t_2$, iff $[\![t_1]\!]_{\mathrm{T}} = [\![t_2]\!]_{\mathrm{T}}$.

Lemma 4.4.2. *(Soundness) Let $t_1, t_2 \in T(BCCSP)$:*

$$\mathrm{T} \vdash t_1 = t_2 \quad \Rightarrow \quad t_1 =_{\mathrm{T}} t_2.$$

Proof. Straightforward using the definitions. □

For trace semantics we need two actions in order to prove T ω-complete. If $|Act| = 1$ then the following axiom is valid:

$$x + a : x = a : x.$$

This can easily be seen by proving $\mathrm{T} \vdash t + a : t = a : t$ for all $t \in T(\text{BCCSP})$ with induction on t if $|Act| = 1$. The axiom $x + a : x = a : x$ is in general not derivable from T, because instantiating x with $b : \delta$ yields $b : \delta + a : b : \delta \neq_{\mathrm{T}} a : b : \delta$ where $a, b \in Act$ are two different actions. In the next theorem we show that if $|Act| \geq 2$ then the axiom set T is ω-complete. First we define the notion of a syntactic summand. This notion is only used in this section.

Definition 4.4.3. Let $t, u \in \mathbb{T}(\text{BCCSP})$. t is a *syntactic summand* of u, notation $t \sqsubseteq u$ if:

- $t \equiv a : t'$ and $u \equiv a : t'$ for some $t' \in \mathbb{T}(\text{BCCSP})$ or,
- $u \equiv u_1 + u_2$ and $t \sqsubseteq u_1$ or $t \sqsubseteq u_2$.

Lemma 4.4.4. *Let $t_1, t_2 \in \mathbb{T}(BCCSP)$. If for each syntactic summand $u \in \mathbb{T}(BCCSP)$,*

$$u \sqsubseteq t_1 \quad \Leftrightarrow \quad u \sqsubseteq t_2$$

then $B \vdash t_1 = t_2$.
Proof. Straightforward. □

Theorem 4.4.5. *If $|Act| \geq 2$ then for each $t_1, t_2 \in \mathbb{T}(BCCSP)$, we have that:*

$$\forall \sigma : V \to T(BCCSP) : \ \sigma(t_1) =_{\mathrm{T}} \sigma(t_2) \quad \Rightarrow \quad \mathrm{T} \vdash t_1 = t_2.$$

Proof. We use the abbreviation $a_1 \star \ldots \star a_n : t$ with $a_1 \star \ldots \star a_n \in Act^\star$ for $a_1 : \ldots : a_n : t$. For $s \in Act^\star$, we define $|s|$ to be $|s : \delta|$, i.e. the length of trace s. For traces $s_1, s_2 \in Act^\star$ we write $s_1 \leq s_2$ if for some $r \in Act^\star$, $s_1 \star r = s_2$ or $s_1 = s_2$. In this case s_1 is a *prefix* of s_2.

First we define a T-*normal form*, which plays a crucial role in this proof. A term $t \in \mathbb{T}(\text{BCCSP})$ is a T-normal form if

$$t \equiv \sum_{i \in I} s_i : \delta + \sum_{i \in J} s_i : x_i$$

with $s_i \in Act^\star$ $(i \in I \cup J)$, satisfying:

(1) for each s_j $(j \in I \cup J)$ with $|s_j| > 1$, there is a $i \in I$ such that $s_i \star a = s_j$ for some $a \in Act$.

(2) for each s_j $(j \in J)$ with $|s_j| > 0$, there is a $i \in I$ such that $s_j = s_i$.

Fact 1. *Let $t \in \mathbb{T}(BCCSP)$. Then there is a T-normal form t' such that:*

$$\mathrm{T} \vdash t = t'.$$

Proof of fact. Straightforward with induction on t. □

Fact 2. *Let t and t' be two T-normal forms such that for some u, $u \sqsubseteq t$, $u \not\sqsubseteq t'$ or vice versa. Then there is a substitution $\sigma : V \to T(BCCSP)$ such that:*

$$\sigma(t) \neq_{\mathrm{T}} \sigma(t').$$

Proof of fact. By symmetry it is sufficient to consider only the case where $u \sqsubseteq t$ and $u \not\sqsubseteq t'$. We can distinguish between:

(1) $u \equiv s : \delta$ with $s \in Act^\star$. Define $\sigma(x) = \delta$ for all $x \in V$. Note that $s \in [\![\sigma(t)]\!]_{\mathrm{T}}$. Moreover, in this case it holds that $s \in [\![\sigma(t')]\!]_{\mathrm{T}}$ iff $s : \delta \sqsubseteq t'$. Note that the conditions (1) and (2) are required to prove this. Hence, as $s : \delta \not\sqsubseteq t'$, $s \notin [\![\sigma(t')]\!]_{\mathrm{T}}$.

(2) $u \equiv s : x$ for some $x \in V$ and $s \in Act^\star$. Let m be a natural number such that $m > \max(|t|, |t'|)$. Define $\sigma(x) = a^m : b : \delta$ where $a, b \in Act$ are two different actions and $\sigma(y) = \delta$ if $y \not\equiv x$. Clearly, $s \star a^m \star b \in [\![\sigma(t)]\!]_{\mathrm{T}}$. We will show that $s \star a^m \star b \notin [\![\sigma(t')]\!]_{\mathrm{T}}$. Therefore we write $t' \equiv \sum_{i \in I} s_i : \delta + \sum_{i \in J} s_i : y_i$ in the following way:

$$\sum_{i \in I} s_i : \delta + \sum_{i \in K_1} s_i : y_i + \sum_{i \in K_2} s_i : x + \sum_{i \in K_3} s_i : x + \sum_{i \in K_4} s_i : x$$

where

$$\begin{aligned} K_1 &= \{i | i \in J \text{ and } y_i \not\equiv x\}, \\ K_2 &= \{i | i \in J,\ y_i \equiv x \text{ and } |s_i| < |s|\}, \\ K_3 &= \{i | i \in J,\ y_i \equiv x \text{ and } |s_i| = |s|\}, \\ K_4 &= \{i | i \in J,\ y_i \equiv x \text{ and } |s_i| > |s|\}. \end{aligned}$$

Note that $J = K_1 \cup K_2 \cup K_3 \cup K_4$. We will show that $s \star a^m \star b$ cannot originate from any of these components. We deal with all five cases separately:

(a) For any $r \in [\![\sum_{i \in I} s_i : \delta]\!]_{\mathrm{T}}$, $|r| < m$ and therefore $r \neq s \star a^m \star b$.

(b) For any $r \in [\![\sum_{i \in K_1} s_i : \sigma(y_i)]\!]_{\mathrm{T}}$, $|r| < m$ because $\sigma(y_i) = \delta$. Hence, $r \neq s \star a^m \star b$.

(c) For any $r \in [\![\sum_{i \in K_2} s_i : \sigma(x)]\!]_{\mathrm{T}}$, $|r| \leq |s_i| + m + 1 < |s| + m + 1 = |s \star a^m \star b|$. Hence, $r \neq s \star a^m \star b$.

(d) For any $r \in [\![\sum_{i\in K_3} s_i : \sigma(x)]\!]_{\mathrm{T}}$, $r \leq s_i \star a^m \star b$ for some $i \in K_3$. If $|r| < |s| + m + 1$, clearly, $r \not\simeq s \star a^m \star b$. If $|r| = |s| + m + 1$, then $r = s_i \star a^m \star b$. As $s : x \not\sqsubseteq t'$, $s_i \neq s$. Therefore $r \not\simeq s \star a^m \star b$.

(e) Let for some $r \in Act^\star$, $r[i]$ be the i^{th} symbol in r. For any $r \in [\![\sum_{i\in K_4} s_i : \sigma(x)]\!]_{\mathrm{T}}$, $r \leq s_i \star a^m \star b$ for some $i \in K_4$. If $|r| \leq |s| + m$, then clearly $r \not\simeq s \star a^m \star b$. If $|r| > |s| + m$, consider $r[|s| + m + 1]$. As $|s_i \star a^m \star b| > |s \star a^m \star b| > |s_i|$, $r[|s| + m + 1] = a$. But, $s \star a^m \star b[|s| + m + 1] = b$. Hence, if $|r| > |s| + m$, it also holds that $r \not\simeq s \star a^m \star b$.

This finishes the proof of the second fact. □

Using both facts it follows almost immediately that T is ω-complete with respect to $=_{\mathrm{T}}$. Suppose $t, t' \in \mathbb{T}(\text{BCCSP})$ such that for each substitution $\sigma : V \to T(\text{BCCSP})$, it holds that $\sigma(t) =_{\mathrm{T}} \sigma(t')$. Both t and t' are provably equal to T-normal forms u and u' (fact 1). If u and u' have different syntactic summands, then by the second fact $\rho(u) \neq_{\mathrm{T}} \rho(u')$ for some substitution $\rho : V \to T(\text{BCCSP})$. This is a contradiction. Hence, by lemma 4.4.4, $B \vdash u = u'$ and therefore:

$$\mathrm{T} \vdash t = u = u' = t'.$$

□

5 Extensions with the parallel operator

We extend the signature BCCSP with operators for parallelism.

5.1 Interleaving without communication

First, we study BCCSP with the merge and the leftmerge, but without communication. The resulting signature is called $\text{BCCSP}_{\mathbin{|\!\lfloor}}$. We will study $\text{BCCSP}_{\mathbin{|\!\lfloor}}$ in the setting of bisimulation where $|Act| = \infty$. The upper half of table 3 contains a complete set of axioms. The completeness follows immediately from the completeness of the axiom set B for BCCSP because any closed term over the signature $\text{BCCSP}_{\mathbin{|\!\lfloor}}$ can be rewritten to a term over the signature BCCSP.

In order to have an ω-complete set of axioms, we add two new axioms (see the lower squares of table 3). These axioms are derivable for all closed instances. Therefore they are valid in bisimulation semantics. The complete set of axioms in table 3 is called $\mathrm{B}_{\mathbin{|\!\lfloor}}$. The following theorem concerns the ω-completeness of $B_{\mathbin{|\!\lfloor}}$.

$x + y = y + x$	$x \parallel y = x \mathbin{|\!\lfloor} y + y \mathbin{|\!\lfloor} x$
$(x + y) + z = x + (y + z)$	$\delta \mathbin{|\!\lfloor} x = \delta$
$x + x = x$	$a : x \mathbin{|\!\lfloor} y = a : (x \parallel y)$
$x + \delta = x$	$(x + y) \mathbin{|\!\lfloor} z = x \mathbin{|\!\lfloor} z + y \mathbin{|\!\lfloor} z$
$x \mathbin{|\!\lfloor} \delta = x$	$x \mathbin{|\!\lfloor} (y \parallel z) = (x \mathbin{|\!\lfloor} y) \mathbin{|\!\lfloor} z$

Table 3: The axioms for BCCSP with the leftmerge

Theorem 5.1.1. *The set of axioms in table 3 is ω-complete if Act contains an infinite number of actions.*

Proof. Suppose two terms $t, t' \in \mathbb{T}(\text{BCCSP}_{\llfloor})$ are given. Define $\rho : V \to T(\text{BCCSP}_{\llfloor})$ by $\rho(x) = a_x : \delta$ where a_x is a unique action for each $x \in V$ and a_x does neither occur in t nor in t'. Define $R : T(\text{BCCSP}_{\llfloor}) \to \mathbb{T}(\text{BCCSP}_{\llfloor})$ as follows:

$R(\delta) = \delta$,
$R(a : t) = a : R(t)$ where $a \neq a_x$ for all $x \in V$,
$R(a_x : t) = x \llfloor R(t)$,
$R(t + u) = R(t) + R(u)$,
$R(t \parallel u) = R(t) \parallel R(u)$,
$R(t \llfloor u) = R(t) \llfloor R(u)$.

In order to show the axioms in table 3 ω-complete we must check properties (1), (2) and (3) of theorem 3.1.

(1) We show that $\text{B}_{\llfloor} \vdash R(\rho(u)) = u$ with induction on $u \in \mathbb{T}(\text{BCCSP}_{\llfloor})$, provided u does not contain actions of the form a_x.
$R(\rho(x)) = x \llfloor \delta = x$,
$R(\rho(\delta)) = \delta$,
$R(\rho(t + u)) = R(\rho(t)) + R(\rho(u)) = t + u$,
$R(\rho(a : t)) = R(a : \rho(t)) =^* a : R(\rho(t)) = a : t$.
$=^*$ follows from the fact that $a \neq a_x$ for all $x \in V$.

(2) For the $+$-operator the proof is straightforward: $\text{B}_{\llfloor} \cup \{R(t_i) = R(u_i) | i = 1, 2\} \vdash R(t_1 + t_2) = R(t_1) + R(t_2) = R(u_1) + R(u_2) = R(u_1 + u_2)$. The function names $\llfloor$ and $\parallel$ can be dealt with in the same way. The action prefix case is slightly more complicated. $R(t_1) = R(u_1) \vdash R(a : t_1) = a : R(t_1) = a : R(u_1) = R(a : u_1)$ if $a \neq a_x$ for all $x \in V$. In the other case $R(t_1) = R(u_1) \vdash R(a_x : t_1) = x \llfloor R(t_1) = x \llfloor R(u_2) = R(a_x : u_1)$.

(3) It is straightforward to check the axioms that do not explicitly refer to actions. Here we only check the axiom $a : x \llfloor y = a : (x \parallel y)$. Let $\sigma : V \to T(\Sigma)$ be defined such that $\sigma(x) = t$ and $\sigma(y) = u$. $\text{B}_{\llfloor} \vdash R(a : t \llfloor u) = a : R(t) \llfloor R(u) = a : (R(t) \parallel R(u)) = R(a : (t \parallel u))$ if $a \neq a_x$ for all $x \in V$. In the other case $\text{B}_{\llfloor} \vdash R(a_x : t \llfloor u) = (x \llfloor R(t)) \llfloor R(u) = x \llfloor (R(t) \parallel R(u)) = R(a_x : (t \parallel u))$.

$\square$

In many cases it is easy to show the ω-completeness of the axioms of new features introduced in $\text{BCCSP}_{\llfloor}$. As examples we introduce the silent step τ into $\text{BCCSP}_{\llfloor}$ and we will consider $\text{BCCSP}_{\llfloor}$ in trace semantics.

Example 5.1.2. We add a constant τ (*the silent step* or *internal move*) to $\text{BCCSP}_{\llfloor}$. The new signature is called $\text{BCCSP}^{\tau}_{\llfloor}$. The internal step has been axiomatized in different ways. In [10] τ is characterized by three τ-laws. This characterization is often called *weak bisimulation.*

$$\begin{aligned} a : \tau : x &= a : x, \\ \tau : x + x &= \tau : x, \\ a : (\tau : x + y) &= a : (\tau : x + y) + a : x. \end{aligned}$$

If one adds these laws to $B_{\llcorner\!\llcorner}$, obtaining $B^{\tau}_{\llcorner\!\llcorner}$, we have to add the following two axioms in order to make $B^{\tau}_{\llcorner\!\llcorner}$ ω-complete. Axioms of this form already appeared in [7].

$$\begin{aligned} x \llcorner\!\llcorner \tau : y &= x \llcorner\!\llcorner y, \\ x \llcorner\!\llcorner (\tau : y + z) &= x \llcorner\!\llcorner (\tau : y + z) + x \llcorner\!\llcorner y. \end{aligned}$$

Both new axioms are derivable for all closed instances, and therefore valid in any model for $B^{\tau}_{\llcorner\!\llcorner}$.

In [5] τ is axiomatized by the single equation:

$$a : (\tau : (x + y) + x) = a : (x + y).$$

This variant is called *branching bisimulation*. The set $B_{\llcorner\!\llcorner}$, together with this axiom is called $B^{b}_{\llcorner\!\llcorner}$. The single axiom:

$$x \llcorner\!\llcorner (\tau : (y + z) + y) = x \llcorner\!\llcorner (y + z)$$

suffices to make $B^{b}_{\llcorner\!\llcorner}$ ω-complete. This axiom is derivable for all closed instances, and therefore it holds in any model for $B^{b}_{\llcorner\!\llcorner}$.

We do not give the ω-completeness proofs as they can easily be given along the lines of the proof of theorem 5.1.1. In fact it suffices to only check condition (3) for the new axioms, because conditions (1) and (2) are provable in exactly the same way.

Example 5.1.3. Here we study the ω-completeness of $\mathrm{BCCSP}_{\llcorner\!\llcorner}$ in trace semantics. As any term over the signature $\mathrm{BCCSP}_{\llcorner\!\llcorner}$ can be rewritten to a term over the signature BCCSP by the axioms in $B_{\llcorner\!\llcorner}$, and T is complete for the signature BCCSP in trace semantics, $B_{\llcorner\!\llcorner} \cup \mathrm{T}$ is complete for $\mathrm{BCCSP}_{\llcorner\!\llcorner}$ in trace semantics. For ω-completeness we must add the equation:

$$x \llcorner\!\llcorner y + x \llcorner\!\llcorner z = x \llcorner\!\llcorner (y + z),$$

which is derivable from $B_{\llcorner\!\llcorner} \cup \mathrm{T}$ for all its closed instances. The proof of this fact follows the lines of the proof of theorem 5.1.1.

5.2 Interleaving with communication

In this section the signature BCCSP is extended with the merge, the leftmerge and the *communication merge* ($|$). The signature obtained in this way is called $\mathrm{BCCSP}_{|}$. Its properties are described by the axioms in table 4 which are taken from [2]. We have an additional associative and commutative operator $|$: $Act \times Act \rightarrow Act$ on actions. We assume that Act is closed under $|$. In fact $(Act, |)$ is an abelian semigroup. The axioms in the upper two squares of table 4 combined with the condition that $|$ on actions is commutative and associative, are already complete for $\mathrm{BCCSP}_{|}$-terms in the bisimulation model. This can again easily be seen by the fact that any term over the signature $\mathrm{BCCSP}_{|}$ can be rewritten to a term over BCCSP. For BCCSP the four axioms in the left upper corner of table 4 are complete in the bisimulation model. The axioms in the lower squares are necessary for an ω-complete axiomatization. We call the axiom system in table 4 $B_{|}$.

Example 5.2.1. The following facts are derivable from $B_{|}$. We leave the proofs to the reader.

$x + y = y + x$	
$(x + y) + z = x + (y + z)$	
$x + x = x$	
$x + \delta = x$	
$x \parallel y = x \mathbin{\vert\!\lfloor} y + y \mathbin{\vert\!\lfloor} x + x \mid y$	$x \mid y = y \mid x$
$a : x \mathbin{\vert\!\lfloor} y = a : (x \parallel y)$	$a : x \mid b : y = (a \mid b) : (x \parallel y)$
$\delta \mathbin{\vert\!\lfloor} x = \delta$	$\delta \mid x = \delta$
$(x + y) \mathbin{\vert\!\lfloor} z = x \mathbin{\vert\!\lfloor} z + y \mathbin{\vert\!\lfloor} z$	$(x + y) \mid z = x \mid z + y \mid z$
$(x \mathbin{\vert\!\lfloor} y) \mathbin{\vert\!\lfloor} z = x \mathbin{\vert\!\lfloor} (y \parallel z)$	$(x \mid y) \mid z = x \mid (y \mid z)$
$x \mathbin{\vert\!\lfloor} \delta = x$	$x \mid (y \mathbin{\vert\!\lfloor} z) = (x \mid y) \mathbin{\vert\!\lfloor} z$

Table 4: The axioms for $\text{BCCSP}_{|}$

$x \parallel y = y \parallel x$,
$(x \parallel y) \parallel z = x \parallel (y \parallel z)$,
$(a_1 \mid ... \mid (a_i \mid a_{i+1}) \mid ... \mid a_n) : x = (a_1 \mid ... \mid (a_{i+1} \mid a_i) \mid ... \mid a_n) : x$,
$(a_1 \mid ... \mid (a_i(a_{i+1} \mid a_{i+2})) \mid ... \mid a_n) : x = (a_1 \mid ... \mid ((a_i \mid a_{i+1}) \mid a_{i+2}) \mid ... \mid a_n) : x$.

The last two identities show that it is not necessary to include axioms for the commutativity and the associativity of $|$ on actions in $\text{B}_{|}$.

Theorem 5.2.2. $\text{B}_{|}$ *is ω-complete if Act contains an infinite number of actions.*
Proof. This proof has the same structure as the proof of theorem 5.1.1. We will only give the non-trivial steps of the proof. Suppose two terms $t, t' \in \mathbb{T}(\text{BCCSP}_{|})$ are given. Define $\rho : V \rightarrow T(\text{BCCSP}_{|})$ as follows:

$$\rho(x) = a_x : \delta$$

where a_x is unique for each $x \in V$ and actions a_x do not occur in t or t'. We define $R : T(\text{BCCSP}_{|}) \rightarrow \mathbb{T}(\text{BCCSP}_{|})$ by:

$R(\delta) = \delta$,
$R((a_1 \mid ... \mid a_n) : t) = (a_1 \mid ... \mid a_n) : R(t)$ if $a_i \not\equiv a_x$ for $1 \leq i \leq n$ and $x \in V$,
$R(a_x : t) = x \mathbin{\vert\!\lfloor} R(t)$,
$R((a_x \mid a_1 \mid ... \mid a_n) : t) = x \mid R((a_1 \mid ... \mid a_n) : t)$ for $n \geq 1$,
$R((a_1 \mid a_2 \mid ... \mid a_n) : t) = R(a_2 \mid ... \mid a_n \mid a_1) : t)$ for $n \geq 2$ provided $a_1 \not\equiv a_x$ for all $x \in V$,
$R(t + u) = R(t) + R(u)$,
$R(t \parallel u) = R(t) \parallel R(u)$,
$R(t \mathbin{\vert\!\lfloor} u) = R(t) \mathbin{\vert\!\lfloor} R(u)$,
$R(t \mid u) = R(t) \mid R(u)$.

For ρ and R we now check properties (1), (2) and (3) of theorem 3.1.

(1) Straightforward. In this step the axiom $x \mathbin{\vert\!\lfloor} \delta = x$ plays an essential role.

(2) Straightforward for almost all cases, the only exception being the action prefix operator $(a_1 \mid ... \mid a_n) : x$ where for some a_i $(1 \leq i \leq n)$, $a_i = a_x$ with

$x \in V$. Assuming that $\mathrm{B}_{|} \vdash R(t) = R(u)$ for $t, u \in T(\mathrm{BCCSP}_{|})$, we show that $\mathrm{B}_{|} \vdash R((a_1 \mid ... \mid a_n) : t) = R((a_1 \mid ... \mid a_n) : u)$.

$R((a_1 \mid ... \mid a_n) : t) =$

(a) $x_j \mid (... \mid (x_{j'} \mid ((a_k \mid ... \mid a_{k'}) : R(t)))...) =$
$x_j \mid (... \mid (x_{j'} \mid ((a_k \mid ... \mid a_{k'}) : R(u)))...) =$
$R((a_1 \mid ... \mid a_n) : u)$ if there is a $1 \leq i \leq n$ such that $a_i \neq a_x$ for all $x \in V$.

(b) $x_1 \mid (... \mid (x_{n-1} \mid (x_n \mathbin{\|\!\lfloor} R(t)))...) =$
$x_1 \mid (... \mid (x_{n-1} \mid (x_n \mathbin{\|\!\lfloor} R(u)))...) = R((a_1 \mid ... \mid a_n) : u)$, otherwise.

(3) Only the axioms containing occurrences of the action prefix operator are non trivial to check. So we consider the axioms $a : (x \mathbin{\|\!\lfloor} y) = a : (x \parallel y)$ and $a : x \mid b : y = (a \mid b) : (x \parallel y)$. We start off with the first one. Let $a = (a_1 \mid ... \mid a_n)$ and let σ be a closed substitution such that $\sigma(x) = t$ and $\sigma(y) = u$. Three cases must be considered.

(a) $a_i \neq a_x$ for all $1 \leq i \leq n$ and $x \in V$.
$\mathrm{B}_{|} \vdash R(a : t \mathbin{\|\!\lfloor} u) = a : R(t) \mathbin{\|\!\lfloor} R(u) = a : (R(t) \parallel R(u)) = R(a : (t \parallel u))$.

(b) $a_i = a_{x_i}$ for each $1 \leq i \leq n$ and $x_i \in V$.
$R((a_1 \mid ... \mid a_n) : t \mathbin{\|\!\lfloor} u) =$
$(x_1 \mid (... \mid (x_{n-1} \mid (x_n \mathbin{\|\!\lfloor} R(t)))...)) \mathbin{\|\!\lfloor} R(u) =$
$(((x_1 \mid ... \mid x_{n-1}) \mid x_n) \mathbin{\|\!\lfloor} R(t)) \mathbin{\|\!\lfloor} R(u) =$
$((x_1 \mid ... \mid x_{n-1}) \mid x_n) \mathbin{\|\!\lfloor} (R(t) \parallel R(u)) =$
$(x_1 \mid (... \mid (x_{n-1}) \mid \mid (x_n \mathbin{\|\!\lfloor} (R(t) \parallel R(u))))...)) =$
$R((a_1 \mid ... \mid a_n) : (t \parallel u))$.

(c) For some $1 \leq i \leq n$, $a_i \neq a_x$ for all $x \in V$ and for some $1 \leq i \leq n$, $a_i = a_x$.
$R((a_1 \mid ... \mid a_n) : t \mathbin{\|\!\lfloor} u) =$
$(x_j \mid (... \mid (x_{j'} \mid ((a_{k_1} \mid ... \mid a_{k'}) : R(t)))...)) \mathbin{\|\!\lfloor} R(u) =$
$(x_j \mid ... \mid x_{j'}) \mid ((a_{k_1} \mid ... \mid a_{k'}) : R(t) \mathbin{\|\!\lfloor} R(u)) =$
$(x_j \mid ... \mid x_{j'}) \mid ((a_k \mid ... \mid a_{k'}) : (R(t) \parallel R(u))) =$
$x_j \mid (... \mid (x_{j'} \mid ((a_k \mid ... \mid a_{k'}) : (R(t) \parallel R(u))))...) =$
$R((a_1 \mid ... \mid a_n) : (t \parallel u))$.

We now check the axiom $a : x \mid b : y = (a \mid b) : (x \parallel y)$. We can distinguish 9 cases (cf. checking the axiom $a : x \mathbin{\|\!\lfloor} y = a : (x \parallel y)$). We will not discuss all of these, but restrict ourselves to the case where some of the actions, but not all, in a and b have the form a_x.

$R(a : t \mid b : u) =$
$(x_{j_1} \mid (... \mid (x_{j_1'} \mid (a_{k_1} \mid ... \mid a_{k_1'}) : R(t))...)) \mid (y_{j_2} \mid (... \mid (y_{j_2'} \mid (b_{k_2} \mid ... \mid$
$b_{k_2'}) : R(u))...)) =$
$(x_{j_1} \mid ... \mid x_{j_1'} \mid y_{j_2} \mid ... \mid y_{j_2'}) \mid ((a_{k_1} \mid ... \mid a_{k_1'}) : R(t) \mid (b_{k_2} \mid ... \mid b_{k_2'}) :$
$R(u)) =$
$(x_{j_1} \mid (... \mid (x_{j_1'} \mid (y_{j_2} \mid (... \mid (y_{j_2'} \mid ((a_{k_1} \mid ... \mid a_{k_1'}) \mid (b_{k_2} \mid ... \mid b_{k_2'})) :$
$(R(t) \parallel R(u)))...)))...)) =$
$R((a_1 \mid ... \mid a_n \mid b_1 \mid ... \mid b_n) : (t \parallel u))$.

In the last step we used example 5.2.1 to rearrange the actions.

□

References

[1] J.A. Bergstra and J. Heering. Which data types have ω-complete initial algebra specifications? Technical Report CS-R8958, Center for Mathematics and Computer Science, Amsterdam, December 1989.

[2] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Computation*, 60(1/3):109–137, 1984.

[3] J.A. Bergstra and J.V. Tucker. Top down design and the algebra of communicating processes. *Science of Computer Programming*, 5(2):171–199, 1984.

[4] R.J. van Glabbeek. The linear time - branching time spectrum. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings Concur90, LNCS*, Amsterdam, 1990. Springer Verlag.

[5] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics (extended abstract). In G.X. Ritter, editor, *Information Processing 89*, pages 613–618. Elsevier Science Publishers B.V. (North Holland), 1989.

[6] J. Heering. Partial evaluation and ω-completeness of algebraic specifications. *Theoretical Computer Science*, 43:149–167, 1986.

[7] M. Hennessy. Axiomatising finite concurrent processes. *SIAM Journal of Computing*, 17(5):997–1017, 1988.

[8] D. Kapur and D.R. Musser. Proof by consistency. *Artificial Intelligence*, 31:125–157, 1987.

[9] A. Lazrek, P. Lescanne, and J.-J. Thiel. Tools for proving inductive equalities, relative completeness, and ω-completeness. *Information and Computation*, 84:47–70, 1990.

[10] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer-Verlag, 1980.

[11] R. Milner. A complete axiomatisation for observational congruence of finite-state behaviours. Technical Report ECS-LFCS-86-8, University of Edinburgh, Edinburgh, July 1986.

[12] F. Moller. *Axioms for Concurrency.* PhD thesis, University of Edinburgh, July 1989.

[13] V.L. Murskiĭ. The existence in three-valued logic of a closed class with finite basis, not having a finite complete system of identities. *Doklady Akademii Nauk SSSR*, 163:815–818, 1965. English translation in: *Soviet Mathematics Doklady*, 6:1020-1024, 1965.

[14] D.L. Musser. On proving inductive properties of abstract data types. In *Proceedings, 7th ACM Symp. on Principles of Programming Languages*, pages 154–162, New York, 1980. ACM.

[15] E. Paul. Proof by induction in equational theories with relations between constructors. In B. Courcelle, editor, *9th Coll. on Trees in Algebra and Programming*, pages 211–225, Bordeaux, France, 1984. Cambridge University Press, London.