Sequencing and Intermediate Acceptance: Axiomatisation and Decidability of Bisimilarity

Astrid Belder
Eindhoven University of Technology, Eindhoven, The Netherlands

Bas Luttik
Eindhoven University of Technology, Eindhoven, The Netherlands

Jos Baeten
CWI, Amsterdam, The Netherlands
University of Amsterdam, Amsterdam, The Netherlands

Abstract

The Theory of Sequential Processes includes deadlock, successful termination, action prefixing, alternative and sequential composition. Intermediate acceptance, which is important for the integration of classical automata theory, can be expressed through a combination of alternative composition and successful termination. Recently, it was argued that complications arising from the interplay between intermediate acceptance and sequential composition can be eliminated by replacing sequential composition by sequencing. In this paper we study the equational theory of the recursion-free fragment of the resulting process theory modulo bisimilarity, proving that it is not finitely based, but does afford a ground-complete axiomatisation if a unary auxiliary operator is added. Furthermore, we prove that bisimilarity is decidable for processes definable by means of a finite guarded recursive specification over the process theory.

2012 ACM Subject Classification Theory of computation → Process calculi

Keywords and phrases Sequencing, Sequential composition, Bisimilarity, Axiomatisation, Decidability

Digital Object Identifier 10.4230/LIPIcs.CALCO.2019.11

Acknowledgements We thank the anonymous reviewers for their elaborate reviews.

1 Introduction

Successful termination has been a source of controversy from the early days of process algebra. The process theory CCS [18] does not make the distinction between deadlock and successful termination at all. The process theory ACP [9] does make the distinction semantically, but, although it includes a constant denoting deadlock, it does not, in its original formulation, include a constant denoting the successfully terminated process. Only later proposals were made for including such a constant [1, 24].

From a concurrency-theoretic perspective, including a constant $\mathbf{1}$ for successful termination raises philosophical questions without clear-cut answers. For instance, what is the behaviour of a process $a.\mathbf{1} + \mathbf{1}$ that may non-deterministically choose between performing the action a and successfully terminating? Can it perform the action a at all? Is it successfully terminated even when it can still perform activity? And what does it mean to sequentially compose $a.\mathbf{1} + \mathbf{1}$ with the process $b.\mathbf{1}$? Can $(a.\mathbf{1} + \mathbf{1}) \cdot b.\mathbf{1}$ do a b immediately or should it wait until $a.\mathbf{1} + \mathbf{1}$ has performed the a?

In the classical theory of automata and formal languages, the constant $\mathbf{1}$ has a more accepted status. The algebras of regular expressions and μ-regular expressions include a constant $\mathbf{1}$ denoting the language consisting of the empty string. Without the inclusion of the constant, the correspondence between regular expressions and finite automata [16], and the correspondence between μ-regular expressions and pushdown automata [17, 21] would be

lost. Finite automata and pushdown automata are endowed with an acceptance predicate separate from the transition relation defined on states, and hence they admit *intermediate acceptance*: states may at the same time satisfy the acceptance predicate and have outgoing transitions.

The research presented in this paper is part of a larger project in which we are trying to explore and strengthen connections between the classical theory of automata and formal languages and concurrency theory [3, 4, 5, 6], with the aim to establish a unified theory. Our aim for such a unified theory motivates us to study process algebras including a constant for successful termination.

The operational semantics for sequential composition in the presence of a constant $\mathbf{1}$ denoting successful termination (see, e.g., [2]) prescribes that the sequential composition $(a.\mathbf{1} + \mathbf{1}) \cdot b.\mathbf{1}$ may perform the b transition immediately, on grounds that $a.\mathbf{1} + \mathbf{1}$ satisfies the termination predicate. We refer to this phenomenon as *transparency*. In the presence of recursion, transparency leads to considerable expressiveness; for instance, it facilitates the specification of unboundedly branching behaviour (cf. Example 1 below). Recently, we proposed a revised operational semantics for sequential composition that leads to a different interplay between successful termination and sequential composition [7]. The revised operational rules closely resembles the rules of the *sequencing operator* proposed by Bloom [10], although his theory does not distinguish between deadlock and successful termination. We shall, in this paper, reserve *sequential composition* (denoted by $\cdot$) for the operator with the operational semantics as described in [2] and use *sequencing* (denoted by ;) for the operator with the revised operational semantics.

Under the sequencing interpretation, the process $(a.\mathbf{1} + \mathbf{1})\,;\, b.\mathbf{1}$ cannot perform the b-transition immediately (no transparency); first, the left argument of the sequencing operator must execute until no further activity is possible. The effect of replacing sequential composition by sequencing indirectly changes the interpretation of the constant $\mathbf{1}$: it no longer refers to the option to terminate, but rather signals acceptance. For instance, the process $(a.\mathbf{1} + \mathbf{1})\,;\,(b.\mathbf{1} + \mathbf{1})$ is in an accepting state since both $a.\mathbf{1} + \mathbf{1}$ and $b.\mathbf{1} + \mathbf{1}$ are accepting; the process $a.\mathbf{1}\,;\,(b.\mathbf{1} + \mathbf{1})$ on the other hand is not in an accepting state.

Replacing sequential composition by sequencing has advantages and disadvantages for the integration of automata theory and concurrency theory. A disadvantage is that language equivalence is not a congruence for sequencing (see Remark 5 at the end of Section 2). As was shown in [7], advantages are that, in the theory with sequencing every context-free behaviour can be simulated by a pushdown automaton up to strong bisimilarity, while this is not the case in the theory with sequential composition, and that every executable processes can be specified, up to divergence-preserving branching bisimilarity, in a process theory without recursion but with a first-order recursive nesting operation.

In this paper, we continue the investigation of the theory of sequential processes with sequencing instead of sequential composition.

First, we consider the equational theory of the recursion-free fragment modulo bisimilarity. We prove that the equational theory is not finitely based (i.e., does not admit a finite equational axiomatisation). Then, we introduce an auxiliary unary operator and prove that, using this auxiliary operator the *ground* equational theory (i.e., the set of all valid equations without variables) admits a finite axiomatisation. And finally we present arguments for the conjecture that, even with the auxiliary operator, the full equational theory (i.e., the set of all valid equations with variables) is not finitely based.

Then, we prove that bisimilarity is decidable for processes definable by means of a guarded recursive specification in the theory with sequencing. To this end, we consider the seminal proof by Christensen, Hüttel and Stirling that bisimilarity is decidable for the theory of

sequential processes without intermediate acceptance [13], and observe that several crucial properties needed in their argument fail in a setting with intermediate acceptance. Our contribution is then to show that, when a form of redundant intermediate acceptance is eliminated from recursive specifications, then these properties are restored and the proof ideas of [13] apply to establish decidability.

This paper is organised as follows: In Section 2 we introduce the Theory of Sequential Processes with sequential composition replaced by sequencing, illustrating the difference between the two operators with an example. In Section 3, we consider the equational theory of the recursion-free fragment. In Section 4, we establish decidability of bisimilarity for processes definable by means of a guarded recursive specification over the Theory of Sequential Processes with sequencing instead of sequential composition. In Section 5 we present some conclusions. For elaborate proofs of the results claimed in this article we refer to the first author's MSc thesis [8].

2 Sequential Processes

In this section we present the Theory of Sequential Processes adopting the revised operational semantics for sequential composition proposed in [7]. To emphasise that the operational semantics for sequential composition deviates from that in [2], we shall refer to it by the term sequencing and denote it by ; instead of by $\cdot$, reserving $\cdot$ for the variant of sequential composition in [2].

Let $\mathcal{A}$ be a set of *actions*, symbols denoting atomic events, and let $\mathcal{P}$ be a finite set of *process identifiers*. The sets $\mathcal{A}$ and $\mathcal{P}$ serve as parameter of the process theory $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$ that we shall introduce below. The set of *process expressions* associated with $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$ is generated by the following grammar ($a \in \mathcal{A}$, $X \in \mathcal{P}$):

$$p ::= \mathbf{0} \mid \mathbf{1} \mid a.p \mid p + p \mid p\,;p \mid X \quad .$$

The constants $\mathbf{0}$ and $\mathbf{1}$ respectively denote the *deadlocked* (i.e., inactive but not successfully terminated) process and the *successfully terminated* process. For each $a \in \mathcal{A}$ there is a unary action prefix operator $a._$. The binary operators $+$ and ; denote alternative composition and sequencing, respectively. We adopt the convention that $a._$ binds strongest and $+$ binds weakest. For a (possibly empty) sequence $p_1, \ldots, p_n$ we inductively define $\sum_{i=1}^{n} p_i = \mathbf{0}$ if $n = 0$ and $\sum_{i=1}^{n} p_i = (\sum_{i=1}^{n-1} p_i) + p_n$ if $n > 0$. The symbol ; is often omitted when writing process expressions. In particular, if $\alpha \in \mathcal{P}^*$, say $\alpha = X_1 \cdots X_n$, then α denotes the process expression inductively defined by $\alpha = \mathbf{1}$ if $n = 0$ and $\alpha = (X_1 \cdots X_{n-1})\,;X_n$ if $n > 0$. We denote by $|\alpha|$ the length of the sequence.

A recursive specification over $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$ is a mapping Δ from $\mathcal{P}$ to the set of process expressions associated with $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$. The idea is that the process expression p associated with a process identifier $X \in \mathcal{P}$ by Δ *defines* the behaviour of X. We prefer to think of Δ as a collection of *defining equations* $X \stackrel{\text{def}}{=} p$, exactly one for every $X \in \mathcal{P}$. We shall, throughout the paper, presuppose a recursive specification Δ defining the process identifiers in $\mathcal{P}$, and we shall usually simply write $X \stackrel{\text{def}}{=} p$ for $\Delta(X) = p$. Note that, by our assumption that $\mathcal{P}$ is finite, Δ is finite too.

We associate behaviour with process expressions by defining, on the set of process expressions, a unary acceptance predicate $\downarrow$ (written postfix) and, for every $a \in \mathcal{A}$, a binary transition relation $\xrightarrow{a}$ (written infix), by means of the transition system specification presented in Fig. 1. We write $p \stackrel{a}{\not\to}$ for "there does not exist p' such that $p \xrightarrow{a} p'$" and $p \not\to$ for "$p \stackrel{a}{\not\to}$ for all $a \in \mathcal{A}$". Furthermore, when $w \in \mathcal{A}^*$, say $w = a_1 \ldots a_n$, then we write $p \xrightarrow{w}\!\!\!\!\!\rightarrow p'$

$$\frac{}{a.p \xrightarrow{a} p} \qquad \frac{p \xrightarrow{a} p'}{p+q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p+q \xrightarrow{a} q'} \qquad \frac{p \xrightarrow{a} p' \quad X \stackrel{\text{def}}{=} p}{X \xrightarrow{a} p'}$$

$$\frac{}{\mathbf{1}\downarrow} \qquad \frac{p\downarrow}{(p+q)\downarrow} \qquad \frac{q\downarrow}{(p+q)\downarrow} \qquad \frac{p\downarrow \quad X \stackrel{\text{def}}{=} p}{X\downarrow}$$

$$\frac{p\downarrow \quad q\downarrow}{(p\,;q)\downarrow} \qquad \frac{p \xrightarrow{a} p'}{p\,;q \xrightarrow{a} p'\,;q} \qquad \frac{p\downarrow \quad p \not\twoheadrightarrow \quad q \xrightarrow{a} q'}{p\,;q \xrightarrow{a} q'}$$

Figure 1 Operational semantics for $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$.

if there exist $p_0, \ldots, p_n$ such that $p = p_0$, $p_{i-1} \xrightarrow{a_i} p_i$ $(1 \leq i \leq n)$ and $p_n = p'$. Also, we write $p \longrightarrow p'$ for there exists $a \in \mathcal{A}$ such that $p \xrightarrow{a} p'$. Similarly, we write $p \longrightarrow\!\!\!\!\!\rightarrow p'$ for there exists $w \in \mathcal{A}^*$ such that $p \xrightarrow{w}\!\!\!\!\!\rightarrow p'$ and say that p' is *reachable* from p.

It is well-known that transition system specifications with negative premises may not define a unique transition relation that agrees with provability from the transition system specification [15, 11, 14]. Indeed, in [7] it was already pointed out that the transition system specification in Fig. 1 gives rise to such anomalies, e.g., if Δ includes for X the defining equation $X \stackrel{\text{def}}{=} X\,;a.\mathbf{1} + \mathbf{1}$. For then, on the one hand, if $X \not\twoheadrightarrow$, according to the rules for sequencing and recursion we find that $X \xrightarrow{a} \mathbf{1}$, while on the other hand, the transition $X \xrightarrow{a} \mathbf{1}$ is not provable from the transition system specification.

We remedy the situation by restricting our attention to *guarded* recursive specifications, i.e., we require that every occurrence of a process identifier in the definition of some (possibly different) process identifier occurs within the scope of an action prefix. If Δ is guarded, then it is straightforward to prove that the mapping S from process expressions to natural numbers inductively defined by $S(\mathbf{1}) = S(\mathbf{0}) = S(a.p) = 0$, $S(p_1 + p_2) = S(p_1\,;p_2) = S(p_1) + S(p_2) + 1$, and $S(X) = S(p)$ if $(X \stackrel{\text{def}}{=} p) \in \Delta$ gives rise to a so-called *stratification* S' from transitions to natural numbers defined by $S'(p \xrightarrow{a} p') = S(p)$ for all $a \in \mathcal{A}$ and process expressions p and p'. In [15] it is proved that whenever such a stratification exists, then the transition system specification defines a unique transition relation that agrees with provability in the transition system specification.

The operational rules in Fig. 1 deviate from the operational rules for the process theory $\mathrm{TSP}(\mathcal{A})$ discussed in [2] in only two ways: to get the rules for $\mathrm{TSP}(\mathcal{A})$, the symbol ; should be replaced by $\cdot$, and the negative premise $p \not\twoheadrightarrow$ should be removed from the right-most rule for sequencing. The replacement of ; by $\cdot$ is, of course, insignificant; the removal of the negative premise $p \not\twoheadrightarrow$, however, does have a significant impact. The negative premise ensures that a sequencing can only proceed to execute its second argument when its first argument not only satisfies the acceptance predicate, but also cannot perform any further activity. The semantic difference between ; and $\cdot$ is illustrated in the following example.

▶ **Example 1.** Consider the recursive specification

$$X \stackrel{\text{def}}{=} a.(XY) + b.\mathbf{1} \qquad Y \stackrel{\text{def}}{=} c.\mathbf{1} + \mathbf{1}\ .$$

Depending on whether we interpret the concatenation of process identifiers as sequential composition $(\cdot)$ or sequencing $(;)$, we obtain the transition system shown in Fig. 2 with or without the dashed c-transitions. Note that, under the $\cdot$-interpretation, the phenomenon of *transparency* plays a role: from Y^n we have c-transitions to every Y^k with $k < n$, by

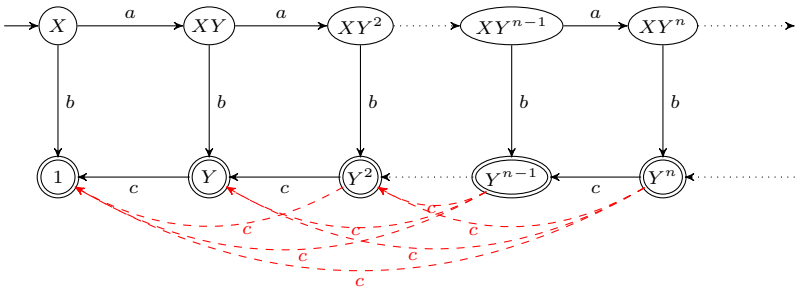

Figure 2 The difference between ; and ·.

executing the c-transition of the kth occurrence of Y, thus skipping the first $k-1$ occurrences of Y. This behaviour is prohibited by the negative premise in the rule for ;, for, since $Y \xrightarrow{c} \mathbf{1}$, none of the occurrences of Y can be skipped.

▶ Remark 2. As Fig. 2 illustrates, the use of sequential composition (as opposed to sequencing) in guarded recursive specifications may give rise to an unbounded reachable branching degree (i.e., there need not be an upper bound on the branching degrees of states reachable from some particular state). As far as we know, this is the only process algebra without an operator for parallel composition that facilitates communication between parallel components that gives rise to unboundedly branching behaviour. It is this kind of behaviour that, e.g., cannot be exhibited by the transition system associated with a pushdown automaton [3].

We proceed to define when two process expressions are behaviourally equivalent.

▶ **Definition 3.** *A binary relation R on the set of process expressions associated with* $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$ *is a* bisimulation *iff R is symmetric and for all p and q such that $(p,q)\in R$:*
1. *If $p \xrightarrow{a} p'$, then there exists a term q', such that $q \xrightarrow{a} q'$, and $(p',q')\in R$.*
2. *If $p{\downarrow}$, then $q{\downarrow}$.*

Process expressions p and q are bisimilar (notation: $p \leftrightarroweq q$) iff there exists a bisimulation R such that $(p,q)\in R$.

The operational rules presented in Fig 1 are in the so-called *panth format* from which it immediately follows that bisimilarity is a congruence [23].

▶ **Proposition 4.** *The relation $\leftrightarroweq$ is a congruence for* $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$.

▶ Remark 5. Note that language equivalence is not a congruence for the sequencing operator: $a.b.\mathbf{1}+a.\mathbf{1}$ and $a.(b.\mathbf{1}+\mathbf{1})$ have the same language $\{ab,a\}$, but the language of $(a.b.\mathbf{1}+a.\mathbf{1})\,;c.\mathbf{1}$ is $\{abc,ac\}$ and the language of $a.(b.\mathbf{1}+\mathbf{1})\,;c.\mathbf{1}$ is $\{abc\}$.

3 Equational theory

In this section we shall consider $\mathrm{TSP}^{;}(\mathcal{A},\emptyset)$, i.e., the recursion-free fragment of the Theory of Sequential Processes. Let us abbreviate $\mathrm{TSP}^{;}(\mathcal{A},\emptyset)$ by $\mathrm{TSP}^{;}(\mathcal{A})$.

For the purpose of concisely expressing equational properties, we shall use variables from some countably infinite set $\mathcal{V}$ of variables. (These variables should be thought of as ranging over process expressions, and should not be confused with process identifiers.) The set of $\mathrm{TSP}^{;}(\mathcal{A})$-terms is generated by the following grammar ($a\in\mathcal{A}$, $x\in\mathcal{V}$):

$$t ::= \mathbf{0} \mid \mathbf{1} \mid a.t \mid t+t \mid t\,;t \mid x \quad .$$

A $\mathrm{TSP}^{;}(\mathcal{A})$-term is *closed* if it does not contain variables. Note that the set of closed $\mathrm{TSP}^{;}(\mathcal{A})$-terms coincides with the set of process expressions associated with $\mathrm{TSP}^{;}(\mathcal{A},\emptyset)$ in the previous section. A *closed substitution* is a mapping σ from variables to process expressions. If t is a $\mathrm{TSP}^{;}(\mathcal{A})$-term and σ is a closed substitution, then we denote by $\sigma(t)$ the process expression obtained by replacing every occurrence of a variable x in t by $\sigma(x)$.

Let t and u be $\mathrm{TSP}^{;}(\mathcal{A})$-terms; an expression of the form $t = u$ is called a $\mathrm{TSP}^{;}(\mathcal{A})$-*equation*. A $\mathrm{TSP}^{;}(\mathcal{A})$-equation $t = u$ is *valid* if $\sigma(t) \leftrightarrows \sigma(u)$ for every closed substitution σ. The *equational theory* of $\mathrm{TSP}^{;}(\mathcal{A})$ is the set of all valid $\mathrm{TSP}^{;}(\mathcal{A})$-equations.

Let E be a set of valid equations and let $t = u$ be a $\mathrm{TSP}^{;}(\mathcal{A})$-equation. We shall write $E \vdash t = u$ if $t = u$ can be derived from the equations in E by means of the rules of equational logic. We wish to characterise the equational theory of $\mathrm{TSP}^{;}(\mathcal{A})$ by giving a finite collection E of valid $\mathrm{TSP}^{;}(\mathcal{A})$-equations such that $E \vdash t = u$ for every valid $\mathrm{TSP}^{;}(\mathcal{A})$-equation $t = u$. Such a collection E is then referred to as a *finite basis* for the equational theory of $\mathrm{TSP}^{;}(\mathcal{A})$; we say that an equational theory is *finitely based* if there exists a finite basis for it.

We shall prove two fundamental results pertaining to the equational theory of $\mathrm{TSP}^{;}(\mathcal{A})$. First, we shall establish that there does not exist a finite basis for the equational theory of $\mathrm{TSP}^{;}(\mathcal{A})$. Second, we shall prove that when an auxiliary operator is added, then the resulting *ground* equational theory (consisting only of all valid $\mathrm{TSP}^{;}(\mathcal{A})$-equations *without* variables) is finitely based. At the end of Section 3.2 we shall conclude with presenting some evidence for a conjecture that, even with the auxiliary operator added, the full equational theory (consisting of all valid $\mathrm{TSP}^{;}(\mathcal{A})$-equations *with* variables) is not finitely based.

3.1 $\mathrm{TSP}^{;}(\mathcal{A})$ is not finitely based

A central axiom of the theory of $\mathrm{TSP}(\mathcal{A})$ of [2] is the axiom $(x + y) \cdot z = x \cdot z + y \cdot z$, which expresses that sequential composition distributes from the right over alternative composition. For sequencing, the axiom is no longer valid in general as the following example illustrates.

▶ **Example 6.** Consider the process expressions

$$p \equiv (a.\mathbf{1} + \mathbf{1})\,;b.\mathbf{1} \text{ and } q \equiv a.\mathbf{1}\,;b.\mathbf{1} + \mathbf{1}\,;b.;\mathbf{1}\ .$$

(We write $\equiv$ for syntactic equality of $\mathrm{TSP}^{;}(\mathcal{A})$-terms and reserve $=$ to express $\mathrm{TSP}^{;}(\mathcal{A})$-equations.) Note that, one the one hand, since $a.\mathbf{1} + \mathbf{1} \xrightarrow{a} \mathbf{1}$, we have that $p \overset{b}{\nrightarrow}$. On the other hand, since $\mathbf{1}{\downarrow}$ and $\mathbf{1} \overset{b}{\nrightarrow}$, we do have that $\mathbf{1}\,;b.\mathbf{1} \xrightarrow{b} \mathbf{1}$ and hence $q \xrightarrow{b} \mathbf{1}$. It follows that p and q are not bisimilar.

Note that, a fortiori, we have that $p \leftrightarrows a.\mathbf{1}\,;b.\mathbf{1}$. That the first argument of the sequencing operator satisfies the acceptance predicate has no effect, because the second argument of the sequencing operator does not satisfy the acceptance predicate. Thus, if the second argument of sequencing does not satisfy the acceptance predicate, then a $\mathbf{1}$-summand in the first argument is redundant.

We shall prove that the redundancy of $\mathbf{1}$ at the left-hand side of sequencing cannot be finitely axiomatised without using an auxiliary operator. To this end, let us fix $\tilde{a}, \tilde{b} \in \mathcal{A}$ and consider the following infinite collection of valid equations ($n \in \mathbb{N}$):

$$(\tilde{a}.\mathbf{1} + \mathbf{1})\,;\sum_{i=1}^{n} \tilde{b}.(\tilde{b}.\mathbf{1} + \mathbf{1})^i = \tilde{a}.\mathbf{1}\,;\sum_{i=1}^{n} \tilde{b}.(\tilde{b}.\mathbf{1} + \mathbf{1})^i\ . \tag{e_n}$$

(For every natural number i process expression p, p^i denotes the *iterated* sequencing of p, inductively defined by $p^0 = \mathbf{1}$ and $p^{i+1} = p^i\,;p$.)

Each of these equations expresses the redundancy of the occurrence of $\mathbf{1}$ in the subexpression $\tilde{a}.\mathbf{1}+\mathbf{1}$ on the left-hand side of the equation. For this redundancy it is important that the right-hand side of the sequencing operator, i.e., the process expression $\sum_{i=1}^{n}\tilde{b}.(\tilde{b}.\mathbf{1}+\mathbf{1})^i$, does not satisfy the acceptance predicate, since it is a summation of $\tilde{b}$-prefixes without $\mathbf{1}$-summand. That the number of summands is n will be used in our argument that (e_n), for sufficiently large $n\in\mathbb{N}$, cannot be derived from a particular finite collection of valid equations. Instead of referring to the notion of number of summands, it is more convenient to refer to the notion of width that we shall now define.

▶ **Definition 7.** *The* width *of a process expression p, written as $width(p)$, is the cardinality of the set $\{p' \mid p \xrightarrow{a} p', \text{ for some } a\in A\}$. We extend the notion of width to $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-terms by defining, for all $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-terms t, that $width(t) = width(\sigma_{\mathbf{0}}(t))$ where $\sigma_{\mathbf{0}}$ denotes the closed substitution that maps all variables to $\mathbf{0}$.*

Note that variables do not contribute to the width of a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term.

Suppose that E is a finite set of valid equations, and let $n\in\mathbb{N}$ exceed the maximum of the widths of all subterms occurring in the equations in E. To prove that (e_n) cannot be derived from E, we define a predicate Ψ_n on $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-terms that is satisfied by the left-hand side (e_n), but not by the right-hand side, and that is maintained by equational derivations from E.

▶ **Definition 8.** *Let p be a process expression. For every $n\in\mathbb{N}$, we define that $\Phi_n(p)$ holds iff $p\equiv p_1\,;p_2$ such that $p_1 \leftrightarrows \tilde{a}.\mathbf{1}+\mathbf{1}$ and $p_2 \leftrightarrows \sum_{i=1}^{n}\tilde{b}.(\tilde{b}.\mathbf{1}+\mathbf{1})^i$. For every $n\in\mathbb{N}$, we define $\Psi_n(p)$ iff $p \leftrightarrows (\tilde{a}.\mathbf{1}+\mathbf{1})\,;\sum_{i=1}^{n}\tilde{b}.(\tilde{b}.\mathbf{1}+\mathbf{1})^i$ and p has a summand $p_1\,;p_2$ such that one of the following cases holds:*

1. *$\Phi_n(p_1\,;p_2)$.*
2. *$p_1 \leftrightarrows \mathbf{1}$ and $\Psi_n(p_2)$.*
3. *$\Psi_n(p_1)$ and $p_2 \leftrightarrows \mathbf{1}$.*

The predicate Φ_n formalises a property satisfied by the left-hand side of (e_n), but not by the right-hand side. The property Φ_n is, however, not preserved by equational derivations due to certain trivial syntactic manipulations involving, e.g., the idempotence of $+$, $\mathbf{0}$ being a neutral element for $+$ and $\mathbf{1}$ being a left- and right neutral element for sequencing (see Table 1 below). The definition of Ψ_n takes such syntactic manipulations into account. Note that the definition of Ψ_n is with recursion on the syntactic structure; it is well-defined since in the last two cases of its definition it is evaluated on a proper subterm.

In general, bisimilarity does not preserve width as defined above, but it does hold that if $p \leftrightarrows q$ and there exist process expressions $p_1,\ldots,p_n$ such that $p \xrightarrow{a} p_i$ and $p_i \leftrightarrows p_j$ implies $i=j$ for all $1\le i,j\le n$, then $width(q)\ge n$. Note that the process expression $\sum_{i=1}^{n}\tilde{b}.(\tilde{b}.\mathbf{1}+\mathbf{1})^i$ has this property. We exploit it to argue that if t is a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term such that $width(t)<n$ and σ is a closed substitution such that $\sigma(t) \leftrightarrows \sum_{i=1}^{n}\tilde{b}.(\tilde{b}.\mathbf{1}+\mathbf{1})^i$, then necessarily t has a variable summand, say x, such that $\sigma(x)\xrightarrow{\tilde{b}}$. This means that with a minor modification of σ, we obtain a substitution $\vartheta_{(\sigma,x)}$ such that $\vartheta_{(\sigma,x)}(t)\downarrow$. We define $\vartheta_{(\sigma,x)}$ as follows:

$$\vartheta_{(\sigma,x)}(y)=\begin{cases}\sigma(y)+\mathbf{1} & \text{if } y=x\\ \sigma(y) & \text{otherwise.}\end{cases}$$

The following lemma essentially applies this idea in a slightly more general situation, where $\sigma(t)$ satisfies Ψ_n.

▶ **Lemma 9.** *Let t be a* $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$*-term and let n be a natural number such that $\mathit{width}(t') < n$ for every subterm t' of t. If $\Psi_n(\sigma(t))$ for some closed substitution σ, then there is a variable x such that $\vartheta_{(\sigma,x)}(t)\!\downarrow$ and either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{\tilde{b}}$.*

Proof. See the proof of Lemma 35 in Appendix A. ◀

The following lemma establishes the converse of Lemma 9.

▶ **Lemma 10.** *Let t be a* $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$*-term, let x be a variable, and let σ be a closed substitution. If $\sigma(t) \leftrightarrows (\tilde{a}.\mathbf{1}+\mathbf{1})\,;\sum_{i=1}^{n}(\tilde{b}.\mathbf{1}\,;(\tilde{b}.\mathbf{1}+\mathbf{1})^i)$, $\vartheta_{(\sigma,x)}(t)\!\downarrow$ and either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{\tilde{b}}$, then $\Psi_n(\sigma(t))$.*

Proof. See the proof of Lemma 40 in Appendix A. ◀

The following theorem states that if E is a set of valid equations and n exceeds the maximum of the widths of all subterms of the equations in E, then equational derivations from E preserve Ψ_n.

▶ **Theorem 11.** *Let E be a finite set of valid* $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$*-equations, and let n be a natural number such that for each axiom $t = u \in E$, for each subterm t' of t and each subterm u' of u, $\mathit{width}(t') < n$ and $\mathit{width}(u') < n$. Furthermore, let p and q be closed* $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$*-terms such that $E \vdash p = q$. It then holds that if $\Psi_n(p)$, then $\Psi_n(q)$.*

Proof. The proof is by induction on a derivation of the equation $p = q$ from E. So, we distinguish cases, according to the last rule used in this derivation, and assume that for each derivation of $p' = q'$ that is a sub-derivation of the derivation of $p = q$, if $\Psi_n(p')$ then $\Psi_n(q')$ (IH). Here we only consider the most interesting case in which the derivation consists of a substitution instance of an axiom in E.

If $p = q$ is a substitution instance of an axiom in E, then there exist $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-terms t and u and a closed substitution σ such that $\sigma(t) = p$, $\sigma(u) = q$ and $t = u \in E$. If $\Psi_n(p)$, then $\Psi_n(\sigma(t))$ and thus $\sigma(t) \leftrightarrows (\tilde{a}.\mathbf{1}+\mathbf{1})\,;\sum_{i=1}^{n}(\tilde{b}.\mathbf{1}\,;(\tilde{b}.\mathbf{1}+\mathbf{1})^i)$. Since $t = u$ is sound with respect to bisimilarity, $\sigma(u) \leftrightarrows \sigma(t) \leftrightarrows (\tilde{a}.\mathbf{1}+\mathbf{1})\,;\sum_{i=1}^{n}(\tilde{b}.\mathbf{1}\,;(\tilde{b}.\mathbf{1}+\mathbf{1})^i)$. Furthermore, by Lemma 9, there must be some variable x such that $\vartheta_{(\sigma,x)}(t)\!\downarrow$ and either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{\tilde{b}} p$ for some closed $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term p. Hence, since $\vartheta_{(\sigma,x)}(t) \leftrightarrows \vartheta_{(\sigma,x)}(u)$, also $\vartheta_{(\sigma,x)}(u)\!\downarrow$. Then, since $\sigma(u) \leftrightarrows (\tilde{a}.\mathbf{1}+\mathbf{1})\,;\sum_{i=1}^{n}(\tilde{b}.\mathbf{1}\,;(\tilde{b}.\mathbf{1}+\mathbf{1})^i)$, and $\vartheta_{(\sigma,x)}(u)\!\downarrow$, by Lemma 10, we conclude that $\Psi_n(\sigma(u))$ holds, and thus $\Psi_n(q)$ holds. ◀

We use Theorem 11 to prove that the equational theory of $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$ is not finitely based by showing that no set E of valid $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-equations can be a finite basis. To this end, let E be a finite set of valid $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-equations. Then, since E has finitely many equations and the terms occurring on both sides of these equations each have finitely many subterms, there exists $n \in \mathbb{N}$ that exceeds the widths of all these subterms. By Theorem 11 we have that whenever $E \vdash p = q$ and $\Psi_n(p)$, then also $\Psi_n(q)$; it follows that $E \nvdash (e_n)$. Since (e_n) is a valid $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-equation, it follows that E is not a finite basis for the equational theory of $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$. Thus, we obtain the following corollary.

▶ **Corollary 12.** *There does not exist a finite basis for* $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$.

3.2 Ground-completeness with an auxiliary operator

By the *ground* equational theory of $\mathrm{TSP}^{;}(\mathcal{A})$ we mean the set of all valid $\mathrm{TSP}^{;}(\mathcal{A})$-equations *without variables*. Note that, since the equations (e_n) do not include variables and the predicate Ψ_n is defined on process expressions, it immediately follows from Theorem 11 that the ground equational theory of $\mathrm{TSP}^{;}(\mathcal{A})$ is not finitely based either. We proceed to extend $\mathrm{TSP}^{;}(\mathcal{A})$ with a unary auxiliary operator NT and show that ground equational theory of the extension $\mathrm{TSP}^{;}_{NT}(\mathcal{A})$ is finitely based.

The syntax of $\mathrm{TSP}^{;}_{NT}(\mathcal{A})$ consists of the syntax of $\mathrm{TSP}^{;}(\mathcal{A})$ with the unary operation NT added; this operation can be used both in the construction of process expressions associated with $\mathrm{TSP}^{;}_{NT}(\mathcal{A})$ and of $\mathrm{TSP}^{;}_{NT}(\mathcal{A})$-terms. Intuitively, $NT(p)$ denotes the non-terminating part of p; for example, $NT(a.p) = a.p$ and $NT(a.p + \mathbf{1}) = a.p$.

$$\frac{p \xrightarrow{a} p'}{NT(p) \xrightarrow{a} p'}$$

Figure 3 The operational rule for NT.

The operational rule for NT is presented in Figure 3. The rule is in the panth format, so bisimilarity is a congruence also for the extended theory. Furthermore, from [22, Theorem 3.9] it follows that $\mathrm{TSP}^{;}_{NT}(\mathcal{A})$ is an operational conservative extension of $\mathrm{TSP}^{;}(\mathcal{A})$, meaning that $\mathrm{TSP}^{;}(\mathcal{A})$ process expressions have the same operational semantics in the extended theory $\mathrm{TSP}^{;}_{NT}(\mathcal{A})$.

Table 1 A finite basis for the ground equational theory of $\mathrm{TSP}^{;}_{NT}(A)$.

$x + y$	$=$	$y + x$	A1
$x + (y + z)$	$=$	$(x + y) + z$	A2
$x + x$	$=$	x	A3
$(x \,;\, y) \,;\, z$	$=$	$x \,;\, (y \,;\, z)$	A5
$x + \mathbf{0}$	$=$	x	A6
$\mathbf{0} \,;\, x$	$=$	$\mathbf{0}$	A7
$x \,;\, \mathbf{1}$	$=$	x	A8
$\mathbf{1} \,;\, x$	$=$	x	A9
$a.x \,;\, y$	$=$	$a.(x \,;\, y)$	A10
$NT(x + y) \,;\, z$	$=$	$NT(x) \,;\, z + NT(y) \,;\, z$	A11
$(a.x + y + \mathbf{1}) \,;\, NT(z)$	$=$	$(a.x + y) \,;\, NT(z)$	A12
$(a.x + y + \mathbf{1}) \,;\, (z + \mathbf{1})$	$=$	$(a.x + y) \,;\, (z + \mathbf{1}) + \mathbf{1}$	A13
$NT(\mathbf{0})$	$=$	$\mathbf{0}$	NT1
$NT(\mathbf{1})$	$=$	$\mathbf{0}$	NT2
$NT(a.x)$	$=$	$a.x$	NT3
$NT(x + y)$	$=$	$NT(x) + NT(y)$	NT4

Table 1 presents a finite collection of valid $\mathrm{TSP}^{;}_{NT}(\mathcal{A})$-equations. It includes the well-known axioms A1–3 and A5–10 adapted from $\mathrm{TSP}(\mathcal{A})$ (see [2]). Note, however, that the axiom A4, which in $\mathrm{TSP}(\mathcal{A})$ expresses distributivity from the right of sequencing over

alternative composition, has been omitted since it is not valid. It has been replaced by axiom A11, which, intuitively, expresses that sequencing distributes from the right over alternative composition only if the alternative composition does not satisfy the acceptance predicate. Axioms A12 and A13 allows us to eliminate redundant occurrences of $\mathbf{1}$ at the left-hand side of sequencing. Finally, axioms NT1–4 express the interaction of NT with the constants $\mathbf{0}$ and $\mathbf{1}$, action prefix and alternative composition.

For detailed proofs that the axioms in Table 1 are valid we refer to [8]. We shall, henceforth, write $\mathrm{TSP}^{\mathrm{i}}_{NT}(A) \vdash t = u$ if the $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$-equation $t = u$ can be derived from the axioms in Table 1 using the rules of equational logic. To prove that the axioms in Table 1 constitute a finite basis for the ground equational theory of $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$, we use the following elimination theorem.

▶ **Theorem 13.** *For every process expression p associated with $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$ there exists a process expression q without occurrences of $;$ and NT such that $\mathrm{TSP}^{\mathrm{i}}_{NT}(A) \vdash p = q$.*

In [2, Theorem 4.4.12] it is proved that axioms A1–3 and A6 constitute a finite basis for the ground equational theory of $\mathrm{BSP}(\mathcal{A})$, which is obtained from $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$ by removing $;$ and NT. Hence, we get the following corollary from Theorem 13.

▶ **Corollary 14.** *The axioms in Table 1 constitute a finite basis for the ground equational theory of $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$.*

The axioms in Table 1 do not constitute a finite basis for the full equational theory of $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$. For example, it is easy to see that the valid equation $NT(NT(x)) = NT(x)$ cannot be derived from $\mathrm{TSP}^{\mathrm{i}}_{NT}(A)$. We proceed to argue that, although the ground equational theory of $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$ is finitely based, the full equational theory of $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$ is not; the argument will be very similar to the argument showing that $\mathrm{TSP}^{\mathrm{i}}_{NT}(A)$ is not finitely based.

Consider the equation:

$$(x + \mathbf{1}) ; x = x ; x \ . \tag{1}$$

To see that it is valid, note that the symmetric closure of the relation

$$R = \{((p + \mathbf{1}) ; p, p ; p), (p, p) \mid p \text{ a } \mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A}) \text{ process expression}\}$$

is a bisimulation relation.

Recall the equations (e_n) used in Section 3.1 to show that the equational theory of $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$ is not finitely based. For the redundancy of the $\mathbf{1}$-summand in the left-hand side of the sequencing operator it is essential that the left-hand side also admits a transition while the right-hand side does not satisfy the acceptance predicate. Equation (1) above does not satisfy this property for every closed substitution. Nevertheless, the $\mathbf{1}$-summand is redundant, due to the fact that x appears in both arguments of the sequencing operator. The idea can be generalised, resulting in the infinite collection of valid equations $n \in \mathbb{N}$):

$$(x + \mathbf{1}) ; \sum_{i=1}^{n} (x ; (a.\mathbf{1} + \mathbf{1})^i) = x ; \sum_{i=1}^{n} (x ; (a.\mathbf{1} + \mathbf{1})^i) \ . \tag{e'_n}$$

Similarly to the equations (e_n) used in Section 3.1, the size of the right hand side of this equation is not bounded. We conjecture that by similar reasoning as used in Section 3.1 it can be argued that there does not exist a finite set of valid $\mathrm{TSP}^{\mathrm{i}}_{NT}(\mathcal{A})$-equations from which the equations e'_n can be derived for all $n \in \mathbb{N}$.

4 Decidability

Christensen, Hüttel and Stirling have established that bisimilarity is decidable for processes definable by means of a guarded recursive BPA specification [13], where BPA can be thought of as $\mathrm{TSP}^{;}(\mathcal{A})$ without intermediate acceptance. Our goal in this section is to extend that decidability result to $\mathrm{TSP}^{;}(\mathcal{A})$. Our proof closely follows the presentation of the decidability proof for BPA in [12], and we shall focus on the extension and skip over parts that are similar.

The starting point is the presupposed $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$ recursive specification Δ, which is finite since we have assumed that $\mathcal{P}$ is finite. The decision problem we wish to solve is: Given any two process expressions p and q does it hold that $p \leftrightarrows q$? We shall first recall a few standard observations to simplify the formulation of the decision problem.

The first observation is that we may assume, without loss of generality, that p and q are both process identifiers. For if not then we could first solve the decision problem for process identifiers, and then decide $p \leftrightarrows q$ by considering $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P}')$, where $\mathcal{P}'$ is $\mathcal{P}$ with two new process identifiers X and Y added and Δ' is Δ with the two extra defining equations $X \stackrel{\mathrm{def}}{=} p$ and $Y \stackrel{\mathrm{def}}{=} q$, and determine whether $X \leftrightarrows Y$.

The second observation is that we may assume, again without loss of generality, that Δ is in so-called *Greibach Normal Form* (GNF): for every defining equation $(X \stackrel{\mathrm{def}}{=} p) \in \Delta$ we have that

$$p \equiv \sum_{i=1}^{n} a_i.\alpha_i(+\mathbf{1}) \quad . \tag{2}$$

Here we assume that $n \in \mathbb{N}$ (recall our convention that the empty summation denotes $\mathbf{0}$), $\alpha_i \in \mathcal{P}^*$, and $(+\mathbf{1})$ denotes an optional $\mathbf{1}$-summand. We refer to [8] for the description of an effective procedure that associates with every recursive specification Δ over $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$ a set $\mathcal{P}' \supseteq \mathcal{P}$ and a recursive specification Δ' over $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P}')$ in GNF such that for all $X, Y \in \mathcal{P}$ we have that $X \leftrightarrows Y$ with respect to Δ if, and only if, $X \leftrightarrows Y$ with respect to Δ'. The advantage of assuming that Δ is in GNF is that then every process expression reachable (by following the transition relation) from a process identifier associated with $\mathrm{TSP}^{;}(\mathcal{A},\mathcal{P})$ is an element of $\mathcal{P}^*$.

The third observation is that it is semi-decidable whether $p \not\leftrightarrows q$. This is a straightforward consequence of the well-known fact that for image-finite transition systems there is a stratified characterisation of bisimilarity; see [8] for such a characterisation taking the acceptance predicate into account. Therefore, to solve the aforementioned decision problem, it suffices to argue that bisimilarity is semi-decidable.

The argument presented in [12] to show that bisimilarity is semi-decidable for BPA then proceeds by showing that process identifiers X and Y are bisimilar if, and only if, there exists a *finite bisimulation base*[1] that contains the pair (X, Y), and that it is semi-decidable whether a finite binary relation on $\mathcal{P}^*$ is a bisimulation base. It then follows that $X \leftrightarrows Y$ is semi-decidable: enumerate all finite binary relations on $\mathcal{P}^*$ containing the pair (X, Y) and check, in parallel, whether one of them is a bisimulation base.

We adapt the definition of bisimulation base, originally from [13], to our setting with an acceptance predicate. It uses the following auxiliary notation: if R is a binary relation on $\mathcal{P}^*$,

[1] Note that a *bisimulation base* is a *bisimulation up to congruence* with respect to the operation of concatenation on finite sequences of process identifiers [20].

then we denote by $\overset{R}{\equiv}$ the least equivalence relation that contains R and all pairs $(\alpha\alpha', \beta\beta')$ whenever it contains the pairs (α, β) and (α', β').

▶ **Definition 15.** *A binary relation R on $\mathcal{P}^*$ is a bisimulation base if, and only if, R is symmetric and for all pairs $(\alpha, \beta) \in R$ and all $a \in \mathcal{A}$, it holds that:*

- *if $\alpha \xrightarrow{a} \alpha'$, then $\beta \xrightarrow{a} \beta'$ for some β' such that $\alpha' \overset{R}{\equiv} \beta'$; and*
- *if $\alpha\downarrow$, then $\beta\downarrow$.*

Our goal will be to show that there exists a finite bisimulation base R such that $\alpha \overset{R}{\equiv} \beta$ if, and only if, $\alpha \leftrightarrows \beta$ for all $\alpha, \beta \in \mathcal{P}^*$. The argument relies on a partitioning of the set of process identifiers into *normed* and *unnormed* process identifiers.

▶ **Definition 16.** *Let p be a $\mathrm{TSP}^{;}(\mathcal{A}, \mathcal{P})$ process expression. The* norm *$n(p)$ of p is the length of a shortest transition sequence from p to a process expression bisimilar to $\mathbf{1}$ if such a sequence exists, and ∞ otherwise, i.e.,*

$$n(p) = \min\left(\{|w| \mid \exists p'.\ p \xrightarrow{w} p' \wedge p' \leftrightarrows \mathbf{1}\} \cup \{\infty\}\right) \ .$$

A process expression p is normed *if $n(p) < \infty$; otherwise it is* unnormed*. We denote by $\mathcal{P}_n$ the set of all* normed *process identifiers and by $\mathcal{P}_u$ the set of all* unnormed *process identifiers.*

In the case of BPA, which does not have intermediate acceptance, the following three properties hold for all sequences of process identifiers α, β and γ:

1. if α is unnormed, then $\alpha\beta \leftrightarrows \alpha$;
2. $|\alpha| \leq n(\alpha)$; and
3. if α, β and γ are normed, then $\alpha\gamma \leftrightarrows \beta\gamma$ implies $\alpha \leftrightarrows \beta$.

These properties are crucial for pruning the cardinality of the bisimulation base. The following example illustrate that neither of these properties holds in our setting with intermediate acceptance:

▶ **Example 17.** Consider the following recursive specification in GNF:

$$X \overset{\text{def}}{=} a.YWZ + a.YW + a.ZZ + a.UV \qquad Z \overset{\text{def}}{=} b.\mathbf{1} \qquad V \overset{\text{def}}{=} \mathbf{0}$$
$$Y \overset{\text{def}}{=} b.\mathbf{1} + \mathbf{1} \qquad U \overset{\text{def}}{=} b.U + \mathbf{1} \qquad W \overset{\text{def}}{=} \mathbf{0} + \mathbf{1}$$

Then we have that U is unnormed, but $UV \not\leftrightarrows U$ since $U\downarrow$ whereas $UV\not\downarrow$, refuting the first property. Furthermore, $|YWZ| = 3 > 2 = n(YWZ)$, refuting the second property. And finally $YWZ \leftrightarrows ZZ$, but $YW \not\leftrightarrows Z$, refuting the third property.

Note that the sequences used in Example 17 to refute the properties above all suffer from some form of redundant intermediate acceptance.

Violation of the first property can only be due to the presence of a process identifier that is bisimilar to $\mathbf{1}$. Note that, in a recursive specification in GNF, a process identifier X is bisimilar to $\mathbf{1}$ if, and only if, $(X \overset{\text{def}}{=} \mathbf{0} + \mathbf{1}) \in \Delta$; let us call such a process identifier a $\mathbf{1}$-*identifier*. Whether some process identifier is a $\mathbf{1}$-identifier can easily be decided. Furthermore, occurrences of $\mathbf{1}$-identifiers can simply be eliminated from the right-hand sides of defining equations of other process identifiers. Thus, it remains to solve the decision problem for recursive specifications in GNF without $\mathbf{1}$-identifiers.

Our main contribution in the remainder of this section will be the notion of *Acceptance Irredundant Greibach Normal Form* (AIGNF), a special variant of GNF that precludes redundant intermediate acceptance from sequences reachable from process identifiers. We

shall prove that it is enough to solve the decision problem for recursive specifications in AIGNF and then show that the argument for the existence of a finite bisimulation base of [13] works for such recursive specifications.

4.1 Acceptance Irredundant Greibach Normal Form

We partition the set of process identifiers $\mathcal{P}$ into sets $\mathcal{P}_{\downarrow} = \{X \in \mathcal{P} \mid X\downarrow\}$ and $\mathcal{P}_{\not\downarrow} = \{X \in \mathcal{P} \mid X\not\downarrow\}$. Furthermore, we define the set $\overline{\mathcal{P}}_{\not\downarrow}$ of *hereditarily non-terminating* process identifiers as the largest subset of $\mathcal{P}_{\not\downarrow}$ such that for all $X \in \overline{\mathcal{P}}_{\not\downarrow}$ we have that if $(X \stackrel{\text{def}}{=} p) \in \Delta$ and Y is a process identifier occurring in p, then $Y \in \overline{\mathcal{P}}_{\not\downarrow}$. The set $\overline{\mathcal{P}}_{\not\downarrow}$ can be computed iteratively: start with $\mathcal{P}' = \mathcal{P}_{\not\downarrow}$ and in every iteration remove from $\mathcal{P}'$ all process identifiers X such that $(X \stackrel{\text{def}}{=} p) \in \Delta$ and p has an occurrence of some process identifier Y with $Y \notin \mathcal{P}'$ until a fixed point is reached (i.e., nothing can be removed from $\mathcal{P}'$ anymore). We shall say that $\alpha \in \mathcal{P}^*$ is *acceptance irredundant* if $\alpha \in \overline{\mathcal{P}}^*_{\not\downarrow}\mathcal{P}_{\not\downarrow}\mathcal{P}^*_{\downarrow} \cup \mathcal{P}^*_{\downarrow}$.

▶ **Definition 18.** *A recursive specification Δ is in* Acceptance Irredundant Greibach Normal Form *(AIGNF) if for every defining equation $(X \stackrel{def}{=} p) \in \Delta$ we have that $p \equiv \mathbf{0}$ or*

$$p \equiv \sum_{i=1}^{n} a_i.\alpha_i (+\mathbf{1}) \ ,$$

with $n \in \mathbb{N}^+$ and each α_i acceptance irredundant.

The following example illustrates how a recursive specification in GNF and without $\mathbf{1}$-identifiers can be transformed into AIGNF.

▶ **Example 19.** Consider the following recursive specification in GNF:

$$X \stackrel{\text{def}}{=} a.YZ + a.Y + a.ZZ + a.UV \qquad Z \stackrel{\text{def}}{=} b.\mathbf{1}$$
$$Y \stackrel{\text{def}}{=} b.\mathbf{1} + \mathbf{1}$$

As $Z\not\downarrow$, the intermediate acceptance of Y in $a.YZ$ is redundant. We cannot simply remove it from the definition of Y, however, since in $a.Y$ the intermediate acceptance of Y is not redundant. Instead, we introduce a fresh variable $\bar{Y}$, that is defined as Y but without the intermediate acceptance. Then, we replace all occurrences of Y of which the intermediate acceptance is redundant with $\bar{Y}$, resulting in:

$$X \stackrel{\text{def}}{=} a.\bar{Y}Z + a.Y + a.ZZ + a.UV \qquad Z \stackrel{\text{def}}{=} b.\mathbf{1}$$
$$Y \stackrel{\text{def}}{=} b.\mathbf{1} + \mathbf{1} \qquad \bar{Y} \stackrel{\text{def}}{=} b.\mathbf{1}$$

The idea explained in the preceding example can be exploited to prove the following proposition.

▶ **Proposition 20.** *For every recursive specification Δ over $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \mathcal{P})$ in GNF without $\mathbf{1}$-identifiers there exist $\mathcal{P}' \supseteq \mathcal{P}$ and a recursive specification Δ' in AIGNF over $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \mathcal{P}')$ such that for all $X, Y \in \mathcal{P}$ we have that $X \leftrightarrows Y$ with respect to Δ if, and only if, $X \leftrightarrows Y$ with respect to Δ'.*

Let $\alpha \in \mathcal{P}^*$; we say that α is Δ-*reachable* if there exists $X \in \mathcal{P}$ such that $X \longrightarrow\!\!\!\!\!\rightarrow \alpha$. If Δ is in AIGNF, then it can be shown that all Δ-reachable sequences are acceptance irredundant. Hence, for recursive specifications in AIGNF we now get the three properties needed for the proof that there exists a finite bisimulation base.

▶ **Proposition 21.** *If Δ is in AIGNF, then for all acceptance irredundant sequences α, β, γ:*
1. *if α is unnormed, then $\alpha\beta \leftrightarrows \alpha$;*
2. *$|\alpha| \leq n(\alpha)$; and*
3. *if α, β and γ are normed, then $\alpha\gamma \leftrightarrows \beta\gamma$ implies $\alpha \leftrightarrows \beta$.*

Proof. See Appendix B. ◀

4.2 The existence of a finite bisimulation base

By the first item of Proposition 21, we can, without loss of generality, assume that all sequences of variables appearing in the right-hand sides of the defining equations in our presupposed recursive specification Δ in AIGNF are elements of $\mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u$. Then all Δ-reachable sequences will not only be acceptance irredundant, but also elements of $\mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u$.

The definition of the finite bisimulation base relies on decomposing sequences.

▶ **Definition 22.** *A pair $(X\alpha, Y\beta)$ satisfying $X\alpha \leftrightarrows Y\beta$ is decomposable if X and Y are normed, and there exists γ such that*
- *$X \twoheadrightarrow \gamma$, $X \leftrightarrows Y\gamma$ and $\gamma\alpha \leftrightarrows \beta$; or*
- *$Y \twoheadrightarrow \gamma$, $Y \leftrightarrows X\gamma$ and $\gamma\beta \leftrightarrows \alpha$.*

Two pairs $(X\alpha, Y\beta)$ and $(X\alpha', Y\beta')$ are *distinct* if $\alpha \not\leftrightarrows \alpha'$ or $\beta \not\leftrightarrows \beta'$. A crucial step towards a finite bisimulation base consists of establishing that a relation containing all indecomposable pairs $(X\alpha, Y\beta)$, where $X\alpha, Y\beta \mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u$ are acceptance irredundant sequences such that $X\alpha \leftrightarrows Y\beta$ is necessarily finite.

For the definition of a finite bisimulation base we now need just one more definition, which allows us to choose appropriate candidates among non-distinct indecomposable pairs.

▶ **Definition 23.** *The finite prefix norm $n_f(\alpha)$ of α is defined as follows:*

$$n_f(\alpha) = \max(\{n(\beta) \mid n(\beta) < \infty \text{ and } \alpha = \beta\gamma \text{ for some } \gamma\}).$$

The pre-order $\preceq$ on pairs is defined as:
$(\alpha_1, \alpha_2) \preceq (\beta_1, \beta_2)$ *iff* $\max(n_f(\alpha_1), n_f(\alpha_2)) \leq \max(n_f(\beta_1), n_f(\beta_2))$.

In the following two lemmas, adapted from [12, Lemmas 28 and 29], a relaxed form of cancellation is established for Δ-reachable sequences of process identifiers.

▶ **Lemma 24.** *If $\alpha \leftrightarrows \gamma\alpha$ and $\beta \leftrightarrows \gamma\beta$ for some $\gamma \not\leftrightarrows \mathbf{1}$ and acceptance irredundant $\gamma\alpha$ and $\gamma\beta$, then $\alpha \leftrightarrows \beta$.*

Using this result, we will show a form of cancellation for (potentially unnormed) acceptance irredundant sequences, if $\alpha\gamma \leftrightarrows \beta\gamma$ for infinitely many non-bisimilar γ.

▶ **Lemma 25.** *Let $\alpha, \beta \in \mathcal{P}^*$. If for infinitely many non-bisimilar $\gamma \in \mathcal{P}^*$ such that $\alpha\gamma$ and $\beta\gamma$ are acceptance irredundant it holds that $\alpha\gamma \leftrightarrows \beta\gamma$, then $\alpha \leftrightarrows \beta$.*

The following lemma is an adaptation of [12, Lemma 32] to our setting.

▶ **Lemma 26.** *For all $X, Y \in \mathcal{P}$, every set R of the form*

$$\{(X\alpha, Y\beta) \mid X\alpha, Y\beta \in \mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u \text{ acceptance irredundant sequences}, \\ X\alpha \leftrightarrows Y\beta, \text{ and } (X\alpha, Y\beta) \text{ indecomposable}\}$$

and contains only distinct pairs must be finite.

We now have everything in place to prove the main result of this section.

▶ **Theorem 27.** *Let* $R_1 = \{(X, \alpha) \mid X \in \mathcal{P}_n,\ \alpha \in \mathcal{P}_n$ *such that* $X \leftrightarrows \alpha\}$*, and let* R_2 *be the largest relation of the form*

$$\{(X\alpha, Y\beta) \mid X\alpha, Y\beta \in \mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u \text{ are acceptance irredundant,}\\ X\alpha \leftrightarrows Y\beta, \text{ and } (X\alpha, Y\beta) \text{ indecomposable}\}$$

containing only distinct pairs and minimal elements with respect to $\preceq$*. Then the symmetric closure* R *of* $R_1 \cup R_2$ *is finite and satisfies* $\alpha \overset{R}{\equiv} \beta$ *if and only if* $\alpha \leftrightarrows \beta$ *for all acceptance irredundant sequences* $\alpha, \beta \in \mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u$*.*

Proof. Since Δ is in AIGNF, we have $|\alpha| \leq n(\alpha)$. Hence, since X is normed and $n(X) = n(\alpha)$ we have $|\alpha| \leq n(X)$, and thus α has a finite maximum length. Hence, there can only be finitely many such α as $\mathcal{P}$ is finite. If follows that R_1 is finite. Furthermore, by Lemma 26, R_2 is finite. So R is finite. Hence, since $\leftrightarrows$ is a congruence for $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$, we have $\overset{R}{\equiv} \subseteq \leftrightarrows$. It remains to show is that $\overset{R}{\equiv} \supseteq \leftrightarrows$. We prove by induction on $\preceq$ that $X\alpha \leftrightarrows Y\beta$ implies $X\alpha \overset{R}{\equiv} Y\beta$, for all acceptance irredundant sequences $X\alpha, Y\beta \in \mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u$.
Suppose that $(X\alpha, Y\beta)$ is decomposable, then $X, Y \in \mathcal{P}_n$ and, without loss of generality, assume that $X \longrightarrow\!\!\!\!\!\rightarrow \gamma$ such that $X \leftrightarrows Y\gamma$ and $\gamma\alpha \leftrightarrows \beta$. Then, $n_f(\gamma\alpha) < n_f(Y\gamma\alpha) = n_f(X\alpha)$ and $n_f(\beta) < n_f(Y\beta)$, so $(\gamma\alpha, \beta) \prec (X\alpha, Y\beta)$. Furthermore, since $X \longrightarrow\!\!\!\!\!\rightarrow \gamma$, $X\alpha \longrightarrow\!\!\!\!\!\rightarrow \gamma\alpha$ and thus $\gamma\alpha \in \mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u$ and $\gamma\alpha$ is acceptance irredundant. Moreover, since $Y\beta \in \mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u$, $Y\beta$ is acceptance irredundant and $Y \in \mathcal{P}_n$, it follows that $\beta \in \mathcal{P}_n^* \cup \mathcal{P}_n^*\mathcal{P}_u$ and β is acceptance irredundant, and hence by induction $\gamma\alpha \overset{R}{\equiv} \beta$. Finally, since $\gamma\alpha$ is acceptance irredundant, γ is acceptance irredundant, and therefore $Y\gamma$ is acceptance irredundant. Hence, $(X, Y\gamma) \in R_1$ and thus $X\alpha \overset{R}{\equiv} Y\gamma\alpha \overset{R}{\equiv} Y\beta$.

Now, suppose that $(X\alpha, Y\beta)$ is not decomposable. Then $(X\alpha', Y\beta') \in R_2$ for some $\alpha' \leftrightarrows \alpha$ and $\beta' \leftrightarrows \beta$ with $(\alpha', \beta') \preceq (\alpha, \beta)$. We distinguish three cases.

- If $X, Y \in V_n$, then $(\alpha, \beta), (\alpha', \beta') \prec (X\alpha, Y\beta)$, so $(\alpha, \alpha'), (\beta, \beta') \prec (X\alpha, Y\beta)$. Hence, by induction $\alpha \overset{R}{\equiv} \alpha'$ and $\beta \overset{R}{\equiv} \beta'$, so $X\alpha \overset{R}{\equiv} X\alpha' R Y\beta' \overset{R}{\equiv} Y\beta$.
- If $X \in V_n$ and $Y \in V_u$, then since $\beta \equiv X_1 \dots X_n$ for some $n \geq 0$ and $YX_i \overset{R}{\equiv} Y$ for each $0 \leq i \leq n$, we find $Y\beta \overset{R}{\equiv} Y$. Furthermore, $n_f(\alpha') \leq n_f(\alpha) < n_f(X\alpha)$, so $(\alpha, \alpha') \prec (X\alpha, Y)$. Hence, by induction $\alpha \overset{R}{\equiv} \alpha'$, and since $(X\alpha', Y) \in R_2$ we find $X\alpha \overset{R}{\equiv} X\alpha' \overset{R}{\equiv} Y \overset{R}{\equiv} Y\beta$. A symmetric argument applies for the case when $X \in V_u$ and $Y \in V_n$.
- If $X, Y \in V_u$, then since $\alpha \equiv X_1 \dots X_n$ for some $n \geq 0$ and $XX_i \overset{R}{\equiv} X$ for each $0 \leq i \leq n$, we find $X\alpha \overset{R}{\equiv} X$. Similarly, we find $Y\beta \overset{R}{\equiv} Y$ and thus since $(X, Y) \in R_2$, we derive $X\alpha \overset{R}{\equiv} X \overset{R}{\equiv} Y \overset{R}{\equiv} Y\beta$. ◀

It follows from Theorem 27 that bisimilarity is semi-decidable, and since also non-bisimilarity is semi-decidable, we obtain the following corollary.

▶ **Corollary 28.** *Bisimilarity is decidable for all processes definable by means of a finite guarded recursive specification over* $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \mathcal{P})$*.*

5 Conclusion

We have considered a variant of the Theory of Sequential Processes proposed in [7] in which sequential composition is replaced by sequencing. The distinguishing feature of the resulting process theory is that it includes the notion of intermediate acceptance relevant for the

theory of automata and formal languages, without also including the complications that arise from transparency. (We should mention here that the variant of successful termination considered by Aceto and Hennessy in [1] also does not lead to transparency, but in their theory a non-deterministic choice successfully terminates only if *both* arguments successfully terminate, and hence it does not have intermediate acceptance.)

We have presented a finite axiomatisation of the ground equational theory of the recursion-free fragment of the Theory of Sequential Processes using the auxiliary operator NT and proved that a finite axiomatisation without auxiliary operators does not exist.

Processes definable by means of a finite guarded recursive specification over $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$ may rightfully be referred to as *context-free processes*. Indeed, the language of a process definable by means of a finite guarded recursive specification is context-free, and for every context-free language there is a process definable by a finite guarded recursive specification over $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$ with that language. In [7] it was already proved that every context-free process is bisimilar to a pushdown process. Here we have proved that bisimilarity is decidable for all context-free processes, extending the seminal result of Christensen, Hüttel and Stirling [13] with intermediate acceptance.

It follows from the work of Moller [19] that not every pushdown process is context-free. We conjecture that extending $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$ with propositional signals suffices to facilitate the definability of all pushdown processes. This will be the topic of a forthcoming paper.

Another interesting remaining open problem is whether bisimilarity is also decidable for the variant of the Theory of Sequential Processes discussed in [2]. In [8] it is argued that properties 2 and 3 of Proposition 21 do not hold in this case, and it seems considerably more difficult to deal with the ensuing complications.

References

1 Luca Aceto and Matthew Hennessy. Termination, Deadlock, and Divergence. *J. ACM*, 39(1):147–187, 1992. doi:10.1145/147508.147527.

2 Jos C. M. Baeten, Twan Basten, and Michel Reniers. *Process algebra: equational theories of communicating processes*, volume 50. Cambridge University Press, 2010.

3 Jos C. M. Baeten, Pieter J. L. Cuijpers, Bas Luttik, and P. J. A. van Tilburg. A Process-Theoretic Look at Automata. In Farhad Arbab and Marjan Sirjani, editors, *Proceedings of FSEN 2009*, volume 5961 of *LNCS*, pages 1–33. Springer, 2009. doi:10.1007/978-3-642-11623-0_1.

4 Jos C. M. Baeten, Pieter J. L. Cuijpers, and P. J. A. van Tilburg. A Context-Free Process as a Pushdown Automaton. In Franck van Breugel and Marsha Chechik, editors, *Proceedings of CONCUR 2008*, volume 5201 of *LNCS*, pages 98–113. Springer, 2008. doi:10.1007/978-3-540-85361-9_11.

5 Jos C. M. Baeten, Bas Luttik, Tim Muller, and Paul van Tilburg. Expressiveness modulo bisimilarity of regular expressions with parallel composition. *Mathematical Structures in Computer Science*, 26(6):933–968, 2016. doi:10.1017/S0960129514000309.

6 Jos C. M. Baeten, Bas Luttik, and Paul van Tilburg. Reactive Turing machines. *Inf. Comput.*, 231:143–166, 2013. doi:10.1016/j.ic.2013.08.010.

7 Jos C. M. Baeten, Bas Luttik, and Fei Yang. Sequential Composition in the Presence of Intermediate Termination (Extended Abstract). In Kirstin Peters and Simone Tini, editors, *Proceedings of EXPRESS/SOS 2017*, volume 255 of *EPTCS*, pages 1–17, 2017. doi:10.4204/EPTCS.255.1.

8 Astrid Belder. Decidability of bisimilarity and axiomatisation for sequential processes in the presence of intermediate termination. Master's thesis, Eindhoven University of Technology, 2018. Available from https://research.tue.nl/en/studentTheses/decidability-of-bisimilarity-and-axiomatisation-for-sequential-pr.

9 Jan A. Bergstra and Jan Willem Klop. Process Algebra for Synchronous Communication. *Information and Control*, 60(1-3):109–137, 1984. doi:10.1016/S0019-9958(84)80025-X.

10 Bard Bloom. When is Partial Trace Equivalence Adequate? *Formal Asp. Comput.*, 6(3):317–338, 1994. doi:10.1007/BF01215409.

11 Roland N. Bol and Jan Friso Groote. The Meaning of Negative Premises in Transition System Specifications. *J. ACM*, 43(5):863–914, 1996. doi:10.1145/234752.234756.

12 Olaf Burkart, Didier Caucal, Faron Moller, and Bernhard Steffen. Verification on Infinite Structures. In J.A. Bergstra, A.J. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, pages 545–623. Elsevier, 2001.

13 Søren Christensen, Hans Hüttel, and Colin Stirling. Bisimulation Equivalence is Decidable for All Context-Free Processes. *Inf. Comput.*, 121(2):143–148, 1995. doi:10.1006/inco.1995.1129.

14 Rob J. van Glabbeek. The meaning of negative premises in transition system specifications II. *J. Log. Algebr. Program.*, 60-61:229–258, 2004. doi:10.1016/j.jlap.2004.03.007.

15 Jan Friso Groote. Transition System Specifications with Negative Premises. *Theor. Comput. Sci.*, 118(2):263–299, 1993. doi:10.1016/0304-3975(93)90111-6.

16 Stephen C. Kleene. Representation of Events in Nerve Nets and Finite Automata. *Automata Studies*, pages 3–41, 1956.

17 Hans Leiß. Towards Kleene Algebra with Recursion. In Egon Börger, Gerhard Jäger, Hans Kleine Büning, and Michael M. Richter, editors, *Proceedings of CSL '91*, volume 626 of *LNCS*, pages 242–256. Springer, 1991. doi:10.1007/BFb0023771.

18 R. Milner. *Communication and Concurrency*. Prentice Hall, Englewood Cliffs, 1989.

19 Faron Moller. Infinite Results. In Ugo Montanari and Vladimiro Sassone, editors, *Proceedings of CONCUR '96*, volume 1119 of *Lecture Notes in Computer Science*, pages 195–216. Springer, 1996. doi:10.1007/3-540-61604-7_56.

20 Damien Pous and Davide Sangiorgi. Enhancements of the bisimulation proof method. In Davide Sangiorgi and Jan Rutten, editors, *Advanced Topics in Bisimulation and Coinduction*, number 52 in Cambridge Tracts in Theoretical Computer Science, pages 233—-289. Cambridge University Press, 2012.

21 Peter Thiemann. Partial Derivatives for Context-Free Languages - From μ-Regular Expressions to Pushdown Automata. In Javier Esparza and Andrzej S. Murawski, editors, *Proceedings of FOSSACS 2017*, volume 10203 of *LNCS*, pages 248–264, 2017. doi:10.1007/978-3-662-54458-7_15.

22 Chris Verhoef. A General Conservative Extension Theorem in Process Algebra. In Ernst-Rüdiger Olderog, editor, *Proceedings of PROCOMET'94*, volume A-56 of *IFIP Transactions*, pages 149–168. North-Holland, 1994.

23 Chris Verhoef. A Congruence Theorem for Structured Operational Semantics with Predicates and Negative Premises. *Nord. J. Comput.*, 2(2):274–302, 1995.

24 Jos L. M. Vrancken. The Algebra of Communicating Processes With Empty Process. *Theor. Comput. Sci.*, 177(2):287–328, 1997. doi:10.1016/S0304-3975(96)00250-2.

A Proofs of Lemmas 9 and 10

In this appendix we shall provide proofs for Lemmas 9 and 10, restated below as Lemmas 35 and 40. For the formulation of our arguments, it is convenient to associate behaviour to $\mathrm{TSP}^{:}(\mathcal{A})$-terms with variables. We assume an extended syntax in which a constant $\bar{x}$ added for every variable x and include the following operational rule to the operational semantics presented in Figure 1:

$$\frac{}{x \xrightarrow{x} \bar{x}} \ .$$

The resulting collection of operational rules is used to derive transitions of $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-terms (with variables). A transition of a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term t may then either result in another $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term, or, if it is due to the transition of a variable, it may result in a term in a syntax extended with the constant $\bar{x}$: For every variable x, we inductively define the set of $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \bar{x})$-terms as follows:

1. the constant $\bar{x}$ is a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \bar{x})$-term; and
2. if t_1 is a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \bar{x})$-term and t_2 is a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term, then $t_1 \,;\, t_2$ is a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \bar{x})$-term.

If t is a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \bar{x})$-term and p is a closed $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term, then by $t[\bar{x} := p]$ we denote the $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term obtained by replacing $\bar{x}$ by p. While every variable can now "take a step", we do not let this contribute to the width of a term, so $\mathit{width}(x) = 0$ for every variable x.

▶ **Lemma 29.** *Let t be a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term.*

1. *If $t \xrightarrow{a} t'$ for some action a, then t' is a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term.*
2. *If $t \xrightarrow{x} t'$ for some variable x, then t' is a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \bar{x})$-term.*

We would like to establish a relationship between transitions from t and transitions from $\sigma(t)$, where σ is a closed substitution. However, we cannot yet fully express that a transition originates from a substitution in a variable. For example, consider the $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term $t \equiv x \,;\, y$ and the closed substitution σ, where $\sigma(x) = \mathbf{1}$ and $\sigma(y) = a.\mathbf{1}$. Clearly, $\sigma(t) = \mathbf{1} \,;\, a.\mathbf{1}$ and hence $\sigma(t) \xrightarrow{a} \mathbf{1}$. However, we cannot express that this a-transition originates from the substitution in y, as $t \overset{y}{\nrightarrow}$. To be able to express this, we define the following substitution.

▶ **Definition 30.** *Given a substitution σ and variable x, the substitution $\mu_{\sigma x}$ is defined as:*

$$\mu_{\sigma x}(y) = \begin{cases} y & \text{if } y = x \\ \sigma(y) & \text{otherwise.} \end{cases}$$

Referring to the example preceding Definition 30, note that $\mu_{\sigma y}(t) \xrightarrow{y} \bar{y}$ and $\sigma(y) \xrightarrow{a} \mathbf{1}$. We can establish several useful relationships between $\sigma(t)$ and $\mu_{\sigma x}(t)$.

▶ **Lemma 31.** *Let t be a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term, σ a closed substitution, x a variable, p a closed $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term and a an action such that $\sigma(x) \xrightarrow{a} p$. Then:*

1. *if $\sigma(t) \not\downarrow$, then $\mu_{\sigma x}(t) \not\downarrow$;*
2. *if $\sigma(t) \nrightarrow$, then $\mu_{\sigma x}(t) \nrightarrow$;*
3. *if $\sigma(t) \nrightarrow$ and $\sigma(t) \downarrow$, then $\mu_{\sigma x}(t) \downarrow$.*

In the following lemma it is proven that if t contains a subterm t_2 such that $\mathit{width}(\sigma(t_2)) > \mathit{width}(t_2)$, then one of the actions that can be executed by $\sigma(t_2)$ must come from a substitution in some variable x.

▶ **Lemma 32.** *Let t be a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term and σ a closed substitution. If $\mathit{width}(\sigma(t)) > \mathit{width}(t)$, then there must exist an action a, closed $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-terms p and p', a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A}, \bar{x})$-term t', and a variable x such that $\sigma(t) \xrightarrow{a} p$, $\mu_{\sigma x}(t) \xrightarrow{x} t'$, $\sigma(x) \xrightarrow{a} p'$ and $p \equiv \sigma(t'[\bar{x} := p'])$.*

▶ **Lemma 33.** *Let t be a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term, x a variable and σ a closed substitution. Then:*

1. *if $t \downarrow$, then $\sigma(t) \downarrow$;*
2. *if $\sigma(t) \downarrow$, then $\vartheta_{(\sigma,x)}(t) \downarrow$;*
3. *if $\vartheta_{(\sigma,x)}(\mu_{\sigma x}(t)) \downarrow$, then $\vartheta_{(\sigma,x)}(t) \downarrow$.*

Using these properties we show that given a term t, variable x and substitution σ as described above, $\vartheta_{(\sigma,x)}(t) \downarrow$ indeed holds.

▶ **Lemma 34.** *Let t be a* TSP$^{\mathrm{i}}(\mathcal{A})$*-term, let x be a variable, and let σ be a substitution. If there exist a* TSP$^{\mathrm{i}}(\mathcal{A}, \bar{x})$*-term t' and a closed* TSP$^{\mathrm{i}}(\mathcal{A})$*-term p such that $t \xrightarrow{x} t'$, $\sigma(x) \xrightarrow{a} p$ and $\sigma(t'[\bar{x} := p]) \downarrow$, then $\vartheta_{(\sigma,x)}(t)\downarrow$.*

By utilizing the results from Lemma 32 and Lemma 34, we show that given a TSP$^{\mathrm{i}}(\mathcal{A})$-term t and substitution σ, if $\Psi_n(\sigma(t))$ holds, then t must contain some variable x such that $\vartheta_{(\sigma,x)}(t)\downarrow$ and either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{\tilde{b}}$.

▶ **Lemma 35.** *Let t be a* TSP$^{\mathrm{i}}(\mathcal{A})$*-term and let n be a natural number such that $\mathit{width}(t') < n$ for every subterm t' of t. If $\Psi_n(\sigma(t))$ for some closed substitution σ, then there is a variable x such that $\vartheta_{(\sigma,x)}(t)\downarrow$ and either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{\tilde{b}}$.*

Proof. We proceed with induction on the structure of t.

- If $t \equiv \mathbf{0}$, $t \equiv \mathbf{1}$ or $t \equiv a.t'$ for some action a and TSP$^{\mathrm{i}}(\mathcal{A})$-term t', then $\sigma(t)$ cannot have a summand of the form $t_1 \,;\, t_2$, so $\Psi_n(\sigma(t))$ does not hold for any substitution σ. Hence the implication vacuously holds.
- Let $t \equiv y$ for some variable y, and suppose that $\Psi_n(\sigma(t))$ holds for some closed substitution σ. Then clearly from $\Psi_n(\sigma(t))$ it follows that $\Psi_n(\sigma(y))$. Furthermore, since $t \xrightarrow{y} \bar{y}$ and since $\sigma(\bar{y}[\bar{y} := \mathbf{1}]) \equiv \mathbf{1}$, we have that $\sigma(\bar{y}[\bar{y} := \mathbf{1}]) \downarrow$. Hence, by Lemma 34, we have that $\vartheta_{(\sigma,y)}(t)\downarrow$ and thus $x = y$.
- Let $t \equiv t_1 + t_2$ for some TSP$^{\mathrm{i}}(\mathcal{A})$-terms t_1 and t_2. If $\Psi_n(\sigma(t))$, then either $\sigma(t_1)$ or $\sigma(t_2)$ must contain a summand p such that one of the three cases of the definition of Ψ_n applies. We proceed to consider the case that p is a summand of $\sigma(t_1)$; the proof in the case that p is a summand of $\sigma(t_2)$ proceeds analogously. Note that, since $p \leftrightarrows (\tilde{a}.\mathbf{1}+\mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1}+\mathbf{1})^i)$, we find that $\sigma(t_1) \xrightarrow{\tilde{a}} p'$ with $p' \leftrightarrows \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1}+\mathbf{1})^i)$. Moreover, since $\sigma(t_1)$ is a summand of $\sigma(t)$ and also $\sigma(t) \leftrightarrows (\tilde{a}.\mathbf{1}+\mathbf{1}) ; \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} ; (\tilde{b}.\mathbf{1}+\mathbf{1})^i)$, we find that $\sigma(t_1) \leftrightarrows (\tilde{a}.\mathbf{1}+\mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1}+\mathbf{1})^i)$, and hence $\Psi_n(\sigma(t_1))$. Since every subterm of t_1 is a subterm of t, we also have that $\mathit{width}(t') < n$ for every subterm t' of t_1. Therefore, we may now apply the induction hypothesis to conclude that either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{b}$, and $\vartheta_{(\sigma,x)}(t_1)\downarrow$; clearly, from the latter it follows that $\vartheta_{(\sigma,x)}(t)\downarrow$.
- Let $t \equiv t_1 \,;\, t_2$ for some TSP$^{\mathrm{i}}(\mathcal{A})$-terms t_1 and t_2, and suppose that $\Psi_n(\sigma(t))$. Then, considering the definition of Ψ_n, one of the following three cases must apply:
  1. If $\Phi_n(\sigma(t_1) \,;\, \sigma(t_2))$, then $\sigma(t_1) \leftrightarrows \tilde{a}.+\mathbf{1}$ and $\sigma(t_2) \leftrightarrows \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1}+\mathbf{1})^i)$. Then $\sigma(t_2) \xrightarrow{\tilde{b}} (\tilde{b}.\mathbf{1}+\mathbf{1})^i$ for all $1 \leq i \leq n$. Clearly, if $i \neq j$, then $(\tilde{b}.\mathbf{1}+\mathbf{1})^i \not\leftrightarrows (\tilde{b}.\mathbf{1}+\mathbf{1})^j$, so $\mathit{width}(\sigma(t_2)) \geq n > \mathit{width}(t_2)$. It follows by Lemma 32 that there exist an action a, closed TSP$^{\mathrm{i}}(\mathcal{A})$-terms p and p', a TSP$^{\mathrm{i}}(\mathcal{A}, \bar{x})$-term t' and a variable x such that $\sigma(t_2) \xrightarrow{a} p$, $\mu_{\sigma x}(t_2) \xrightarrow{x} t'$, $\sigma(x) \xrightarrow{a} p'$ and $p \equiv \sigma(t'[\bar{x} := p'])$. Clearly, we must have $a = \tilde{b}$ and $p \leftrightarrows (\tilde{b}.\mathbf{1}+\mathbf{1})^i$ for some $1 \leq i \leq n$. To see that $\vartheta_{(\sigma,x)}(t)\downarrow$, note that, since $\sigma(t_1) \leftrightarrows \tilde{a}.\mathbf{1}+\mathbf{1}$, we have that $\sigma(t_1) \downarrow$ and hence $\vartheta_{(\sigma,x)}(t_1)\downarrow$. Moreover, since $\sigma(t'[\bar{x} := p']) \leftrightarrows (\tilde{b}.\mathbf{1}+\mathbf{1})^i$, we find that $\sigma(t'[\bar{x} := p']) \downarrow$, and hence, by Lemma 34, we get that $\vartheta_{(\sigma,x)}(\mu_{\sigma x}(t_2))\downarrow$. Finally, by Lemma 33(3), we conclude that $\vartheta_{(\sigma,x)}(t_2)\downarrow$ and thus $\vartheta_{(\sigma,x)}(t)\downarrow$
  2. If $\sigma(t_1) \leftrightarrows \mathbf{1}$ and $\Psi_n(\sigma(t_2))$, then since every subterm of t_2 is a subterm of t we find that $\mathit{width}(t_2') < n$ for all subterms t_2' of t_2. Hence, by the induction hypothesis, for some variable x we have that either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{\tilde{b}}$ and, moreover, $\vartheta_{(\sigma,x)}(t_2)\downarrow$. From $\sigma(t_1) \leftrightarrows \mathbf{1}$ it follows that $\sigma(t_1) \downarrow$, so, by Lemma 33(2), $\vartheta_{(\sigma,x)}(t_1)\downarrow$, and hence $\vartheta_{(\sigma,x)}(t)\downarrow$.

3. If $\Psi_n(\sigma(t_1))$ and $\sigma(t_2) \leftrightarrows \mathbf{1}$, then the proof that $\vartheta_{(\sigma,x)}(t)\downarrow$ is analogous to the previous case. ◀

We have established that if $\Psi_n(\sigma(t))$ holds for some substitution σ and $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term t such that $\mathit{width}(t') < n$ for each subterm t' of t, then t must confirm to certain properties. Now, for any term u such that $t \leftrightarrows u$, these properties must be valid as well. Hence, it remains to show is that if u contains these properties, then $\Psi_n(\sigma(u))$ must hold as well. This is shown in Lemma 40. In order to prove this result, some useful properties are established in Lemma 36 to Lemma 39.

▶ **Lemma 36.** *Let p and q be closed $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-terms and suppose that $p \leftrightarrows q$. Then $\mathit{depth}(p) = \mathit{depth}(q)$.*

Proof. Assume that $p \leftrightarrows q$ and, for the sake of contradiction, suppose that $\mathit{depth}(p) = n$ and $\mathit{depth}(q) = m$, for some $n > m$. Then, by definition, $p \longrightarrow^n p'$ and since $p \leftrightarrows q$, $q \longrightarrow^n q'$, such that $p' \leftrightarrows q'$. Clearly, since $n > m$, this contradicts $\mathit{depth}(q) = m$. Hence, we conclude $\mathit{depth}(p) = \mathit{depth}(q)$. ◀

▶ **Lemma 37.** *For all closed $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-terms p_1 and p_2, if $p_1 \,;\, p_2 \leftrightarrows \tilde{a}.\mathbf{1} \,;\, \sum_{i=1}^{n} \tilde{b}.(\tilde{b}.\mathbf{1} + \mathbf{1})^i$, then one of the following cases must hold:*

1. *$p_1 \leftrightarrows \mathbf{1}$ and $p_2 \leftrightarrows \tilde{a}.\mathbf{1} \,;\, \sum_{i=1}^{n} \tilde{b}.(\tilde{b}.\mathbf{1} + \mathbf{1})^i$; or*
2. *$p_1 \leftrightarrows \tilde{a}.\mathbf{1}$ and $p_2 \leftrightarrows \sum_{i=1}^{n} \tilde{b}.(\tilde{b}.\mathbf{1} + \mathbf{1})^i$; or*
3. *$p_1 \leftrightarrows \tilde{a}.\mathbf{1} + \mathbf{1}$ and $p_2 \leftrightarrows \tilde{a}.\mathbf{1} \,;\, \sum_{i=1}^{n} \tilde{b}.(\tilde{b}.\mathbf{1} + \mathbf{1})^i$; or*
4. *$p_1 \leftrightarrows \tilde{a}.\mathbf{1} \,;\, \sum_{i=1}^{n} \tilde{b}.(\tilde{b}.\mathbf{1} + \mathbf{1})^i$ and $p_2 \leftrightarrows \mathbf{1}$.*

▶ **Lemma 38.** *For any $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term t, variable x and closed substitution σ, if $\sigma(t) \not\downarrow$ and $\vartheta_{(\sigma,x)}(t)\downarrow$, then t contains x.*

Proof. Let t be a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term, x a variable and σ a closed substitution such that $\sigma(t) \not\downarrow$ and $\vartheta_{(\sigma,x)}(t)\downarrow$. Now suppose t does not contain x. Then, by the definition of $\vartheta_{(\sigma,x)}$, $\vartheta_{(\sigma,x)}(t) \equiv \sigma(t)$, which means that if $\sigma(t) \not\downarrow$ we should also have $\vartheta_{(\sigma,x)}(t) \not\downarrow$. Since this contradicts $\vartheta_{(\sigma,x)}(t)\downarrow$, we conclude t must contain x. ◀

▶ **Lemma 39.** *For any $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term t, variable x and closed substitution σ, if t contains x and $\sigma(x) \xrightarrow{a_1 \ldots a_n} p$ for some sequence of actions $a_1 \ldots a_n$ and closed $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term p, then*

1. *either $\sigma(t) \longrightarrow^* p' \xrightarrow{a_1 \ldots a_n} p''$, for some p' and p'',*
2. *or $\sigma(t) \longrightarrow^* p'$, for some p' such that $p' \leftrightarrows \mathbf{0}$.*

Proof. Let t be a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term, x a variable and σ a closed substitution such that t contains x and $\sigma(x) \xrightarrow{a_1 \ldots a_n} p$ for some sequence of actions $a_1 \ldots a_n$ and closed $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term p. It can then be proved with induction on the structure of t that one of the two cases of the lemma must hold. ◀

▶ **Lemma 40.** *For any $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term t, variable x and closed substitution σ, if $\sigma(t) \leftrightarrows (\tilde{a}.\mathbf{1} + \mathbf{1}) \,;\, \sum_{i=1}^{n} (\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$, $\vartheta_{(\sigma,x)}(t)\downarrow$ and either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{\tilde{b}}$, then $\Psi_n(\sigma(t))$.*

Proof. Let t be a $\mathrm{TSP}^{\mathrm{i}}(\mathcal{A})$-term, x a variable and let σ be closed substitution such that $\sigma(t) \leftrightarrows (\tilde{a}.\mathbf{1} + \mathbf{1}) \,;\, \sum_{i=1}^{n} (\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$, $\vartheta_{(\sigma,x)}(t)\downarrow$ and either $\Psi_n(\sigma(x))$ or $\sigma(x) \xrightarrow{b}$. We prove by induction on the structure of t that $\Psi_n(\sigma(t))$ must hold.

- If $t \equiv \mathbf{0}$ or $t \equiv \mathbf{1}$, then $\sigma(t) \not\leftrightarrows (\tilde{a}.\mathbf{1} + \mathbf{1}) \,;\, \sum_{i=1}^{n} (\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$, hence, the implication vacuously holds.

- If $t \equiv a.t'$, then $\vartheta_{(\sigma,x)}(t)\not\downarrow$ which contradicts $\vartheta_{(\sigma,x)}(t)\downarrow$, hence, the implication vacuously holds.
- If $t \equiv y$ for some variable y, then since $\sigma(t) \leftrightarroweq (\tilde{a}.\mathbf{1}+\mathbf{1});\sum_{i=1}^{n}(\tilde{b}.\mathbf{1};(\tilde{b}.\mathbf{1}+\mathbf{1})^i)$, we must have $\sigma(t)\not\downarrow$. Moreover, since $\vartheta_{(\sigma,x)}(t)\downarrow$, by Lemma 38, $\sigma(t)$ contains x, thus we must have $y \equiv x$. Now suppose $\sigma(x) \xrightarrow{\tilde{b}}$, then $\sigma(t) \xrightarrow{\tilde{b}}$, contradicting $\sigma(t) \leftrightarroweq (\tilde{a}.\mathbf{1}+\mathbf{1});\sum_{i=1}^{n}(\tilde{b}.\mathbf{1};(\tilde{b}.\mathbf{1}+\mathbf{1})^i)$. Hence, it must be the case that $\Psi_n(\sigma(x))$ holds and thus also $\Psi_n(\sigma(t))$ holds.
- Suppose $t \equiv t_1 + t_2$ and suppose that the lemma holds for t_1 and t_2 (IH). Since $\sigma(t) \leftrightarroweq (\tilde{a}.\mathbf{1} + \mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$, we have $\sigma(t)\not\downarrow$ and thus both $\sigma(t_1)\not\downarrow$ and $\sigma(t_2)\not\downarrow$. Moreover, since $\vartheta_{(\sigma,x)}(t)\downarrow$ either $\vartheta_{(\sigma,x)}(t_1)\downarrow$ or $\vartheta_{(\sigma,x)}(t_2)\downarrow$. Without loss of generality assume $\vartheta_{(\sigma,x)}(t_1)\downarrow$. Then, by Lemma 38, t_1 must contain x. Since either $\sigma(x) \xrightarrow{\tilde{b}}$ or $\Psi_n(\sigma(x))$ and thus $\sigma(x) \xrightarrow{\tilde{a}}$, by Lemma 39, either $\sigma(t_1) \longrightarrow^* q_1 \xrightarrow{a}$ for some action a and closed TSP$^{\mathrm{i}}(\mathcal{A})$-term q_1, or $\sigma(t_1) \longrightarrow^* q_2$ for some closed TSP$^{\mathrm{i}}(\mathcal{A})$-term q_2 such that $q_2 \leftrightarroweq \mathbf{0}$. The second case clearly contradicts $\sigma(t) \leftrightarroweq (\tilde{a}.\mathbf{1}+\mathbf{1});\sum_{i=1}^{n}(\tilde{b}.\mathbf{1};(\tilde{b}.\mathbf{1}+\mathbf{1})^i)$. Hence, it must be the case that $\sigma(t_1) \longrightarrow^* q_1 \xrightarrow{\tilde{a}}$. Since $\sigma(t_1)$ is able to execute an action and $\sigma(t_1)$ is a summand of $\sigma(t)$, we must have $\sigma(t_1) \xrightarrow{a} p$ such that $p \leftrightarroweq \sum_{i=1}^{n}(\tilde{b}.\mathbf{1};(\tilde{b}.\mathbf{1}+\mathbf{1})^i)$. Hence, we must have $\sigma(t_1) \leftrightarroweq (\tilde{a}.\mathbf{1}+\mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1}+\mathbf{1})^i)$, and thus by the induction hypothesis we conclude $\Psi_n(\sigma(t_1))$, and since $\sigma(t_1)$ is a summand of $\sigma(t)$ also $\Psi_n(\sigma(t))$.
- Suppose $t \equiv t_1 \,;\, t_2$ and suppose that the lemma holds for t_1 and t_2 (IH). Since $\sigma(t) = \sigma(t_1 \,;\, t_2) = \sigma(t_1) \,;\, \sigma(t_2)$, we have $\sigma(t_1) \,;\, \sigma(t_2) \leftrightarroweq (\tilde{a}.\mathbf{1}+\mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1}+\mathbf{1})^i)$ and thus by Lemma 37, one of the following cases must hold:

1. If $\sigma(t_1) \leftrightarroweq \mathbf{1}$ and $\sigma(t_2) \leftrightarroweq (\tilde{a}.\mathbf{1} + \mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$, then, by the induction hypothesis, $\Psi_n(\sigma(t_2))$. Moreover, since $\sigma(t_1) \leftrightarroweq \mathbf{1}$, by case 2 of Ψ_n we conclude $\Psi_n(\sigma(t))$.
2. If $\sigma(t_1) \leftrightarroweq \tilde{a}.\mathbf{1}$ and $\sigma(t_2) \leftrightarroweq \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$, then $\sigma(t_1)\not\downarrow$. Moreover, since $\vartheta_{(\sigma,x)}(t)\downarrow$ we must have $\vartheta_{(\sigma,x)}(t_1)\downarrow$, and thus by by Lemma 38, t_1 must contain x. We distinguish two cases.
   If $\Psi_n(\sigma(x))$, then $\sigma(x) \leftrightarroweq (\tilde{a}.\mathbf{1} + \mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$ and, by Lemma 39, either $\sigma(t_1) \longrightarrow^* q_1 \xrightarrow{\tilde{a}} q_1' \xrightarrow{\tilde{b}} q_1''$ for some closed TSP$^{\mathrm{i}}(\mathcal{A})$-terms q_1, q_1' and q_1'', or $\sigma(t_1) \longrightarrow^* q_2$ for some closed TSP$^{\mathrm{i}}(\mathcal{A})$-term q_2 such that $q_2 \leftrightarroweq \mathbf{0}$. Both cases clearly contradict $\sigma(t_1) \leftrightarroweq \tilde{a}.\mathbf{1}$.
   If $\sigma(x) \xrightarrow{b}$, then, by Lemma 39, either $\sigma(t_1) \longrightarrow^* q_1 \xrightarrow{\tilde{b}}$ for some closed TSP$^{\mathrm{i}}(\mathcal{A})$-term q_1, or $\sigma(t_1) \longrightarrow^* q_2$ for some closed TSP$^{\mathrm{i}}(\mathcal{A})$-term q_2 such that $q_2 \leftrightarroweq \mathbf{0}$. Again, both cases contradict $\sigma(t_1) \leftrightarroweq \tilde{a}.\mathbf{1}$, hence the case where $\sigma(t_1) \leftrightarroweq \tilde{a}.\mathbf{1}$ and $\sigma(t_2) \leftrightarroweq (\tilde{a}.\mathbf{1} + \mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$ can never occur.
3. If $\sigma(t_1) \leftrightarroweq \tilde{a}.\mathbf{1} + \mathbf{1}$ and $\sigma(t_2) \leftrightarroweq \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$, then clearly $\Phi_n(\sigma(t_1) \,;\, \sigma(t_2))$ and thus, by case 1 of Ψ_n we conclude $\Psi_n(\sigma(t))$.
4. If $\sigma(t_1) \leftrightarroweq (\tilde{a}.\mathbf{1} + \mathbf{1}) \,;\, \sum_{i=1}^{n}(\tilde{b}.\mathbf{1} \,;\, (\tilde{b}.\mathbf{1} + \mathbf{1})^i)$ and $\sigma(t_2) \leftrightarroweq \mathbf{1}$, then, by the induction hypothesis, $\Psi_n(\sigma(t_1))$. Moreover, since $\sigma(t_2) \leftrightarroweq \mathbf{1}$, by case 3 of Ψ_n we conclude $\Psi_n(\sigma(t))$. ◀

B Proof of Proposition 21

The three properties of Proposition 21 are proved below as Lemmas 45, 46 and 47. Throughout this appendix it will be assumed that Δ is in AIGNF.

Let us first establish that then all Δ-reachable sequences are acceptance irredundant.

▶ **Lemma 41.** *For every sequence $\alpha \in \overline{\mathcal{P}}^*_{\not\downarrow}$, if $\alpha \xrightarrow{a} \alpha'$, then $\alpha' \in \overline{\mathcal{P}}^*_{\not\downarrow}$.*

▶ **Lemma 42.** *For every acceptance irredundant sequence $\alpha\beta$, we have that either $\alpha \in \overline{\mathcal{P}}^*_{\not\downarrow}$ and $\beta \in \overline{\mathcal{P}}^*_{\not\downarrow}\overline{\mathcal{P}}_{\not\downarrow}\mathcal{P}^*_{\downarrow}$, or $\alpha \in \overline{\mathcal{P}}^*_{\not\downarrow}\mathcal{P}_{\not\downarrow}{\mathcal{P}_{\downarrow}}^* \cup {\mathcal{P}_{\downarrow}}^*$ and $\beta \in \mathcal{P}^*_{\downarrow}$.*

Using the previous lemma, we show that acceptance irredundant sequences maintain their shape when executing an action.

▶ **Lemma 43.** *If α is acceptance irredundant and $\alpha \xrightarrow{a} \alpha'$, then α' is acceptance irredundant.*

▶ **Corollary 44.** *If Δ is in AIGNF, then all Δ-reachable sequences are acceptance irredundant.*

▶ **Lemma 45.** *Suppose that Δ is in AIGNF and let $\alpha, \beta \in \mathcal{P}^*$. If $\alpha\beta$ is acceptance irredundant and α is unnormed, then $\alpha\beta \leftrightarrows \alpha$.*

Proof. We prove that the relation

$$R = \{(\alpha, \alpha\beta) \mid \alpha\beta \text{ is acceptance irredundant and } \alpha \text{ is unnormed}\}$$

is a bisimulation relation.

If $\alpha \xrightarrow{a} \alpha'$, then $\alpha\beta \xrightarrow{a} \alpha'\beta$ and since α is unnormed, so is α'. Moreover, since $\alpha\beta$ is acceptance irredundant, we have by Lemma 43 that $\alpha'\beta$ is acceptance irredundant and hence $(\alpha', \alpha'\beta) \in R$. Furthermore, if $\alpha\downarrow$, then $\alpha \in {\mathcal{P}_{\downarrow}}^*$, hence, since $\alpha\beta$ is acceptance irredundant, we have $\beta \in {\mathcal{P}_{\downarrow}}^*$ and thus $\alpha\beta\downarrow$. A symmetric argument applies for the cases where $\alpha\beta \xrightarrow{a} \alpha'\beta$ and $\alpha\beta\downarrow$. ◀

▶ **Lemma 46.** *Suppose that Δ is in AIGNF. Then for all acceptance irredundant sequences α we have $|\alpha| \leq n(\alpha)$.*

Proof. If α contains an unnormed process identifier, then clearly α is unnormed hand hence $|\alpha| \leq n(\alpha) = \infty$. So, suppose that α is normed. Then all process identifiers in α are normed and must have a defining equation of the shape $\sum_{i=1}^{n} a_i.\alpha_i(+\mathbf{1})$ for some $n \in \mathbb{N}^+$ with α_i acceptance irredundant. Since every variable must be able to execute at least one action, the norm of each variable must be greater or equal than 1. Hence, in this case also $|\alpha| \leq n(\alpha)$. ◀

▶ **Lemma 47.** *Suppose that Δ is in AIGNF and let $\alpha\gamma, \beta\gamma \in \mathcal{P}^*$ be acceptance irredundant. If α, β and γ are normed, then $\alpha\gamma \leftrightarrows \beta\gamma$ implies $\alpha \leftrightarrows \beta$.*

Proof. It suffices to prove that

$$R = \{(\alpha, \beta) \mid \exists\gamma.\ \alpha\gamma \leftrightarrows \beta\gamma \text{ and } \alpha\gamma,\ \beta\gamma \text{ are acceptance irredundant}\}$$

is a bisimulation relation.

Suppose $\alpha \xrightarrow{a} \alpha'$, then $\beta\gamma \xrightarrow{a} \delta$ such that $\delta \leftrightarrows \alpha'\gamma$. We distinguish two cases.

- If $\beta \xrightarrow{a} \beta'$ and $\delta = \beta'\gamma$, then since $\alpha\gamma$ and $\beta\gamma$ are acceptance irredundant, by Lemma 43, $\alpha'\gamma$ and $\beta'\gamma$ are acceptance irredundant and therefore $(\alpha', \beta') \in R$.
- If $\beta\downarrow$, $\beta \not\to$, $\gamma \xrightarrow{a} \gamma'$ and $\delta = \gamma'$, then $n(\beta\gamma) = n(\gamma) < n(\alpha\gamma)$, contradicting $\alpha\gamma \leftrightarrows \beta\gamma$. Hence, in this case the implication vacuously holds.

Moreover, if $\alpha\downarrow$, then $\alpha \in {\mathcal{P}_{\downarrow}}^*$ and since $\alpha\gamma$ is acceptance irredundant also $\gamma \in {\mathcal{P}_{\downarrow}}^*$. Hence, we have $\alpha\gamma\downarrow$ and thus $\beta\gamma\downarrow$ and $\beta\downarrow$. A symmetric argument applies for the cases where $\beta \xrightarrow{a} \beta'$ and $\beta\downarrow$. ◀