Expressiveness modulo bisimilarity of regular expressions with parallel composition

JOS C. M. BAETEN, BAS LUTTIK, TIM MULLER and PAUL VAN TILBURG

Mathematical Structures in Computer Science / Volume 26 / Special Issue 06 / September 2016, pp 933 - 968
DOI: 10.1017/S0960129514000309, Published online: 02 January 2015

Math. Struct. in Comp. Science (2016), *vol.* 26, *pp.* 933–968. © Cambridge University Press 2015
doi:10.1017/S0960129514000309 First published online 2 January 2015

Expressiveness modulo bisimilarity of regular expressions with parallel composition

JOS C. M. BAETEN[†,‡], BAS LUTTIK[‡,§], TIM MULLER[¶]
and PAUL VAN TILBURG[‡]

[†]*CWI, P.O. Box 94079, NL-1090 GB, the Netherlands*
Email: Jos.Baeten@cwi.nl
[‡]*Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB Eindhoven, the Netherlands*
Email: s.p.luttik@tue.nl, paul@luon.net
[§]*Department of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1081, NL-1081 HV Amsterdam, the Netherlands*
[¶]*Faculty of Science, Technology and Communication, University of Luxembourg, 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg*
Email: tim.muller@uni.lu

Received 14 March 2011; revised 20 February 2013

The languages accepted by finite automata are precisely the languages denoted by regular expressions. In contrast, finite automata may exhibit behaviours that cannot be described by regular expressions up to bisimilarity. In this paper, we consider extensions of the theory of regular expressions with various forms of parallel composition and study the effect on expressiveness. First we prove that adding pure interleaving to the theory of regular expressions strictly increases its expressiveness modulo bisimilarity. Then, we prove that replacing the operation for pure interleaving by ACP-style parallel composition gives a further increase in expressiveness, still insufficient, however, to facilitate the expression of all finite automata up to bisimilarity. Finally, we prove that the theory of regular expressions with ACP-style parallel composition and encapsulation is expressive enough to express all finite automata up to bisimilarity. Our results extend the expressiveness results obtained by Bergstra, Bethke and Ponse for process algebras with (the binary variant of) Kleene's star operation.

1. Introduction

A well-known theorem by Kleene states that the languages accepted by finite automata are precisely the languages denoted by regular expressions (see, e.g., the textbook by Hopcroft *et al.* (2006)). Milner (1984) showed that how regular expressions can be used to describe *behaviour* by directly associating finite automata with them. He then observed that the process-theoretic counterpart of Kleene's theorem – stating that every finite automaton is described by a regular expression – fails: there exist finite automata whose behaviours cannot faithfully, i.e., up to bisimilarity, be described by regular expressions. Baeten *et al.* (2007) found a structural property on finite automata that characterizes those that are denoted with a regular expression modulo bisimilarity. In this paper, we study to what extent the expressiveness of regular expressions increases when various forms of parallel composition are added.

Our first result is to show that adding an operation for pure interleaving to regular expressions strictly increases their expressiveness modulo bisimilarity. A crucial step in our proof consists of characterizing the strongly connected components in finite automata denoted by regular expressions. The characterization allows us to prove a property pertaining to the exit transitions from such strongly connected components. If interleaving is added, then it is possible to denote finite automata violating this property.

Our second result is to show that replacing the operation for pure interleaving by ACP-style parallel composition of Bergstra and Klop (1984), which implements a form of synchronization by communication between components, leads to a further increase in expressiveness. To this end, we first characterize the strongly connected components in finite automata denoted by regular expressions with interleaving, and deduce a property on the exit transitions from such strongly connected components. Then, we present an expression in the theory of regular expressions with ACP-style parallel composition that denotes a finite automaton violating this property.

Our third result is to show that the theory of regular expressions extended with ACP-style parallel composition, but not including encapsulation, is not expressive enough to denote all finite automata.

Our fourth result is to establish that adding ACP-style parallel composition and encapsulation to the theory of regular expressions actually yields a theory in which every finite automaton can be expressed up to isomorphism, and hence, since bisimilarity is coarser than isomorphism, also up to bisimilarity. Every expression in the resulting theory, in turn, denotes a finite automaton, so this result can be thought of as an alternative process-theoretic counterpart of Kleene's theorem.

The results in this paper are inspired by the results of Bergstra *et al.* (1994) pertaining to the relative expressiveness of process algebras with a binary variant of Kleene's star operation. They establish an expressiveness hierarchy on the extensions of the process theories BPA($\mathcal{A}$), BPA$_\delta$($\mathcal{A}$), PA($\mathcal{A}$), PA$_\delta$($\mathcal{A}$), ACP($\mathcal{A}, \gamma$), and ACP$_\tau$($\mathcal{A}, \gamma$) with binary Kleene star. The reason that their results are based on extensions with the binary version of the Kleene star is that they want to avoid the process-theoretic complications arising from the notion of intermediate termination (a state in a finite automaton is *intermediately terminating* if it is terminating but also admits a transition). Most of the expressiveness results of Bergstra *et al.* (1994) are included in Bergstra *et al.* (2001), with more elaborate proofs.

For a comparison of our results with the results of Bergstra *et al.* (1994) and Bergstra *et al.* (2001), we first cast our contributions in process-theoretic terminology: regular expressions are the expressions of $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$, regular expressions extended with pure interleaving are the expressions of $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$, regular expressions extended with ACP-style parallel composition, but without encapsulation, are the expressions of $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$, and regular expressions extended with ACP-style parallel composition and encapsulation are the expressions of $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A}, \gamma)$. Our results then establish a strict relative expressiveness hierarchy on the process theories $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$, $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$, $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ and $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A}, \gamma)$, modulo bisimilarity.

The differences between the process theories $\mathrm{BPA}_\delta(\mathcal{A})$, $\mathrm{PA}_\delta(\mathcal{A})$ and $\mathrm{ACP}(\mathcal{A},\gamma)$ considered by Bergstra *et al.* (1994) and Bergstra *et al.* (2001) on the one hand, and the process theories $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ and $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ considered in this paper on the other hand are as follows: we write **0** for the constant deadlock which is denoted by δ in Bergstra *et al.* (1994, 2001), we include the unary Kleene star instead of its binary variant, and we include a constant **1** denoting the successfully terminated process. The first difference is, of course, cosmetic, and with the addition of the constant **1** the unary and binary variants of Kleene's star are interdefinable. So, our results pertaining to the relative expressiveness of $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ and $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ extend the expressiveness hierarchy of Bergstra *et al.* (1994) and Bergstra *et al.* (2001) with the constant **1**.

The paper is organized as follows. In Section 2, we present the process theory $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$, and its subtheories $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, and $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. In Section 3, we provide a characterization of the strongly connected components in the subtheories $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ and $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. In Section 4, we prove that $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is more expressive than $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. In Section 5 we prove that $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$, by which we denote $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ without encapsulation, is more expressive than $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. In Section 6, we prove that $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ is more expressive than $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$. In Section 7 we prove that every finite automaton is denoted, up to isomorphism, by an $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expression. We end the paper in Section 8 with some conclusions.

An extended abstract of this article without proofs and without the results of Section 6 has already appeared in the proceedings of EXPRESS 2010 edited by Fröschle and Valencia (2010).

2. Regular expressions with $\mathrm{ACP}(\mathcal{A},\gamma)$-style parallel composition and encapsulation

In this section, we present the relevant definitions for the process theory $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ and its subtheories $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, and $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. We give their syntax and operational semantics, and the notion of (strong) bisimilarity. We also introduce some auxiliary technical notions that we need in the remainder of the paper. The expressions of the process theory $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ are precisely the well-known regular expressions, but we shall consider the automata associated with them, modulo bisimilarity instead of modulo language equivalence.

The process theory $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ is parameterized by a non-empty set $\mathcal{A}$ of *actions*, and a *communication function* γ on $\mathcal{A}$, i.e., an associative and commutative binary partial operation $\gamma : \mathcal{A}\times\mathcal{A} \rightharpoonup \mathcal{A}$. $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ incorporates a form of synchronization between the components of a parallel composition by allowing certain actions to engage in a *communication* resulting in another action. The communication function γ then defines which actions may communicate and what is the result. The details of this feature will become clear when we present the operational semantics of parallel composition.

The set of $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions $\mathcal{P}_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}$ is generated by the following grammar:

$$p ::= \mathbf{0} \mid \mathbf{1} \mid a \mid p\cdot p \mid p+p \mid p^* \mid p\parallel p \mid \partial_H(p),$$

with a ranging over $\mathcal{A}$ and H ranging over subsets of $\mathcal{A}$.

The constants **0** and **1** respectively stand for the deadlocked process and the successfully terminated process, and the constants $a \in \mathcal{A}$ denote processes of which the only behaviour is to execute the action a and then successfully terminate. An expression of the form $p \cdot q$ is called a *sequential composition*, an expression of the form $p + q$ is called an *alternative composition*, and an expression of the form p^* is called a *star expression*. An expression of the form $p \parallel q$ is called a *parallel composition*, and an expression of the form $\partial_H(p)$ is called an *encapsulation*.

The process theory ACP$(\mathcal{A}, \gamma)$ (excluding the constants **0** and **1**, but including a constant δ with exactly the same behaviour as **0**, and without the operation *) originates with Bergstra and Klop (1984). The extension of ACP$(\mathcal{A}, \gamma)$ with a constant denoting the successfully terminated process was investigated by Koymans and Vrancken (1985), Baeten and van Glabbeek (1987), Vrancken (1997), and by Baeten *et al.* (2010) (in the first three works, the constant was denoted ε). The extension of ACP$(\mathcal{A}, \gamma)$ with the binary version of the Kleene star was first proposed by Bergstra *et al.* (1994). The reader already familiar with the process theory $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ will have noticed that the operations $\lfloor\!\lfloor$ (*left merge*) and $|$ (*communication merge*) are missing from our syntax definition. Bergstra and Klop (1984) included these operations as auxiliary operations necessary for a finite axiomatization of the theory. They do not, however, add expressiveness in our setting with Kleene star instead of a general form of recursion (see Appendix A for the proof of this fact); we have omitted them to achieve a more efficient presentation of our results.

From the names for the constructions in the syntax of $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$, the reader probably has already an intuitive understanding of the behaviour of the corresponding processes. The operational rules listed in Table 1 formalize the operational behaviour in the style of structural operational semantics (Plotkin 2004). Note how the communication function in rule 15 is employed to model a form of communication between parallel components: if one of the components of a parallel composition can execute a transition labelled with a, the other can execute a transition labelled with b, and the communication function γ is defined on a and b, then the parallel composition can execute a transition labelled with $\gamma(a, b)$. (It may help to think of the action a as standing for the event of sending some datum d, the action b as standing for the event of receiving datum d, and the action $\gamma(a, b)$ as standing for the event that two components communicate datum d.) The $\mathcal{A}$-labelled transition relation $\longrightarrow_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}$ and the termination relation $\downarrow_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}$ on $\mathcal{P}_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}$ are the least relations $\longrightarrow \subseteq \mathcal{P}_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)} \times \mathcal{A} \times \mathcal{P}_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}$ and $\downarrow \subseteq \mathcal{P}_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}$ satisfying the rules in Table 1.

The triple

$$\mathcal{T}_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)} = (\mathcal{P}_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}, \longrightarrow_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}, \downarrow_{\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)}),$$

consisting of the $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ expressions together with the $\mathcal{A}$-labelled transition relation and the termination predicate associated with them, is an example of an $\mathcal{A}$-labelled transition system space. In general, an $\mathcal{A}$-*labelled transition system space* $(S, \longrightarrow, \downarrow)$ consists of a (non-empty) set S, the elements of which are called *states*, together with an $\mathcal{A}$-labelled transition relation $\longrightarrow \subseteq S \times \mathcal{A} \times S$ and a subset $\downarrow \subseteq S$. We shall in this paper consider three more examples of transition system spaces, obtained by restricting the syntax of $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ and making special assumptions about the communication function.

Table 1. *Operational rules for* $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$*, with* $a \in \mathcal{A}$ *and* $H \subseteq \mathcal{A}$.

$$1\ \frac{}{\mathbf{1}\downarrow} \qquad 2\ \frac{}{a \xrightarrow{a} \mathbf{1}} \qquad 3\ \frac{p \xrightarrow{a} p'}{p+q \xrightarrow{a} p'} \qquad 4\ \frac{q \xrightarrow{a} q'}{p+q \xrightarrow{a} q'} \qquad 5\ \frac{p\downarrow}{p+q\downarrow} \qquad 6\ \frac{q\downarrow}{p+q\downarrow}$$

$$7\ \frac{p \xrightarrow{a} p'}{p \cdot q \xrightarrow{a} p' \cdot q} \qquad 8\ \frac{p\downarrow \quad q \xrightarrow{a} q'}{p \cdot q \xrightarrow{a} q'} \qquad 9\ \frac{p\downarrow \quad q\downarrow}{p \cdot q\downarrow} \qquad 10\ \frac{p \xrightarrow{a} p'}{p^{*} \xrightarrow{a} p' \cdot p^{*}} \qquad 11\ \frac{}{p^{*}\downarrow}$$

$$12\ \frac{p \xrightarrow{a} p'}{p \parallel q \xrightarrow{a} p' \parallel q} \qquad 13\ \frac{q \xrightarrow{a} q'}{p \parallel q \xrightarrow{a} p \parallel q'} \qquad 14\ \frac{p\downarrow \quad q\downarrow}{p \parallel q\downarrow}$$

$$15\ \frac{p \xrightarrow{a} p' \quad q \xrightarrow{b} q' \quad \gamma(a,b) \text{ is defined}}{p \parallel q \xrightarrow{\gamma(a,b)} p' \parallel q'} \qquad 16\ \frac{p \xrightarrow{a} p' \quad a \notin H}{\partial_H(p) \xrightarrow{a} \partial_H(p')} \qquad 17\ \frac{p\downarrow}{\partial_H(p)\downarrow}$$

Firstly, we define the $\mathcal{A}$-labelled transition system space

$$\mathcal{T}_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)} = (\mathcal{P}_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)}, \longrightarrow_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)}, \downarrow_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)})$$

corresponding with the process theory $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. The set of $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions $\mathcal{P}_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)}$ consists of the $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ process expressions without occurrences of the construct $\partial_H(\_)$. The $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ transition relation $\longrightarrow_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)}$ on $\mathcal{P}_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)}$ and the termination predicate $\downarrow_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)}$ on $\mathcal{P}_{\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)}$ are the transition relation and termination predicate induced on $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions by the operational rules in Table 1 minus the rules 16–17.

Secondly, we define the $\mathcal{A}$-labelled transition system space

$$\mathcal{T}_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})} = (\mathcal{P}_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}, \longrightarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}, \downarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})})$$

corresponding with the process theory $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$. The set of $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expressions $\mathcal{P}_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ consists of the $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ process expressions without occurrences of the construct $\partial_H(\_)$. The $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ transition relation $\longrightarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ on $\mathcal{P}_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ and the termination predicate $\downarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ on $\mathcal{P}_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ are the transition relation and termination predicate induced on $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expressions by the operational rules in Table 1 minus the rules 15–17. Alternatively (and equivalently) the transition relation $\longrightarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ can be defined as the restriction of the transition relation $\longrightarrow_{\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\varnothing)}$, with $\varnothing$ denoting the communication function that is everywhere undefined, to $\mathcal{P}_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$.

Thirdly, we define the $\mathcal{A}$-labelled transition system space

$$\mathcal{T}_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})} = (\mathcal{P}_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}, \longrightarrow_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}, \downarrow_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})})$$

associated with the process theory $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$. Let $\mathcal{P}_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ consist of all $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expressions without occurrences of the construct $\_ \parallel \_$. The $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ transition relation $\longrightarrow_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ and the $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ termination predicate $\downarrow_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ are the transition relation and the termination predicate induced on $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expressions by the operational rules in Table 1 minus the rules 12–17. That is, $\longrightarrow_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ and $\downarrow_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ are obtained by restricting $\longrightarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ and $\downarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ to $\mathcal{P}_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$.

Henceforth, we omit the subscripts $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ and $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ from transition relations and termination predicates whenever it is clear from the context which transition relation or termination predicate is meant. Furthermore, we shall often use $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ and $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ to denote the associated transition system spaces $\mathcal{T}_{\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)}$, $\mathcal{T}_{\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)}$, $\mathcal{T}_{\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})}$ and $\mathcal{T}_{\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})}$.

Let $\mathcal{T} = (S, \longrightarrow, \downarrow)$ be an $\mathcal{A}$-labelled transition system space. Let $s, s' \in S$; we write $s \longrightarrow s'$ if there exists $a \in \mathcal{A}$ such that $s \xrightarrow{a} s'$, and $s \not\longrightarrow s'$ if there does not exist $a \in \mathcal{A}$ such that $s \xrightarrow{a} s'$. We denote by $\longrightarrow^+$ the transitive closure of $\longrightarrow$, and by $\longrightarrow^*$ the reflexive-transitive closure of $\longrightarrow$. If $s \longrightarrow^* s'$, then we say that s' is *reachable* from s; the set of all states reachable from s is denoted by $[s]_{\rightarrow}$. $\mathcal{T}$ is called *regular* if $[s]_{\rightarrow}$ is finite for all $s \in S$.

Lemma 2.1. The transition system spaces $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$, and $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ are all regular.

Proof. By induction on the maximal depth of *-nestings it can be established for every p in the respective transition system spaces that the set $[p]_{\rightarrow}$ is finite (see also Bergstra *et al.* (1994, Lemma 3.1)). □

The following lemma records some straightforward consequences of the operational rules that will turn out to be useful in the technical developments to follow.

Lemma 2.2. Let p, q and r be $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions.

1. If $p \cdot q \longrightarrow^* r$, then there exists an $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression p' such that $p \longrightarrow^* p'$ and either $r = p' \cdot q$ or $p'\downarrow$ and $q \longrightarrow^* r$.
2. If $p \cdot q^* \longrightarrow^* r$, then either there exists an $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression p' such that $p \longrightarrow^* p'$ and $r = p' \cdot q^*$, or there exist $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions p' and q' such that $p \longrightarrow^* p'$, $p'\downarrow$, $q \longrightarrow^* q'$, and $r = q' \cdot q^*$.
3. If $p \parallel q \longrightarrow^* r$, then there exist $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions p' and q' such that $r = p' \parallel q'$, $p \longrightarrow^* p'$ and $q \longrightarrow^* q'$.

Proof. It is immediate from the operational rules that if $p \cdot q \longrightarrow r$, then either there exists p' such that $p \longrightarrow p'$ and $r = p' \cdot q$, or $p\downarrow$ and $q \longrightarrow r$. In the latter case it is, moreover, immediate from the operational rules that if $q = (q')^*$ for some $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression q', then there exists q'' such that $q' \longrightarrow q''$ and $r = q'' \cdot (q')^*$. Finally, it is also immediate from the operational rules that if $p \parallel q \longrightarrow r$, then either there exists p' such that $p \longrightarrow p'$ and $r = p' \parallel q$, or there exists q' such that $q \longrightarrow q'$ and $r = p \parallel q'$, or there exist p' and q' such that $p \longrightarrow p'$, $q \longrightarrow q'$ and $r = p' \parallel q'$. From these observations, the lemma follows with a straightforward induction on the lengths of transitions sequences from $p \cdot q$, $p \cdot q^*$ and $p \parallel q$ to r, respectively. □

We say that a state s is *normed* if there exists s' such that $s \longrightarrow^* s'$ and $s'\downarrow$.

Lemma 2.3. Let p and q be process expressions. Then

1. $p \cdot q$ is normed iff p and q are both normed; and
2. $p \parallel q$ is normed iff p and q are both normed.

Proof. Suppose that p and q are both normed. Then there exists p' such that $p \longrightarrow^* p'$ and $p'\downarrow$, and there exists q' such that $q \longrightarrow^* q'$ and $q'\downarrow$.

With a straightforward induction on the sum of the lengths of a transition sequences from p to p' and q to q', it follows from rules 7 and 8 that either $p \cdot q \longrightarrow^* p' \cdot q'$ (if the length of the transition sequence from q to q' is zero), or $p \cdot q \longrightarrow^* q'$ (if the length of the transition sequence from q to q' is non-zero), and $p' \cdot q'\downarrow$ follows from rule 9 and $q'\downarrow$ holds by assumption. Hence, $p \cdot q$ is normed.

With a similar induction it follows from rules 12 and 13 that $p \parallel q \longrightarrow^* p' \parallel q'$. Moreover, from rule 14 it follows that $p' \parallel q'\downarrow$. Hence $p \parallel q$ is normed.

Suppose that $p \cdot q$ is normed. Then there exists r such that $p \cdot q \longrightarrow^* r$ and $r\downarrow$. So, by Lemma 2.2, either there exists p' such that $p \longrightarrow^* p'$ and $r = p' \cdot q$, or there exist p' and q' such that $p \longrightarrow^* p'$, $p'\downarrow$, and $q \longrightarrow^* r$. In the first case, from $r = p' \cdot q$ and $r\downarrow$ it follows, according to the operational rules in Table 1, that $p'\downarrow$ and $q\downarrow$, and hence p and q are both normed. In the second case, since $p'\downarrow$ and $r'\downarrow$, it also follows that p and q are both normed.

Suppose that $p \parallel q$ is normed. Then, there exists r such that $p \cdot q \longrightarrow^* r$ and $r\downarrow$, and hence, by Lemma 2.2, there exist p' and q' such that $p \longrightarrow^* p'$, $q \longrightarrow^* q'$ and $r = p' \parallel q'$. From $r\downarrow$ it follows, according to the operational rules in Table 1, that $p'\downarrow$ and $q'\downarrow$. Hence, p and q are both normed. □

With every state s in $\mathcal{T}$ we can associate an *automaton* (or: *transition system*)

$$([s]_{\rightarrow}, \longrightarrow \cap ([s]_{\rightarrow} \times \mathcal{A} \times [s]_{\rightarrow}), \downarrow \cap [s]_{\rightarrow},\ s).$$

Its states are the states reachable from s, its transition relation and termination predicate are obtained by restricting $\longrightarrow$ and $\downarrow$ accordingly, and the state s is declared as the *initial state* of the automaton. If a transition system space is regular, then the automaton associated with a state in it is finite, i.e., it is a finite automaton in the terminology of automata theory. Thus, we get by Lemma 2.1 that the operational semantics of $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, and, *a fortiori*, that of $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ and $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$, associates a finite automaton with every process expression.

In automata theory, automata are usually considered as language acceptors and two automata are deemed indistinguishable if they accept the same languages. Language equivalence is, however, arguably too coarse in process theory, where the prevalent notion is bisimilarity (Milner 1989; Park 1981).

Definition 2.4. Let $\mathcal{T}_1 = (S_1, \longrightarrow_1, \downarrow_1)$ and $\mathcal{T}_2 = (S_2, \longrightarrow_2, \downarrow_2)$ be transition system spaces. A binary relation $\mathcal{R} \subseteq S_1 \times S_2$ is a *bisimulation* between $\mathcal{T}_1$ and $\mathcal{T}_2$ if it satisfies, for all $a \in \mathcal{A}$ and for all $s_1 \in S_1$ and $s_2 \in S_2$ such that $s_1\ \mathcal{R}\ s_2$, the following conditions:

— if there exists $s_1' \in S_1$ such that $s_1 \xrightarrow{a}_1 s_1'$, then there exists $s_2' \in S_2$ such that $s_2 \xrightarrow{a}_2 s_2'$ and $s_1'\ \mathcal{R}\ s_2'$;
— if there exists $s_2' \in S_2$ such that $s_2 \xrightarrow{a}_2 s_2'$, then there exists $s_1' \in S_1$ such that $s_1 \xrightarrow{a}_1 s_1'$ and $s_1'\ \mathcal{R}\ s_2'$; and
— $s_1\downarrow_1$ if, and only if, $s_2\downarrow_2$.

States $s_1 \in S_1$ and $s_2 \in S_2$ are *bisimilar* (notation: $s_1 \leftrightarrow s_2$) if there exists a bisimulation $\mathcal{R}$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ such that $s_1 \mathrel{\mathcal{R}} s_2$.

To achieve a sufficient level of generality, we have defined bisimilarity as a relation between transition system spaces; to obtain a suitable notion of bisimulation between automata one should add the requirement that the initial states of the automata be related.

Based on the associated transition system spaces, we can now define what we mean when some transition system space is, modulo bisimilarity, less expressive than some other transition system space.

Definition 2.5. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be transition system spaces. We say that $\mathcal{T}_1$ is *less expressive* than $\mathcal{T}_2$ (notation: $\mathcal{T}_1 \prec \mathcal{T}_2$) if every state in $\mathcal{T}_1$ is bisimilar to a state in $\mathcal{T}_2$, and, moreover, there is a state in $\mathcal{T}_2$ that is *not* bisimilar to some state in $\mathcal{T}_1$.

When we investigate the expressiveness of $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, we want to be able to choose γ. So, we are actually interested in the expressiveness of the (disjoint) union of all transition system spaces $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ with γ ranging over all communication functions; we denote this transition system space by $\bigcup_\gamma \mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$. Similarly, we denote by $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ the (disjoint) union of all transition system spaces $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ with γ ranging over all communication functions. In this paper we shall establish that

$$\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A}) \prec \mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A}) \prec \bigcup_{\gamma} \mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma) \prec \bigcup_{\gamma} \mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma).$$

Remark 2.6. Gorla (2010) proposed a set of general criteria for comparing the relative expressiveness of process calculi. There are two assumptions in his work that preclude its application in our setting. First, Gorla (2010) presupposes that the compared calculi both include a notion of parallel composition; this would exclude $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ from consideration. Second, the criteria formulated by Gorla (2010) are based on (the transitive-reflexive closure) of a reduction relation on expressions. Such a reduction relation naturally arises in a calculus in which the result of synchronization between processes is considered unobservable (i.e., is semantically modelled as a transition labelled with the special unobservable τ). In the process calculi we consider, however, there is no notion of unobservability; our results are up to strong bisimilarity.

3. Strongly connected components

A crucial step in establishing the strictness of the aforementioned expressiveness hierarchy will be to characterize the strongly connected components in the respective transition system spaces. First, we recall below the notion of strongly connected component. Then we shall provide an inductive characterization of the strongly connected components in $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$, and $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$.

Definition 3.1. A *strongly connected component* in a transition system space $\mathcal{T} = (S, \longrightarrow, \downarrow)$ is a maximal subset C of S such that $s \longrightarrow^* s'$ for all $s, s' \in C$. A strongly connected component C is *trivial* if it consists of only one state, say $C = \{s\}$, and $s \not\longrightarrow s$; otherwise, it is *non-trivial*.

Note that every state of a transition system space is in precisely one strongly connected component of that space. Furthermore, if s is a state of a non-trivial strongly connected component, then $s \longrightarrow^{+} s$. Since in a strongly connected component from every state every other state can be reached, we get as a corollary to Lemma 2.1 that strongly connected components in $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ and $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ are finite.

Let $\mathcal{T} = (S, \longrightarrow, \downarrow)$ be a transition system space, let $s \in S$, and let $C \subseteq S$ be a strongly connected component in S. We say that C is *reachable* from s if $s \longrightarrow^{*} s'$ for some $s' \in C$; clearly, this means that $s \longrightarrow^{*} s'$ for all $s' \in C$.

Lemma 3.2. Let $\mathcal{T}_1 = (S_1, \longrightarrow_1, \downarrow_1)$ and $\mathcal{T}_2 = (S_2, \longrightarrow_2, \downarrow_2)$ be regular transition system spaces, and let $s_1 \in S_1$ and $s_2 \in S_2$ be such that $s_1 \leftrightarrows s_2$. If s_1 is an element of a strongly connected component C_1 in $\mathcal{T}_1$, then there exists a strongly connected component C_2 reachable from s_2 satisfying that for all $s_1' \in C_1$ there exists $s_2' \in C_2$ such that $s_1' \leftrightarrows s_2'$.

Proof. According to Lemma 2.1, the set $[s_2]_{\rightarrow}$ is finite; we use induction on $|[s_2]_{\rightarrow}|$.

Let C_1 be the strongly connected component in $\mathcal{T}_1$ that contains s_1. If the unique strongly connected component C_2 in $\mathcal{T}_2$ containing s_2 is such that for all $s_1' \in C_1$ there exists $s_2' \in C_2$ such that $s_1' \leftrightarrows s_2'$, then we are done. In particular, this is the case if $|[s_2]_{\rightarrow}| = 1$. Otherwise, there exists a state in C_1, say s_1', for which there is no bisimilar state in the strongly connected component containing s_2, and since $s_1 \longrightarrow^{+} s_1' \longrightarrow^{+} s_1$, it follows that there exists s_2' distinct from s_2 such that $s_2 \longrightarrow^{+} s_2'$ and $s_1 \leftrightarrows s_2'$. Clearly, $|[s_2']_{\rightarrow}| < |[s_2]_{\rightarrow}|$, so by the induction hypothesis there exists a strongly connected component C_2 reachable from s_2', and hence also from s_2, satisfying that for all $s_1' \in C_1$ there exists $s_2' \in C_2$ such that $s_1' \leftrightarrows s_2'$. □

We shall now establish that a non-trivial strongly connected component in the transition system space $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ is either of the form

$$\{p_1 \cdot q^{*}, \dots, p_n \cdot q^{*}\}$$

with p_i $(0 \leqslant i \leqslant n)$ reachable from q and $\{p_1, \dots, p_n\}$ not a strongly connected component, or of the form

$$\{p_1 \cdot q, \dots, p_n \cdot q\}$$

where $\{p_1, \dots, p_n\}$ is a strongly connected component, or of the form

$$\{p_1 \parallel q_1, \dots, p_n \parallel q_n\}$$

where both $\{p_1, \dots, p_n\}$ and $\{q_1, \dots, q_n\}$ are strongly connected components. To this end, we first prove, by reasoning on the basis of the operational semantics, that process expressions in a non-trivial strongly connected component are necessarily sequential compositions or parallel compositions; at the heart of the argument will be the following measure on $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions.

Definition 3.3. Let p an $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression; then $\#(p)$ is defined with recursion on the structure of p by the following clauses:

— $\#(\mathbf{0}) = \#(\mathbf{1}) = 0$, and $\#(a) = 1$;

— $\#(p \cdot q) = \begin{cases} 0 & \text{if } q \text{ is a star expression} \\ \#(q) + 1 & \text{otherwise;} \end{cases}$
— $\#(p + q) = \max\{\#(p), \#(q)\} + 1$;
— $\#(p^*) = 1$; and
— $\#(p \parallel q) = 0$.

We establish that $\#(\_)$ is non-increasing over transitions, and, in fact, in most cases decreases.

Lemma 3.4. If p and p' are $\text{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions such that $p \longrightarrow^+ p'$, then $\#(p) \geqslant \#(p')$. Moreover, if $\#(p) = \#(p')$, then either $p = p_1 \cdot q$ and $p' = p'_1 \cdot q$, or $p = p_1 \parallel p_2$ and $p' = p'_1 \parallel p'_2$ for some $\text{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions p_1, p_2, p'_1, p'_2, and q.

Proof. Let p and p' be $\text{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions. Note that if the lemma holds in the special case that $p \longrightarrow p'$, then the general case follows with a straightforward induction on the length of a transition sequence from p to p'. In the remainder of the proof we concentrate on the special case: we prove that $p \longrightarrow p'$ implies $\#(p) \geqslant \#(p')$, and, moreover, whenever $\#(p) = \#(p')$ then, for some $\text{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions p_1, p'_1, p_2, p'_2, and q, either $p = p_1 \cdot q$ and $p' = p'_1 \cdot q$, or $p = p_1 \parallel p_2$ and $p' = p'_1 \parallel p_2$ with $p_1 \longrightarrow p'_1$, or $p = p_1 \parallel p_2$ and $p' = p_1 \parallel p'_2$ with $p_2 \longrightarrow p'_2$ or $p = p_1 \parallel p_2$, $p' = p'_1 \parallel p'_2$, and there exist $b, c \in \mathcal{A}$ such that $p_1 \xrightarrow{b} p'_1$, $p_2 \xrightarrow{c} p'_2$ and $\gamma(b, c)$ is defined.

Let $a \in \mathcal{A}$ be such that $p \xrightarrow{a} p'$; we reason by induction on a derivation according to the operational rules for $\text{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ (rules 1–15) of the transition $p \xrightarrow{a} p'$. We distinguish cases according to which rule is applied last in such a derivation. (Clearly, since the rules 1, 5, 6, 9, 11 and 14 do not have a transition as a conclusion, we need not consider them.)

1. Suppose that the last rule applied is rule 2; then $p = a$ and $p' = \mathbf{1}$, so $\#(p) = 1 > 0 = \#(p')$.
2. Suppose that the last rule applied is rule 3. Then there exist p_1 and p_2 such that $p = p_1 + p_2$ and $p_1 \xrightarrow{a} p'$. Since there is a derivation of $p_1 \xrightarrow{a} p'$ that is a proper subderivation of the considered derivation of $p \xrightarrow{a} p'$, it follows by the induction hypothesis that $\#(p_1) \geqslant \#(p')$, and hence $\#(p) = \max\{\#(p_1), \#(p_2)\} + 1 \geqslant \#(p_1) + 1 \geqslant \#(p') + 1 > \#(p')$.
3. If the last rule applied is rule 4, then the proof that $\#(p) > \#(p')$ is analogous to the proof in the previous case.
4. Suppose that the last rule applied is rule 7. Then there exist p_1, p'_1 and q such that $p = p_1 \cdot q$, $p_1 \xrightarrow{a} p'_1$, and $p' = p'_1 \cdot q$. There are two cases: if q is a star expression then $\#(p) = 0 = \#(p')$, and otherwise $\#(p) = \#(q) + 1 = \#(p')$.
5. Suppose that the last rule applied is rule 8. Then there exist p_1 and q such that $p = p_1 \cdot q$, with $p_1\downarrow$ and $q \xrightarrow{a} p'$. Note that, since there is a derivation of $q \xrightarrow{a} p'$ that is a proper subderivation of the considered derivation of $p \xrightarrow{a} p'$, we get by the induction hypothesis that $\#(q) \geqslant \#(p')$. There are two subcases. On the one hand, if q is a star expression, then the last rule applied in the derivation of $q \xrightarrow{a} p'$ is rule 10, so there exists p'_1 such that $p' = p'_1 \cdot q$. Since q is a star expression, it

follows that $\#(p) = 0 = \#(p')$. On the other hand, if q is not a star expression, then $\#(p) = \#(q) + 1 \geqslant \#(p') + 1 > \#(p')$.

6. If the last rule applied is rule 10, then there exist q and q' such that $p = q^*$, $q \xrightarrow{a} q'$, and $p' = q' \cdot q^*$. It then follows immediately from the definition of $\#(\_)$ that $\#(p) = 1 > 0 = \#(q' \cdot q^*)$.
7. Suppose that the last rule applied is rule 12. Then there exist p_1, p_1' and p_2 such that $p = p_1 \parallel p_2$, $p_1 \xrightarrow{a} p_1'$, and $p' = p_1' \parallel p_2$, and hence $\#(p) = 0 = \#(p')$.
8. Suppose that the last rule applied is rule 13. Then there exist p_1, p_2 and p_2' such that $p = p_1 \parallel p_2$, $p_2 \xrightarrow{a} p_2'$, and $p' = p_1 \parallel p_2'$, and hence $\#(p) = 0 = \#(p')$.
9. Suppose that the last rule applied is rule 15. Then there exist p_1, p_2, p_1', p_2', b and c, such that $p = p_1 \parallel p_2$, $p_1 \xrightarrow{b} p_1'$, $p_2 \xrightarrow{c} p_2'$, $p' = p_1' \parallel p_2'$, and $\gamma(b, c) = a$, and hence $\#(p) = 0 = \#(p')$.

To complete the proof, we note that $\#(p) = \#(p')$ only in case 4, in the first subcase of 5, in case 7, in case 8, and in case 9, and in each of these cases the extra requirement of the lemma is satisfied. $\square$

For a succinct presentation of the characterization of non-trivial strongly connected components in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$, it is convenient to introduce some auxiliary notation. Let P and Q be sets of $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ expressions, and let q be an $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ expression; we define the sets of $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ expressions $P \cdot q$ and $P \parallel Q$, respectively, by

$$P \cdot q = \{p \cdot q \mid p \in P\} \ \ ; \text{ and}$$
$$P \parallel Q = \{p \parallel q \mid p \in P \wedge q \in Q\}.$$

We also write $P \parallel q$ and $p \parallel Q$ for $P \parallel \{q\}$ and $\{p\} \parallel Q$, respectively.

Proposition 3.5. If C_1 and C_2 are strongly connected components in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$, then $C_1 \parallel C_2$ is also a strongly connected component in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$; moreover, if at least one of C_1 and C_2 is nontrivial, then so is $C_1 \parallel C_2$.

Proof. Suppose that C_1 and C_2 are strongly connected components; we need to prove that $C_1 \parallel C_2$ is a strongly connected component. To this end, we need to establish that

1. $p_1 \parallel p_2 \longrightarrow^* p_1' \parallel p_2'$ for all $p_1, p_1' \in C_1$ and $p_2, p_2' \in C_2$, and
2. for all $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ expressions p_1, p_2, q_1 and q_2 such that $p_1 \parallel p_2 \in C_1 \parallel C_2$, $p_1 \parallel p_2 \longrightarrow q_1 \parallel q_2$ and $q_1 \parallel q_2 \notin C_1 \parallel C_2$ it holds that $q_1 \parallel q_2 \not\longrightarrow^* p_1 \parallel p_2$.

To prove (1), note that if $p_1, p_1' \in C_1$ and $p_2, p_2' \in C_2$, then, since C_1 and C_2 are strongly connected components, we have that $p_1 \longrightarrow^* p_1'$ and $p_2 \longrightarrow^* p_2'$, so, according to operational rules 12 and 13, it follows that $p_1 \parallel p_2 \longrightarrow^* p_1' \parallel p_2'$.

To prove (2), note that if $q_1 \parallel q_2 \notin C_1 \parallel C_2$, then $q_1 \notin C_1$ or $q_2 \notin C_2$; without loss of generality we can assume that $q_1 \notin C_1$. Note that $p_1 \parallel p_2 \longrightarrow q_1 \parallel q_2$ by Lemma 2.2(3) implies $p_1 \longrightarrow q_1$. So, since $p_1 \longrightarrow q_1$, $q_1 \notin C_1$, and C_1 is a strongly connected component, it follows that $q_1 \not\longrightarrow^* p_1$, and hence $q_1 \parallel q_2 \not\longrightarrow^* p_1 \parallel p_2$.

It remains to establish that if at least one of C_1 and C_2 is non-trivial, then so is $C_1 \parallel C_2$. We proceed by contraposition. Suppose that $C_1 \parallel C_2$ is trivial. Then there exist $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ expressions p_1 and p_2 such that $C_1 \parallel C_2 = \{p_1 \parallel p_2\}$, and $p_1 \parallel p_2 \not\longrightarrow p_1 \parallel p_2$.

Then $C_1 = \{p_1\}$ and $C_2 = \{p_2\}$, and from $p_1 \parallel p_2 \not\longrightarrow p_1 \parallel p_2$ it follows by operational rules 12 and 13 that $p_1 \not\longrightarrow p_1$ and $p_2 \not\longrightarrow p_2$. Hence, C_1 and C_2 are both trivial. □

Lemma 3.6. If C is a non-trivial strongly connected component in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, then either there exist a set of $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions C' and a $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression q such that $C = C' \cdot q$, or there exist strongly connected components C_1 and C_2 in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, at least one of them non-trivial, such that $C = C_1 \parallel C_2$.

Proof. Let $p \in C$. Then, since $p \longrightarrow^+ p$, by Lemma 3.4, either $p = p_1 \cdot q$ or $p = p_1 \parallel p_2$, for some $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions p_1, p_2 and q. We consider these cases separately:

1. Suppose that $p = p_1 \cdot q$, and consider an arbitrary $p' \in C$. Then, since $p \longrightarrow^+ p' \longrightarrow^+ p$, by Lemma 3.4, $\#(p) \geqslant \#(p') \geqslant \#(p)$, so $\#(p) = \#(p')$. Therefore, again by Lemma 3.4, there exist p'_1 such that $p' = p'_1 \cdot q$. It follows that there exists a set of $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions C' such that $C = C' \cdot q$.
2. Suppose that $p = p_1 \parallel p_2$, and let C_1 and C_2 be the strongly connected components containing p_1 and p_2, respectively. We establish that $C = C_1 \parallel C_2$ and that one of C_1 and C_2 is non-trivial.
   To see that $C \subseteq C_1 \parallel C_2$, consider an arbitrary $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression $p' \in C$. Then, since $p' \longrightarrow^+ p \longrightarrow^+ p'$, by Lemma 2.2(3) there exist $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions p'_1 and p'_2 such that $p' = p'_1 \parallel p'_2$, $p'_1 \longrightarrow^* p_1 \longrightarrow^* p'_1$, and $p'_2 \longrightarrow^* p_2 \longrightarrow^* p'_2$. It follows that p'_1 and p'_2 are in the same strongly components as p_1 and p_2, respectively, so $p'_1 \in C_1$ and $p'_2 \in C_2$. Hence, $p' \in C_1 \parallel C_2$. Moreover, from $p' \longrightarrow^+ p'$ it can be easily deduced that either $p'_1 \longrightarrow^+ p'_1$ or $p'_2 \longrightarrow^+ p'_2$, so at least one of C_1 and C_2 is non-trivial.
   To see that $C_1 \parallel C_2 \subseteq C$, consider an arbitrary $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression $p' \in C_1 \parallel C_2$. Then there exist $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions $p'_1 \in C_1$ and $p'_2 \in C_2$ such that $p' = p'_1 \parallel p'_2$. Since C_1 is a strongly connected component containing p_1 and p'_1, we have that $p'_1 \longrightarrow^* p_1 \longrightarrow^* p'_1$, and, since C_2 is a strongly connected component containing p_2 and p'_2, we have that $p'_2 \longrightarrow^* p_2 \longrightarrow^* p'_2$. It follows that $p' \longrightarrow^* p \longrightarrow^* p'$, so $p' \in C$.

□

We proceed to give an inductive description of the non-trivial strongly connected components in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. The basis for the inductive description is the following notion of basic strongly connected component.

Definition 3.7. A non-trivial strongly connected component C in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ is *basic* if there exist a set of $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions C' and an $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression q such that $C = C' \cdot q^*$ and C' is not a strongly connected component in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$.

Proposition 3.8. Let C be a non-trivial strongly connected component in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. Then one of the following holds:

1. C is a basic strongly connected component; or
2. there exist a non-trivial strongly connected component C' in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ and a $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression q such that $C = C' \cdot q$; or
3. there exist strongly connected components C_1 and C_2 in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, at least one of them non-trivial, such that $C = C_1 \parallel C_2$.

Proof. Let C be a non-trivial strongly connected component in $\mathrm{ACP}^{*;-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. According to Lemma 3.6 there are two cases: either there exist a set of process expressions C' and a process expression q such that $C = C' \cdot q$, or there exist strongly connected components C_1 and C_2, at least one of them non-trivial, such that $C = C_1 \parallel C_2$.

In the second case, there is nothing left to prove.

In the first case, we should argue that either C is basic, or C' is a non-trivial strongly connected component. So, suppose that C' is not a non-trivial strongly connected component. Then there are $p, p' \in C'$ such that $p \not\longrightarrow^+ p'$. Since C is a non-trivial strongly connected component and $C = C' \cdot q$, it holds that $p \cdot q \longrightarrow^+ p' \cdot q$. Using that $p \not\longrightarrow^+ p'$, it can be established with induction on the length of the transition sequence from $p \cdot q$ to $p' \cdot q$ that $q \longrightarrow^+ p' \cdot q$. It follows by Lemma 3.4 that $\#(q) \geqslant \#(p' \cdot q)$, and therefore, according to the definition of $\#(\_)$, q must be a star expression. We conclude that C is basic. □

Note that, in the above proposition, one of the strongly connected components C_1 and C_2 may be trivial in which case it consists of a single $\mathrm{ACP}^{*;-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression.

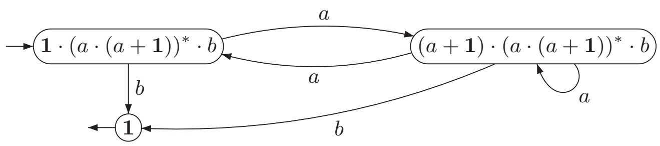
Every $\mathrm{ACP}^{*;-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression is a $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression. Moreover, although the transition relation associated with $\mathrm{ACP}^{*;-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ does not coincide with the transition relation associated with $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$, the relation $\longrightarrow^*$ associated with $\mathrm{ACP}^{*;-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ actually coincides with the relation $\longrightarrow^*$ associated with $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$. Hence, it immediately follows that a set of $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expressions is a strongly connected component in $\mathrm{ACP}^{*;-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ iff it is a strongly connected component in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$. Thus, we may use Proposition 3.5 also for strongly connected components in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$, and Proposition 3.8 also provides a characterization of the non-trivial strongly connected components in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$. For the sake of clarity, we restate both propositions for $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ in the following two corollaries.

Corollary 3.9. If C_1 and C_2 are strongly connected components in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$, then $C_1 \parallel C_2$ is also a strongly connected component in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$; moreover, if at least one of C_1 and C_2 is non-trivial, then so is $C_1 \parallel C_2$.

Corollary 3.10. Let C be a non-trivial strongly connected component in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$. Then one of the following holds:

1. C is a basic strongly connected component; or
2. there exist a non-trivial strongly connected component C' in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ and a $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression q such that $C = C' \cdot q$; or
3. there exist strongly connected components C_1 and C_2 in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$, at least one of them non-trivial, such that $C = C_1 \parallel C_2$.

Recall that $\longrightarrow_{\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ is obtained by restricting $\longrightarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}$ to $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expressions, which are $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expressions without occurrences of $\parallel$. It is easy to see that if p is a $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression and $p \longrightarrow_{\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})}{}^{*}\, p'$, then p' is a $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression too. Hence, if a strongly connected component in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ contains a $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression, then it consists entirely of $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expressions. We therefore obtain the following inductive characterization of non-trivial strongly connected components in $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ as a consequence of the preceding corollary.



$\mathbf{1} \cdot (a \cdot (a + \mathbf{1}))^* \cdot b$
$(a + \mathbf{1}) \cdot (a \cdot (a + \mathbf{1}))^* \cdot b$
$\mathbf{1}$
a
a
a
b
b

Fig. 1. A transition system in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ with a cycle with multiple exit transitions.

Corollary 3.11. Let C be a non-trivial strongly connected component in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. Then one of the following holds:

1. C is a basic strongly connected component; or
2. there exist a non-trivial strongly connected component C' in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ and a $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ expression q such that $C = C' \cdot q$.

4. Relative expressiveness of $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ and $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$

Bergstra *et al.* (1994) prove that $\mathrm{BPA}^*_{\mathbf{0}}(\mathcal{A})$ is less expressive than $\mathrm{PA}^*_{\mathbf{0}}(\mathcal{A})$, by arguing that the $\mathrm{PA}^*_{\mathbf{0}}(\mathcal{A})$ expression $(a \cdot b)^*c \parallel d$ is not bisimilar with any $\mathrm{BPA}^*_{\mathbf{0}}(\mathcal{A})$ expression. Bergstra *et al.* (2001) present an alternative and more general proof that the $\mathrm{PA}^*_{\mathbf{0}}(\mathcal{A})$ expression above is not expressible in $\mathrm{BPA}^*_{\mathbf{0}}(\mathcal{A})$. (Actually, the $\mathrm{PA}^*_{\mathbf{0}}(\mathcal{A})$ expression they employ uses only a single action a, i.e., considers the $\mathrm{PA}^*_{\mathbf{0}}(\mathcal{A})$ expression $(a \cdot a)^*a \parallel a$; we use the actions b, c and d for clarity.) They establish that the $\mathrm{PA}^*_{\mathbf{0}}(\mathcal{A})$ expression above fails the following general property, which is satisfied by all $\mathrm{BPA}^*_{\mathbf{0}}(\mathcal{A})$-expressible automata:

If C is a cycle in an automaton associated with a $\mathrm{BPA}^*_{\mathbf{0}}(\mathcal{A})$ expression, then there is at most one state $p \in C$ that has an exit transition.

(A cycle is a sequence $(p_1, \ldots, p_n)$ such that $p_i \longrightarrow p_{i+1}$ $(1 \leqslant i < n)$ and $p_n \longrightarrow p_1$; an exit transition from p_i is a transition $p_i \longrightarrow p'_i$ such that no element of the cycle is reachable from p'_i.)

The following example shows that automata associated with $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ expressions do not satisfy the property above.

Example 4.1. Consider the automaton associated with the $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ expression $\mathbf{1} \cdot (a \cdot (a + \mathbf{1}))^* \cdot b$ (see Figure 1). It has a cycle consisting of two states, and both states on the cycle have a b-transition off the cycle.

In this section, we shall establish that $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is less expressive than $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. As in Bergstra *et al.* (2001) we prove that $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$-expressible automata satisfy a general property that some automaton expressible in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ fails to satisfy. We find it technically convenient, however, to base our relative expressiveness proofs on the notion of strongly connected component, instead of cycle. Note, e.g., that every process expression is an element of precisely one strongly connected component, while it may reside on more than one cycle. Furthermore, if $p \longrightarrow q$ and p and q are in distinct strongly connected components, then we can be sure that $p \longrightarrow q$ is an exit transition, while if p and q are on distinct cycles, then it may happen that p is reachable from q.

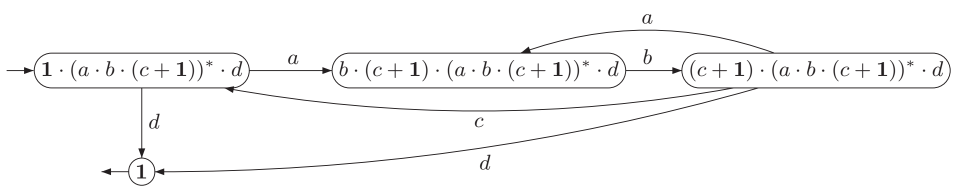

$\mathbf{1}\cdot(a\cdot b\cdot(c+\mathbf{1}))^*\cdot d$
$b\cdot(c+\mathbf{1})\cdot(a\cdot b\cdot(c+\mathbf{1}))^*\cdot d$
$(c+\mathbf{1})\cdot(a\cdot b\cdot(c+\mathbf{1}))^*\cdot d$
$\mathbf{1}$
a
b
a
c
d
d

Fig. 2. A non-trivial strongly connected component in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ with multiple exit transitions.

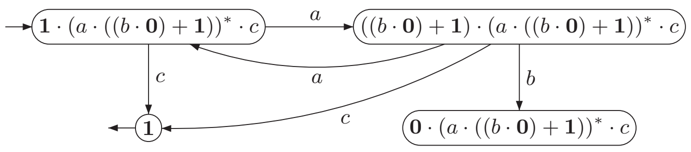

$\mathbf{1}\cdot(a\cdot((b\cdot\mathbf{0})+\mathbf{1}))^*\cdot c$
$((b\cdot\mathbf{0})+\mathbf{1})\cdot(a\cdot((b\cdot\mathbf{0})+\mathbf{1}))^*\cdot c$
$\mathbf{0}\cdot(a\cdot((b\cdot\mathbf{0})+\mathbf{1}))^*\cdot c$
$\mathbf{1}$
a
a
b
c
c

Fig. 3. A strongly connected component with an unnormed exit transition.

The crucial tool that will allow us to establish that $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is less expressive than $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ will be a special property of states with a transition out of their strongly connected component in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. Roughly, if C is a strongly connected component in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, then all states with a transition out of C have the same transitions out of C.

Definition 4.2. Let C be a strongly connected component in the transition system space $\mathcal{T}=(S,\longrightarrow,\downarrow)$ and let $s\in C$. An exit transition from s is a pair (a,s') such that $s\xrightarrow{a}s'$ and $s'\notin C$. An element $s\in C$ is called an *exit state* if $s\downarrow$ or there exists an exit transition from s.

Example 4.3. Consider the automaton associated with the $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ expression $\mathbf{1}\cdot(a\cdot b\cdot(c+\mathbf{1}))^*\cdot d$ (see Figure 2). It has a strongly connected component with two exit states, both with a single exit transition $(d,\mathbf{1})$.

Non-trivial strongly connected components in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ arise from executing the argument of a Kleene star. An exit state of a strongly connected component in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is then a state with the termination option. Due to the presence of $\mathbf{0}$ in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ this is, however, not the only type of exit state in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ strongly connected components.

Example 4.4. Consider the automaton associated with the $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ expression $\mathbf{1}\cdot(a\cdot((b\cdot\mathbf{0})+\mathbf{1}))^*\cdot c$ (see Figure 3). The strongly connected component contains two exit states and two (distinct) exit transitions. One of these exit transitions leads to a deadlocked state.

The preceding example illustrates that the special property of strongly connected components in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ that we are after, should exclude from consideration any exit transition arising from an occurrence of $\mathbf{0}$. This is achieved in the following definition.

Definition 4.5. Let C be a strongly connected component and let $s \in C$. An exit transition (a, s') from s is *normed* if s' is normed. We denote by $ET_n(s)$ the set of normed exit transitions from s.

An exit state $s \in C$ is *alive* if $s\downarrow$ or there exists a normed exit transition from s.

Our goal is to prove that any two alive exit states in a strongly connected component C in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ have the same normed exit transitions. We proceed by first characterizing the set of exit transitions of a state in C, using its inductive description provided by Corollary 3.11: Lemma 4.7 will deal with the case that C is a basic non-trivial strongly connected component, and Lemma 4.9 will deal with the case that $C = C' \cdot q$ for some non-trivial strongly connected component C' in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ and a $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression q.

The following three lemmas are valid not only for $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$, but also for $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ and $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$; in fact, we will reuse them in subsequent sections.

Lemma 4.6. Let $C = C' \cdot q^*$ be a basic strongly connected component. Then $q \longrightarrow^+ p$ for all $p \in C'$.

Proof. Let $p \in C'$. Note that, since C' is not a strongly component, there exists $r \in C'$ such that either $r \not\longrightarrow^+ p$ or $p \not\longrightarrow^+ r$.

First we consider the case that $r \not\longrightarrow^+ p$. Then, since C is a strongly connected component, it holds that $r \cdot q^* \longrightarrow^+ p \cdot q^*$, so, by Lemma 2.2(2), there exist $r_0, \ldots, r_k$ such that $r \cdot q^* = r_0 \cdot q^* \longrightarrow \cdots \longrightarrow r_k \cdot q^* = p \cdot q^*$. From $r \not\longrightarrow^+ p$ it follows that $r_\ell \not\longrightarrow r_{\ell+1}$ for some $0 \leqslant \ell < k$; we let ℓ be the largest such natural number, so that in addition we have $r_{\ell+1} \longrightarrow^* p$. Then, from $r_\ell \not\longrightarrow r_{\ell+1}$ we conclude that the last rule applied in the derivation of $r_\ell \cdot q^* \longrightarrow r_{\ell+1} \cdot q^*$ must be rule 8, so $q^* \longrightarrow r_{\ell+1} \cdot q^*$, which by rule 10 implies that $q \longrightarrow r_{\ell+1}$. Thus, we have now established that $q \longrightarrow r_{\ell+1} \longrightarrow^* p$.

It remains to consider the case that $r \longrightarrow^+ p$ and $p \not\longrightarrow^+ r$. Then from $p \not\longrightarrow^+ r$ it can be concluded, in a similar manner as above, that $q \longrightarrow^+ r$, and hence $q \longrightarrow^+ p$. □

Lemma 4.7. If C is a basic strongly connected component, then $ET_n(p) = \emptyset$ for all $p \in C$.

Proof. Let $p \in C$. To prove that $ET_n(p) = \emptyset$, we suppose that p has a normed exit transition and derive a contradiction. So, let r and r' be process expressions such that $r \notin C$ and $p \xrightarrow{a} r \longrightarrow^* r'\downarrow$. Since C is a basic strongly connected component, there exist a set of process expressions C' and a process expression q such that $C = C' \cdot q^*$. It follows from $p \in C$ that there exists $p' \in C'$ such that $p = p' \cdot q^*$, and hence, by Lemma 2.2(2), it follows from $p \longrightarrow^* r'$ that there exists a process expression s such that $r' = s \cdot q^*$. Since $r'\downarrow$, according to rule 9, also $s\downarrow$. Hence, since $q \longrightarrow^+ p$ by Lemma 4.6, it follows that $r' \longrightarrow^+ p$. But then $r' \in C$ and, *a fortiori*, $r \in C$, contradicting our assumption that $r \notin C$. We conclude that p does not admit normed exit transitions, and hence $ET_n(p) = \emptyset$. □

Lemma 4.8. Let C be a non-trivial strongly connected component, let $p \in C$, and let q be a process expression such that $C \cdot q$ is a strongly connected component. Then $p \cdot q$ is an alive exit state in $C \cdot q$ iff p is an alive exit state in C and q is normed.

Proof. We prove the implications from left to right and from right to left separately.

($\Rightarrow$)If $p \cdot q$ is an alive exit state in $C \cdot q$, then either $p \cdot q{\downarrow}$, or there exist $a \in \mathcal{A}$ and a normed process expression r such that $p \cdot q \xrightarrow{a} r$ and $r \notin C \cdot q$.
In the first case it is immediately clear from rule 9 that $p{\downarrow}$ and $q{\downarrow}$, so p is an alive exit state in C and q is normed.
In the second case, since $p \cdot q \xrightarrow{a} r$ and r is normed, it follows that $p \cdot q$ is normed, and hence, by Lemma 2.3(1), q is normed. It remains to prove that p is an alive exit state. To this end, note that from $p \cdot q \xrightarrow{a} r$ it follows that either $p{\downarrow}$ and $q \xrightarrow{a} r$ or there exists p' such that $p \xrightarrow{a} p'$ and $r = p' \cdot q$. If $p{\downarrow}$, then p is an alive exit state of C directly by definition. If there exists p' such that $p \xrightarrow{a} p'$ and $r = p' \cdot q$, then it remains to prove that $p' \notin C$ and p' is normed. Since $p' \in C$ incorrectly implies $r = p' \cdot q \in C \cdot q$, it follows that $p' \notin C$, and since r is normed, by Lemma 2.3(1) so is p'.

($\Leftarrow$)Suppose that p is an alive exit state in C and q is normed. Then there are two cases: either $p{\downarrow}$, or there exist an $a \in \mathcal{A}$ and a normed process expression such that $p \xrightarrow{a} p'$ and $p' \notin C$. We consider these two cases separately.
Suppose that $p{\downarrow}$. If also $q{\downarrow}$, then $p \cdot q{\downarrow}$, and hence $p \cdot q$ is an alive exit state in $C \cdot q$. If $q{\not\downarrow}$, then, since q is normed, there exist $a \in \mathcal{A}$ and a normed process expression q' such that $q \xrightarrow{a} q'$. From $p \cdot q \xrightarrow{a} q'$ it follows by Lemma 3.4 that either $\#(p \cdot q) > \#(q')$, or $\#(p \cdot q) = \#(q')$ and there is a p' such that $q' = p' \cdot q$. If $\#(p \cdot q) > \#(q')$, then (a, q') is a normed exit transition from $p \cdot q$, so $p \cdot q$ is an alive exit state in $C \cdot q$. The other case, that $\#(p \cdot q) = \#(q')$ and there is a p' such that $q' = p' \cdot q$, cannot occur, for $q \xrightarrow{a} p' \cdot q$ implies by Lemma 3.4 and the definition of $\#(\_)$ that q is a star expression, which is in contradiction with our assumption that $q{\not\downarrow}$.
Suppose there exist $a \in \mathcal{A}$ and a normed process expression p' such that $p \xrightarrow{a} p'$ and $p' \notin C$. Then $p \cdot q \xrightarrow{a} p' \cdot q$, and since $p' \notin C$ it follows that $p' \cdot q \notin C \cdot q$. We conclude that $(a, p' \cdot q)$ is an exit transition of $p \cdot q$, which, by Lemma 2.3(1), is normed since p' and q are both normed.

□

For a characterization of the set of normed exit transitions of a sequential composition, it is convenient to have the following notation: if E is a set of exit transitions and p is a expression, then $E \cdot p$ is defined by

$$E \cdot p = \{(a, q \cdot p) \mid (a, q) \in E\}.$$

Lemma 4.9. Let C be a non-trivial strongly connected component, let $p \in C$, and let q be a normed process expression such that $C \cdot q$ is a strongly connected component. Then

$$ET_n(p \cdot q) = \begin{cases} ET_n(p) \cdot q & \text{if } p{\not\downarrow}\text{; and} \\ ET_n(p) \cdot q \cup \{(a, r) \mid r \notin C \cdot q \wedge r \text{ is normed} \wedge q \xrightarrow{a} r\} & \text{if } p{\downarrow}. \end{cases}$$

Proof. We distinguish cases according to whether $p{\downarrow}$ or $p{\not\downarrow}$:

1. Suppose that $p{\not\downarrow}$.
To see that $ET_n(p \cdot q) \subseteq ET_n(p) \cdot q$, consider an exit transition $(a, r) \in ET_n(p \cdot q)$. Then r is a normed process expression such that $p \cdot q \xrightarrow{a} r$ and $r \notin C \cdot q$. Since $p{\not\downarrow}$, it follows that there exists p' such that $p \xrightarrow{a} p'$ and $r = p' \cdot q$. From $r \notin C \cdot q$, it follows that

$p' \notin C$. Moreover, since r is normed, by Lemma 2.3(1) p' is normed too. Thereby, we have now established that $(a,p') \in ET_n(p)$, and hence $(a,r) \in ET_n(p) \cdot q$.

To see that $ET_n(p) \cdot q \subseteq ET_n(p \cdot q)$, consider and exit transition $(a,p') \in ET_n(p)$. Then p' is a normed process expression such that $p \xrightarrow{a} p'$ and $p' \notin C$. Now, since both p' and q are normed, by Lemma 2.3(1), $p' \cdot q$ is normed. Furthermore, $p \cdot q \xrightarrow{a} p' \cdot q$, and from $p' \notin C$ it follows that $p' \cdot q \notin C \cdot q$. Hence, $(a, p' \cdot q) \in ET_n(p \cdot q)$.

2. Suppose that $p\downarrow$.

   To see that $ET_n(p \cdot q) \subseteq ET_n(p) \cdot q \cup \{(a,r) \mid r \notin C \cdot q \wedge r \text{ is normed} \wedge q \xrightarrow{a} r\}$, consider an exit transition $(a,r) \in ET_n(p \cdot q)$. Then r is a normed process expression such that $p \cdot q \xrightarrow{a} r$ and $r \notin C \cdot q$. From $p \cdot q \xrightarrow{a} r$ and $p\downarrow$ it follows that either $q \xrightarrow{a} r$ or there exists p' such that $p \xrightarrow{a} p'$ and $r = p' \cdot q$. In the first case it follows that $(a,r) \in \{(a,r) \mid r \notin C \cdot q \wedge r \text{ is normed} \wedge q \xrightarrow{a} r\}$. In the second case, note that, since r is normed, by Lemma 2.3(1) so is p'. Furthermore, $p' \notin C$, for otherwise $r = p' \cdot q \in C \cdot q$ contradicting $(a,r) \in ET_n(p \cdot q)$. Hence $(a,p') \in ET_n(p)$.

   To see that $ET_n(p) \cdot q \subseteq ET_n(p \cdot q)$, consider an exit transition $(a,p') \in ET_n(p)$. Then p' is a normed process expression such that $p \xrightarrow{a} p'$ and $p' \notin C$. From $p \xrightarrow{a} p'$ it follows that $p \cdot q \xrightarrow{a} p' \cdot q$, and from $p' \notin C$ it follows that $p' \cdot q \notin C \cdot q$. Furthermore, since p' and q are both normed, by Lemma 2.3(1) so is $p' \cdot q$. Hence $(a, p' \cdot q) \in ET_n(p \cdot q)$.

   That $\{(a,r) \mid r \notin C \cdot q \wedge r \text{ is normed} \wedge q \xrightarrow{a} r\} \subseteq ET_n(p \cdot q)$ is immediate.

□

Proposition 4.10. Let C be a non-trivial strongly connected component in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$. If p_1 and p_2 are alive exit states in C, then $ET_n(p_1) = ET_n(p_2)$.

Proof. Suppose that p_1 and p_2 are alive exit states; we prove by induction on the structure of non-trivial strongly connected components in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ as given by Corollary 3.11 that $ET_n(p_1) = ET_n(p_2)$, and $p_1\downarrow$ iff $p_2\downarrow$.

If C is basic, then by Lemma 4.7 $ET_n(p_1) = \varnothing = ET_n(p_2)$, and, since p_1 and p_2 are alive exit states, it also follows from this that both $p_1\downarrow$ and $p_2\downarrow$.

Suppose that $C = C' \cdot q$, with C' a non-trivial strongly connected component, and let $p_1', p_2' \in C'$ be such that $p_1 = p_1' \cdot q$ and $p_2 = p_2' \cdot q$. Since p_1 and p_2 are alive exit states, by Lemma 4.8 so are p_1' and p_2', furthermore, q is normed. Hence, by the induction hypothesis, $ET_n(p_1') = ET_n(p_2')$ and $p_1'\downarrow$ iff $p_2'\downarrow$. From the latter it immediately follows that $p_1\downarrow$ iff $p_2\downarrow$. We now apply Lemma 4.9: if, on the one hand, $p_1\downarrow$ and $p_2\downarrow$, then

$$\begin{aligned} ET_n(p_1) &= ET_n(p_1') \cdot q \cup \{(a,r) \mid r \notin C \wedge r \text{ is normed} \wedge q \xrightarrow{a} r\} \\ &= ET_n(p_2') \cdot q \cup \{(a,r) \mid r \notin C \wedge r \text{ is normed} \wedge q \xrightarrow{a} r\} \\ &= ET_n(p_2), \end{aligned}$$

and if, on the other hand, $p_1\not\downarrow$ and $p_2\not\downarrow$, then

$$ET_n(p_1) = ET_n(p_1') \cdot q = ET_n(p_2') \cdot q = ET_n(p_2).$$ □

Theorem 4.11. $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ is less expressive than $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$.

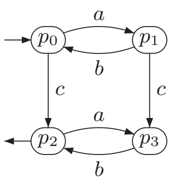


Fig. 4. A $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$-expressible automaton that is not expressible in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$.

Proof. According to Definition 2.5 we should prove that every state in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ is bisimilar to a state in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ and that there exists a state in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ for which there is no bisimilar state in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$.

That every state in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ is bisimilar to a state in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ is immediately clear.

To prove that there exists a state in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ for which there is no bisimilar state in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$, consider the $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression $\mathbf{1} \cdot (a \cdot b)^* \parallel c$. Let us use the following abbreviations:

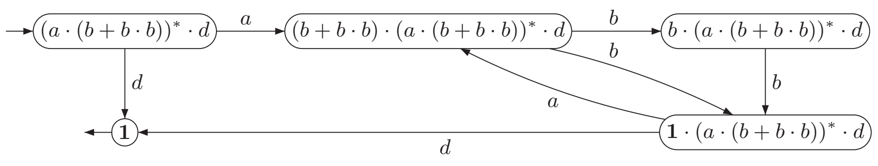
$$p_0 = \mathbf{1} \cdot (a \cdot b)^* \parallel c,$$
$$p_1 = b \cdot (a \cdot b)^* \parallel c,$$
$$p_2 = \mathbf{1} \cdot (a \cdot b)^* \parallel \mathbf{1}, \text{ and}$$
$$p_3 = b \cdot (a \cdot b)^* \parallel \mathbf{1};$$

the automaton associated with this $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression, with the states labelled with the above abbreviations, is shown in Figure 4.

To establish that there is no $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression bisimilar to $\mathbf{1} \cdot (a \cdot b)^* \parallel c$, we assume that p is a $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression bisimilar to $\mathbf{1} \cdot (a \cdot b)^* \parallel c$ and derive a contradiction. Note that the set $C = \{p_0, p_1\}$ is a non-trivial strongly connected component in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ and $p_0 \leftrightarrows p$. Hence, by Lemma 3.2, there is a strongly connected component C' in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ reachable from p satisfying the condition that there exist $p'_0, p'_1 \in C'$ such that $p_0 \leftrightarrows p'_0$ and $p_1 \leftrightarrows p'_1$. From $p_0 \leftrightarrows p'_0$ and $p_0 \xrightarrow{c} p_2$ it follows that there exists a $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression p'_2 such that $p'_0 \xrightarrow{c} p'_2$ and $p_2 \leftrightarrows p'_2$. Similarly, from $p_1 \leftrightarrows p'_1$ and $p_1 \xrightarrow{c} p_3$ it follows that there exists a $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression p'_3 such that $p'_1 \xrightarrow{c} p'_3$ and $p_3 \leftrightarrows p'_3$. It is easy to see that $p'_2 \notin C'$, for $p'_2 \in C'$ would imply the existence of a transition sequence $p'_2 \longrightarrow^* p'_0 \xrightarrow{c} p'_2$ that clearly cannot be simulated by p_2. For similar reasons, $p'_3 \notin C'$. So (b, p'_2) and (c, p'_3) are exit transitions. From $p_2 \leftrightarrows p'_2$ and $p_2\downarrow$ it follows that $p'_2\downarrow$, so (b, p'_2) is normed. From $p_3 \leftrightarrows p'_3$ and $p_3 \xrightarrow{b} p_2$ it follows that there exists p''_2 such that $p'_3 \xrightarrow{b} p''_2$ and $p_2 \leftrightarrows p''_2$, and therefore $p''_2\downarrow$, and hence (c, p'_3) is normed. We conclude that both p'_0 and p'_1 are alive exit states. By Proposition 4.10 p'_0 and p'_1 have the same normed exit transitions, so, in particular, $p'_0 \xrightarrow{c} p'_3$. Since $p_0 \leftrightarrows p'_0$, it follows that $p_2 \leftrightarrows p'_3 \leftrightarrows p_3$, and hence $p_2 \leftrightarrows p_3$, which is clearly not the case. Thus, we have now arrived at a contradiction. □

5. Relative expressiveness of $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ and $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$

Bergstra *et al.* (2001) proof that $\mathrm{PA}^*_{\delta}(\mathcal{A})$ is less expressive than $\mathrm{ACP}^*(\mathcal{A}, \gamma)$ uses the same expression as the one showing that $\mathrm{BPA}^*_{\delta}(\mathcal{A})$ is less expressive than $\mathrm{PA}^*_{\delta}(\mathcal{A})$, but it



$(a \cdot (b + b \cdot b))^* \cdot d$
a
$(b + b \cdot b) \cdot (a \cdot (b + b \cdot b))^* \cdot d$
b
$b \cdot (a \cdot (b + b \cdot b))^* \cdot d$
b
d
a
b
$\mathbf{1}$
$\mathbf{1} \cdot (a \cdot (b + b \cdot b))^* \cdot d$
d

Fig. 5. A $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$-expressible automaton with a cycle and multiple transitions going out of the cycle.

presupposes that $\gamma(c, d) = e$. It is claimed that the associated automaton fails the following general property of cycles in $\mathrm{PA}^*_\delta(\mathcal{A})$:

If C is a cycle reachable from a $\mathrm{PA}^*_\delta(\mathcal{A})$ process term and there is a state in C with a transition to a terminating state, then all other states in C have only successors in C.

The claim, however, is incorrect, as illustrated by the following example. (To avoid having to introduce the syntax and operational semantics of $\mathrm{PA}^*_\delta(\mathcal{A})$ formally, we present the example in the syntax of $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. To translate it into the syntax of $\mathrm{PA}^*_\delta(\mathcal{A})$ the occurrence of $\mathbf{1}\cdot$ should be removed, and ${}^*\cdot$ should be replaced by (the binary version of) *; we refer to Bergstra *et al.* (2001) for the operational semantics of $\mathrm{PA}^*_\delta(\mathcal{A})$.)

Example 5.1. Consider the $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ expression $(a \cdot (b + b \cdot b))^* \cdot d$ (see Figure 5 for the complete associated automaton), from which the cycle

$$C = \{\mathbf{1} \cdot (a \cdot (b + b \cdot b))^* \cdot d,\ (b + b \cdot b) \cdot (a \cdot (b + b \cdot b))^* \cdot d\}$$

is reachable. Clearly, the first expression in C can perform a d-transition to $\mathbf{1}$. Then, according to the property above, every other expression only has transitions to expressions in C. However,

$$(b + b \cdot b) \cdot (a \cdot (b + b \cdot b))^* \cdot d \xrightarrow{b} b \cdot (a \cdot (b + b \cdot b))^* \cdot d \notin C.$$

If we replace, in the property above, the notion of cycle by the notion of strongly connected component, then the resulting property does hold for $\mathrm{PA}^*_{\mathbf{0}}(\mathcal{A})$, but it still fails for $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$.

Example 5.2. Consider the $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ expression $(a \cdot b)^* \parallel c$ (see Figure 6 for the associated automaton); it gives rise to the following non-trivial strongly connected component:

$$\{\mathbf{1} \cdot (a \cdot b)^* \parallel c,\ b \cdot (a \cdot b)^* \parallel c\}.$$

The expression $\mathbf{1} \cdot (a \cdot b)^* \parallel c$ can do a c-transition to $\mathbf{1} \cdot (a \cdot b)^* \parallel \mathbf{1}$, for which the termination predicate holds, but at the same time $b \cdot (a \cdot b)^* \parallel c$ has an exit transition $(c, b \cdot (a \cdot b)^* \parallel \mathbf{1})$.

In this section, we shall establish that $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is less expressive than $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$. We use the syntactic characterization of strongly connected components in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ as given by Corollary 3.10 to conclude that a weakened version of the aforementioned property

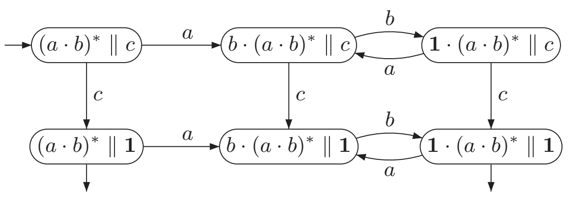


Fig. 6. A $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$-expressible automaton with a strongly connected component and multiple (different) exit transitions.

for strongly connected components holds in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. Then, we present an $\mathrm{ACP}^{*,\text{-}\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expression that does not satisfy it.

In Section 4 we deduced, from our syntactic characterization of strongly connected components in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, the property that all alive exit states of a strongly connected component have the same sets of normed exit transitions. This property may fail for strongly connected components in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$: the automaton in Figure 4 is $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$-expressible, but the alive exit states p_0 and p_1 of the strongly connected component $\{p_0, p_1\}$ have different normed exit transitions. Note, however, that these normed exit transitions both end up in another strongly connected component $\{p_2, p_3\}$. It turns out that we can relax the requirement on normed exit transitions from strongly connected components in $\mathrm{BPA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ to get a requirement that holds for strongly connected components in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. The idea is to identify exit transitions if they have the same action and end up in the same strongly connected component.

Definition 5.3. Let $\mathcal{T} = (S, \longrightarrow, \downarrow)$ be an $\mathcal{A}$-labelled transition system space. We define a binary relation $\sim$ on $\mathcal{A} \times S$ by $(a,s) \sim (a',s')$ iff $a = a'$ and s and s' are in the same strongly connected component in $\mathcal{T}$.

Since the relation of being in the same strongly connected component is an equivalence on states in a transition system space, it is clear that $\sim$ is an equivalence relation on exit transitions. The following lemma, which is valid both in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ and $\mathrm{ACP}^{*,\text{-}\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$, will give some further properties of the relation $\sim$.

Lemma 5.4. Let p, q and r be process expressions, and let a and b be actions.

1. If $(a,p) \sim (b,q)$, then $(a, p \cdot r) \sim (b, q \cdot r)$.
2. If $(a,p) \sim (b,q)$, then $(a, p \parallel r) \sim (b, q \parallel r)$.
3. If $(a,p) \sim (b,q)$, then $(a, r \parallel p) \sim (b, r \parallel q)$.

Proof. Suppose that $(a,p) \sim (b,q)$. Then $a = b$, and p and q are in the same strongly connected component of $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$. So $p \longrightarrow^* q \longrightarrow^* p$, and hence $p \cdot r \longrightarrow^* q \cdot r \longrightarrow^* p \cdot r$, $p \parallel r \longrightarrow^* q \parallel r \longrightarrow^* p \parallel r$, and $r \parallel p \longrightarrow^* r \parallel q \longrightarrow^* r \parallel p$. From this it follows that $p \cdot r$ and $q \cdot r$, $p \parallel r$ and $q \parallel r$, and $r \parallel p$ and $r \parallel q$, are in the same strongly connected component, and hence $(a, p \cdot r) \sim (b, q \cdot r)$, $(a, p \parallel r) \sim (b, q \parallel r)$, and $(a, r \parallel p) \sim (b, r \parallel q)$. □

The following notation will be convenient in the sequel: if $e = (a, p)$ is an exit transition, then $e \parallel q = (a, p \parallel q)$ and $q \parallel e = (a, q \parallel p)$. If E is a set of exit transitions E and p is a $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression, then $E \parallel p$ and $p \parallel E$ are defined by

$$E \parallel p = \{e \parallel p \mid e \in E\}, \text{ and}$$
$$p \parallel E = \{p \parallel e \mid e \in E\}.$$

For a succinct formulation of the property that will allow us to property that $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ is less expressive than $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$, it is convenient to extend the definition of normedness to strongly connected components.

Definition 5.5. A strongly connected component C is *normed* if all states in C are *normed*.

The following lemma explains how extending the definition of normedness to strongly connected components is conducive to succinct formulation: to ensure that a strongly connected component has alive exit states, it suffices to require that it is normed.

Lemma 5.6. Let C be a strongly connected component. Then C is normed iff it has alive exit states.

Proof. Note that if C contains a normed state, then C is normed. Hence, since an alive exit state is clearly normed, the implication from right to left is immediate. It remains to establish the implication from left to right. To this end, let $s \in C$. Since C is normed, there exists a process expression s' such that $s \longrightarrow^* s'$ and $s'\downarrow$. On the one hand, if $s' \in C$, then s' is an alive exit state. On the other hand, if $s' \notin C$, then there exist states $s_1 \in C$ and $s_2 \notin C$ and an action a such that $s \longrightarrow^* s_1 \xrightarrow{a} s_2 \longrightarrow^* s'$. Then, clearly, (a, s_2) is a normed exit transition from s_1, so s_1 is an alive exit state. $\square$

We proceed, in the following lemma, to characterize the set of normed exit transitions in a normed strongly connected component of the form $C_1 \parallel C_2$ in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ in terms of the normed exit transitions in the constituent normed strongly connected components C_1 and C_2.

Lemma 5.7. Let C_1 and C_2 be normed strongly connected components in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$. Then $C_1 \parallel C_2$ is a normed strongly connected component, and for all $p \in C_1$ and $q \in C_2$

$$ET_n(p \parallel q) = (ET_n(p) \parallel q) \cup (p \parallel ET_n(q)).$$

Proof. Suppose that C_1 and C_2 are normed strongly connected components. Then, by Corollary 3.9, $C_1 \parallel C_2$ is a strongly connected component too, which, by Lemma 2.3(2), is normed. It remains to prove that for all $p \in C_1$ and $q \in C_2$

$$ET_n(p \parallel q) = (ET_n(p) \parallel q) \cup (p \parallel ET_n(q)).$$

To prove that $ET_n(p \parallel q) \subseteq (ET_n(p) \parallel q) \cup (p \parallel ET_n(q))$, consider an arbitrary $(a, r) \in ET_n(p \parallel q)$. Then r is a normed $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression such that $p \parallel q \xrightarrow{a} r$ and $r \notin C_1 \parallel C_2$. From the operational rules for $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ with $\parallel$ in the conclusion it follows that we can distinguish two cases: either there exists p' such that $r = p' \parallel q$ and $p \xrightarrow{a} p'$, or there exists q' such that $r = p \parallel q'$ and $q \xrightarrow{a} q'$. The proofs for these cases are entirely analogous; we only present details for the first case. Since $q \in C_2$, $p' \parallel q \notin C_1 \parallel C_2$ implies that $p' \notin C_1$, and, since r is normed and $r = p' \parallel q$, by Lemma 2.3(1) p' is normed too. It follows that $(a, p') \in ET_n(p)$, and hence $(a, r) \in ET_n(p) \parallel q$.

To prove that $(ET_n(p) \parallel q) \subseteq ET_n(p \parallel q)$ consider an arbitrary $(a, r) \in ET_n(p) \parallel q$. Then there exists a $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression p' such that $(a, p') \in ET_n(p)$ and $r = p' \parallel q$. From $(a, p') \in ET_n(p)$ it follows that p' is a normed $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression such that $p \xrightarrow{a} p'$ and $p' \notin C_1$. From the latter, it follows that $p' \not\longrightarrow^* p$ and hence, by Lemma 2.2(3), $p' \parallel q \not\longrightarrow^* p \parallel q$, so $p' \parallel q \notin C_1 \parallel C_2$. Further note that, by our assumption that C_2 is normed, q is normed, so, by Lemma 2.3(2), $p' \parallel q$ is normed. We conclude that $(a, p' \parallel q) \in ET_n(p \parallel q)$.

The proof that $(p \parallel ET_n(q)) \subseteq ET_n(p \parallel q)$ is completely analogous to the proof that $(ET_n(p) \parallel q) \subseteq ET_n(p \parallel q)$, so the proof of the lemma is now complete. □

To formulate the special property of strongly connected components in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ that will allow us to prove that some $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ expressions do not have a counterpart in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$, we need the following notion of maximal alive exit state.

Definition 5.8. Let $\mathcal{T} = (S, \longrightarrow, \downarrow)$ be an $\mathcal{A}$-labelled transition system space, let $\sim$ be the equivalence relation on $\mathcal{A} \times S$ associated with $\mathcal{T}$ according to Definition 5.3, let C be a strongly connected component in $\mathcal{T}$, and let $s \in C$ be an alive exit state. We say that s is *maximal* (modulo $\sim$) if for all alive exit states $s' \in C$ and for all $e' \in ET_n(s')$ there exists an exit transition $e \in ET_n(s)$ such that $e \sim e'$.

We are now in a position to establish the property with which we shall prove that $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ is less expressive than $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$.

Proposition 5.9. Every normed strongly connected component in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ has a maximal alive exit state.

Proof. Let C be a normed strongly connected component in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$. If C is trivial, then C is a singleton, say $C = \{p_m\}$, and, since C is normed, by Lemma 5.6 p_m is an alive exit state that is, vacuously, maximal. So, for the remainder of the proof we may assume that C is non-trivial. We proceed to prove with induction on the structure of C, as given by Corollary 3.10, that C has a maximal alive exit state p_m such that, in addition, $p_m\downarrow$ if $p\downarrow$ for some $p \in C$. We distinguish three cases:

1. If C is basic, then, by Lemma 4.7, $ET_n(p) = \emptyset$ for all $p \in C$. Since C is normed, by Lemma 5.6 it contains an alive exit state, say p_m, which is then vacuously maximal and satisfies $p_m\downarrow$.
2. Suppose that $C = C' \cdot q$ with C' a non-trivial strongly connected component and q a $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ expression. Since C is normed, by Lemma 5.6 it contains an alive exit state, say $p' \cdot q$ with $p' \in C'$; then, by Lemma 4.8, p' is an alive exit state in C' and q is normed. Hence, by the induction hypothesis, C' has a maximal alive exit state $p_m \in C'$ such that $p_m\downarrow$ if $p\downarrow$ for some $p \in C'$. By Lemma 4.8 $p_m \cdot q$ is an alive exit state in C. If $p \cdot q\downarrow$ for some $p \in C'$, then $p\downarrow$ and $q\downarrow$, so $p_m\downarrow$, and hence $p_m \cdot q\downarrow$.
   For this case it therefore remains to prove that $p_m \cdot q$ is maximal. To this end, consider an alive exit state $p \cdot q \in C$ and let (a, r) be a normed exit transition from $p \cdot q$. Then

by Lemma 4.9 either there exist p' such that $r = p' \cdot q$ and (a, p') is a normed exit transition from p, or $p\downarrow$ and $q \xrightarrow{a} r$. In the first case, since p_m is maximal, there exists a $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ expression p'_m such that (a, p'_m) is a normed exit transition from p_m and $(a, p'_m) \sim (a, p')$; it follows by Lemma 4.9 that $(a, p'_m \cdot q)$ is a normed exit transition from $p_m \cdot q$, and by Lemma 5.4 $(a, p'_m \cdot q) \sim (a, r)$. In the second case, since $p\downarrow$ implies $p_m\downarrow$, (a, r) is a normed exit transition from $p_m \cdot q$.

3. Suppose that $C = C_1 \parallel C_2$ with C_1 and C_2 strongly connected components. Since C is normed, by Lemma 2.3(2) both C_1 and C_2 are normed. Hence, by the induction hypothesis C_1 has a maximal alive exit state p_m and C_2 has a maximal alive exit state q_m. Moreover, also by the induction hypothesis, $p_m\downarrow$ if $p\downarrow$ for some $p \in C_1$ and $q_m\downarrow$ if $q\downarrow$ for some $q \in C_2$. We argue that $p_m \parallel q_m$ is a maximal alive exit state in C, and that whenever $p \parallel q\downarrow$ for some $p \parallel q \in C$, then also $p_m \parallel q_m\downarrow$.
Note that if $p \parallel q\downarrow$ for some $p \parallel q \in C$, then, according to operational rule 14, $p\downarrow$ and $q\downarrow$, so $p_m\downarrow$ and $q_m\downarrow$, and hence $p_m \parallel q_m\downarrow$.
To see that $p_m \parallel q_m$ is maximal, suppose that $e \in ET_n(p \parallel q)$ for some $p \parallel q \in C$. Then by Lemma 5.7 there are two cases: either there exists $e' \in ET_n(p)$ such that $e = e' \parallel q$, or there exists $e' \in ET_n(q)$ such that $e = p \parallel e'$. In the first case, if $e' \in ET_n(p)$, then by the induction hypothesis there exists $e'' \in ET_n(p_m)$ such that $e'' \sim e'$. By Lemma 5.7, $e'' \parallel q_m \in ET_n(p_m \parallel q_m)$ and since q_m and q are in the same strongly connected component C_2, it is clear that $e'' \parallel q_m \sim e$. In the second case, if $e' \in ET_n(p_m)$ we find by an analogous reasoning e'' such that $p_m \parallel e'' \in ET_n(p_m \parallel q_m)$ and $p_m \parallel e'' \sim e$.

□

We can now prove the main result of this section.

Theorem 5.10. $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is less expressive than $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$.

Proof. According to Definition 2.5, we should prove that every state in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is bisimilar to a state in $\mathrm{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ and that there exists a state in $\mathrm{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ for which there is no bisimilar state in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$.

That every state in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is bisimilar to a state in $\mathrm{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ is immediate since $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$ is included in $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ as $\mathrm{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \emptyset)$.

To prove that there exists a state in $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ for which there is no bisimilar state in $\mathrm{PA}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A})$, consider the $\mathrm{ACP}^{*,-\partial}_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ expression $\mathbf{1} \cdot (a \cdot b)^* \cdot d \parallel c$ with γ satisfying $\gamma(b, c) = \gamma(c, b) = e$ and γ undefined everywhere else. Let us use the following abbreviations:

$$p_0 = \mathbf{1} \cdot (a \cdot b)^* \cdot d \parallel c,$$
$$p_1 = \mathbf{1} \cdot b \cdot (a \cdot b)^* \cdot d \parallel c,$$
$$p_2 = \mathbf{1} \cdot (a \cdot b)^* \cdot d \parallel \mathbf{1},$$
$$p_3 = \mathbf{1} \cdot b \cdot (a \cdot b)^* \cdot d \parallel \mathbf{1},$$
$$p_4 = \mathbf{1} \parallel c, \text{ and}$$
$$p_5 = \mathbf{1} \parallel \mathbf{1};$$

the automaton associated with p_0, with the states labelled with the above abbreviations, is shown in Figure 7.

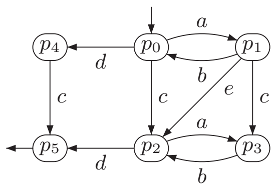


Fig. 7. An $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$-expressible automaton that is not expressible in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$.

To establish that there is no $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression bisimilar to p_0, we assume that p is such a $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression and derive a contradiction. Note that the set $C = \{p_0, p_1\}$ is a strongly connected component in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. Hence, since $p_0 \mathrel{\underline{\leftrightarrow}} p$, by Lemma 3.2 there is a strongly connected component C' in $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ reachable from p satisfying the condition that there exist $p_0', p_1' \in C'$ such that $p_0 \mathrel{\underline{\leftrightarrow}} p_0'$ and $p_1 \mathrel{\underline{\leftrightarrow}} p_1'$. Clearly, C' is normed, so by Proposition 5.9 it has a maximal alive exit state. From $p_0 \mathrel{\underline{\leftrightarrow}} p_0'$ it follows that there exists $p_4' \notin C'$ such that $p_0 \xrightarrow{d} p_4'$ and $p_4 \mathrel{\underline{\leftrightarrow}} p_4'$. From $p_1 \mathrel{\underline{\leftrightarrow}} p_1'$ it follows that there exists $p_2' \notin C'$ such that $p_1' \xrightarrow{e} p_2'$ and $p_2 \mathrel{\underline{\leftrightarrow}} p_2'$. So, on the one hand, both p_0' and p_1' are alive exit states. On the other hand, p_0' does not have a normed exit transition labelled with e and p_1 does not have a normed exit transition labelled with d. It is, moreover, easy to see that every state $p' \in C'$ is bisimilar to either p_0' or p_1'. We conclude that C' does not have a maximal alive exit state, and thus arrive at a contradiction. We conclude that p_0 is not $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$-expressible. □

6. Relative expressiveness of $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ and $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$

In the next section, we shall prove that every finite automaton is expressible in $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. The purpose of this section is to prove that encapsulation is actually an essential ingredient in this result: we shall prove that $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ is less expressive than $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, which extends $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ with encapsulation. The proof proceeds by an adaptation of the proof of the previous section. It suffices to relax the property stating the existence of a maximal alive exit state, which, as we have seen in Figure 7, fails for strongly connected components in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. In $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ there does not necessarily exist an alive exit state that, up to $\sim$, has *all* the exit transitions of all other alive exit states. But there does exist a special alive exit state that, up to $\sim$, always shares *at least one* exit transition with every other alive exit state.

Operational rule 15 gives rise to additional transitions from parallel compositions. As a consequence, the characterization of the set of normed exit transitions in a normed strongly connected component of the form $C_1 \parallel C_2$ in Lemma 5.7 does not hold for normed strongly connected components in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. But we do have the following weaker property.

Lemma 6.1. Let C_1 and C_2 be normed strongly connected components in $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. Then $C_1 \parallel C_2$ is a normed strongly connected component, and for all $p \in C_1$ and $q \in C_2$

$$(ET_n(p) \parallel q) \cup (p \parallel ET_n(q)) \subseteq ET_n(p \parallel q);\ \text{and}$$
$$ET_n(p) \cup ET_n(q) = \emptyset \ \text{iff}\ ET_n(p \parallel q) = \emptyset.$$

Proof. Suppose that C_1 and C_2 are strongly connected components with alive exit states. Then, by Proposition 3.5, $C_1 \parallel C_2$ is a strongly connected component too, which, by Lemma 2.3(2), is normed. So it remains to prove that for all $p \in C_1$ and $q \in C_2$

$$(ET_n(p) \parallel q) \subseteq ET_n(p \parallel q);$$
$$(p \parallel ET_n(q)) \subseteq ET_n(p \parallel q);\ \text{and}$$
$$ET_n(p) \cup ET_n(q) = \emptyset \ \text{iff}\ ET_n(p \parallel q) = \emptyset.$$

To prove that $(ET_n(p) \parallel q) \subseteq ET_n(p \parallel q)$ consider an arbitrary $(a,r) \in ET_n(p) \parallel q$. Then there exists an $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression p' such that $(a,p') \in ET_n(p)$ and $r = p' \parallel q$. From $(a,p') \in ET_n(p)$ it follows that p' is a normed $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression such that $p \xrightarrow{a} p'$ and $p' \notin C_1$. From the latter, it follows that $p' \not\longrightarrow^{*} p$ and hence, by Lemma 2.2(3), $p' \parallel q \not\longrightarrow^{*} p \parallel q$, so $p' \parallel q \notin C_1 \parallel C_2$. Note that, since C_2 is normed, q is normed, so, by Lemma 2.3(2), $p' \parallel q$ is normed. We conclude that $(a, p' \parallel q) \in ET_n(p \parallel q)$.

The proof that $(p \parallel ET_n(q)) \subseteq ET_n(p \parallel q)$ is completely analogous to the proof that $(ET_n(p) \parallel q) \subseteq ET_n(p \parallel q)$.

Note that from $(ET_n(p) \parallel q) \subseteq ET_n(p \parallel q)$ and $(p \parallel ET_n(q)) \subseteq ET_n(p \parallel q)$ it immediately follows that if $ET_n(p \parallel q) = \emptyset$, then $ET_n(p) = \emptyset$ and $ET_n(q) = \emptyset$. So, it remains to prove that if $ET_n(p) = \emptyset$ and $ET_n(q) = \emptyset$, then $ET_n(p \parallel q) = \emptyset$. We proceed by contraposition and assume $ET_n(p \parallel q) \neq \emptyset$. Then there is $(a,r) \in ET_n(p \parallel q)$, so r is a normed $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression such that $p \parallel q \xrightarrow{a} r$ and $r \notin C_1 \parallel C_2$. From the operational rules for $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ with $\parallel$ in the conclusion it follows that we can distinguish three cases: either there exists p' such that $r = p' \parallel q$ and $p \xrightarrow{a} p'$, or there exists q' such that $r = p \parallel q'$ and $q \xrightarrow{a} q'$, or there exists p' and q' such that $r = p' \parallel q'$, $p \xrightarrow{b} p'$ and $q \xrightarrow{c} q'$ and $\gamma(b,c) = a$.

In the first case, since $q \in C_2$, $p' \parallel q \notin C_1 \parallel C_2$ implies that $p' \notin C_1$, and, since r is normed and $r = p' \parallel q$, by Lemma 2.3(2) p' is normed too. It follows that $(a,p') \in ET_n(p)$, and hence $ET_n(p) \neq \emptyset$.

In the second case, by an analogous argument as in the first case, $(a,q') \in ET_n(q)$, and hence $ET_n(q) \neq \emptyset$.

Finally, in the third case, as $p' \parallel q' \notin C_1 \parallel C_2$, either $p' \notin C_1$ or $q' \notin C_2$. Without loss of generality, we can assume that $p' \notin C_1$. Since r is normed and $r = p' \parallel q'$, by Lemma 2.3(2) p' is normed too. It follows that $(a,p') \in ET_n(p)$, and hence $ET_n(p) \neq \emptyset$. □

To formulate the property of strongly connected components in $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ that will allow us to prove that some $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions do not have a counterpart in $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$, we use the following notion.

Definition 6.2. Let $\mathcal{T} = (S, \longrightarrow, \downarrow)$ be an $\mathcal{A}$-labelled transition system space, let $\sim$ be the equivalence relation on $\mathcal{A} \times S$ associated with $\mathcal{T}$ according to Definition 5.3, let C be a strongly connected component in $\mathcal{T}$, and let $s \in C$ be an alive exit state. We say that s is *dominating* (modulo $\sim$) if for every alive exit state $s' \in C$ such that $ET_n(s') \neq \emptyset$ there exist $e \in ET_n(s)$ and $e' \in ET_n(s')$ such that $e \sim e'$.

Note that if an alive exit state is maximal, then it is also dominating.

The following proposition establishes the property with which we shall prove that $\text{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ is less expressive than $\text{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$.

Proposition 6.3. If C is a normed strongly connected component in $\text{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$, then C has a dominating alive exit state.

Proof. Suppose that C is a normed strongly connected component in $\text{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$. If C is trivial, then C is a singleton, say $C = \{p_d\}$, and, since C is normed, by Lemma 5.6 p_d is an alive exit state that is, vacuously, dominating. So, for the remainder of the proof we may assume that C is non-trivial. We proceed to prove with induction on the structure of C, as given by Proposition 3.8, that C has a dominating alive exit state p_d such that, in addition, $p_d\downarrow$ if $p\downarrow$ for some $p \in C$. We distinguish three cases:

1. If C is basic, then, by Lemma 4.7, $ET_n(p) = \emptyset$ for all $p \in C$. Since C is normed, by Lemma 5.6 it contains an alive exit state, say p_d, which is then vacuously dominating and satisfies $p_d\downarrow$.
2. Suppose that $C = C' \cdot q$ with C' a non-trivial strongly connected component and q an $\text{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ expression. Since C is normed, by Lemma 5.6 it contains an alive exit state, say $p' \cdot q$ with $p' \in C'$; then, by Lemma 4.8, p' is an alive exit state in C' and q is normed. Hence, by the induction hypothesis, C' has a dominating alive exit state $p_d \in C'$ such that $p_d\downarrow$ if $p\downarrow$ for some $p \in C'$. By Lemma 4.8, $p_d \cdot q$ is an alive exit state in C. If $p \cdot q\downarrow$ for some $p \in C'$, then $p\downarrow$ and $q\downarrow$, so $p_d\downarrow$, and hence $p_d \cdot q\downarrow$.
   For this case it therefore remains to prove that $p_d \cdot q$ is dominating. To this end, consider an alive exit state $p \cdot q \in C$ such that $ET_n(p \cdot q) \neq \emptyset$. We distinguish two cases:
   On the one hand, if $ET_n(p) = \emptyset$, then, according to Lemma 4.9, $p\downarrow$ and
   $$ET_n(p \cdot q) = \{(a, r) \mid r \notin C \wedge r \text{ is normed} \wedge q \xrightarrow{a} r\}.$$
   Since $ET_n(p \cdot q) \neq \emptyset$, there exists r such that $r \notin C$, r is normed, and $q \xrightarrow{a} r$. Since, by the induction hypothesis, $p_d\downarrow$, it follows that $p_d \cdot q \xrightarrow{a} r$, so $(a, r) \in ET_n(p_d \cdot q)$, and clearly $(a, r) \sim (a, r)$.
   On the other hand, if $ET_n(p) \neq \emptyset$, then, by the induction hypothesis, there exist $e \in ET_n(p_d)$ and $e' \in ET_n(p)$ such that $e \sim e'$. By Lemma 4.9, $e \cdot q \in ET_n(p_d \cdot q)$ and $e' \cdot q \in ET_n(p \cdot q)$, and, by Lemma 5.4, $e \cdot q \sim e' \cdot q$.
3. Suppose that $C = C_1 \parallel C_2$ with C_1 and C_2 strongly connected components. Since C is normed, by Lemma 2.3(2), both C_1 and C_2 are normed. Hence, by the induction hypothesis, C_1 has a dominating alive exit state p_d and C_2 has a dominating alive exit state q_d. Moreover, also by the induction hypothesis, $p_d\downarrow$ if $p\downarrow$ for some $p \in C_1$, and

$q_d\!\downarrow$ if $q\!\downarrow$ for some $q \in C_2$. We argue that $p_d \parallel q_d$ is a dominating alive exit state in C, and that if $p \parallel q\!\downarrow$ for some $p \parallel q \in C$, then $p_d \parallel q_d\!\downarrow$.

Note that if $p \parallel q\!\downarrow \in C$, then, according to operational rule 14, $p\!\downarrow$ and $q\!\downarrow$, so $p_d\!\downarrow$ and $q_d\!\downarrow$, and hence $p_d \parallel q_d\!\downarrow$.

To prove that $p_d \parallel q_d$ is a dominating alive exit state, let $p \parallel q$ be an alive exit state in C such that $ET_n(p \parallel q) \neq \emptyset$. Then, by Lemma 6.1, at least one of $ET_n(p)$ and $ET_n(q)$ is nonempty; without loss of generality, we can assume that $ET_n(p) \neq \emptyset$. Since p_d is a dominating alive exit state in C_1, there exist $(a, p'_d) \in ET_n(p_d)$ and $(a, p') \in ET_n(p)$ such that $(a, p'_d) \sim (a, p')$.

Note that, by Lemma 6.1, on the one hand, $(a, p'_d \parallel q_d) \in ET_n(p_d \parallel q_d)$, and, on the other hand, $(a, p' \parallel q) \in ET_n(p \parallel q)$. Moreover, from $(a, p'_d) \sim (a, p')$ it follows that p'_d and p' are in the same strongly connected component, and hence, since also q_d and q are in the same strongly connected component, $p'_d \parallel q_d$ and $p' \parallel q$ are in the same strongly connected component. Therefore, $(a, p'_d \parallel q_d) \sim (a, p' \parallel q)$.

□

We can now prove the main result of this section.

Theorem 6.4. $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ is less expressive than $\bigcup_\gamma \mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$.

Proof. We have to prove that every state in $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ is bisimilar to a state in $\bigcup_\gamma \mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ and that there exists a state in $\bigcup_\gamma \mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ for which there is no bisimilar state in $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$.

That every state in $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ is bisimilar to a state in $\bigcup_\gamma \mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ is immediate.

It remains to prove that there exists a state in $\bigcup_\gamma \mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ for which there is no bisimilar state in $\bigcup_\gamma \mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$. To this end, we define

$$C = \{enter_i, leave_{\alpha,i} \mid 0 \leqslant i \leqslant 2, \alpha \in \{a, b, c, d\}\};$$

we define a communication function γ such that

$$\begin{aligned}
&\gamma(enter_0, leave_{b,0}) = \gamma(leave_{b,0}, enter_0) = b,\\
&\gamma(enter_1, leave_{a,1}) = \gamma(leave_{a,1}, enter_1) = a,\\
&\gamma(enter_2, leave_{c,2}) = \gamma(leave_{c,2}, enter_2) = c,\\
&\gamma(enter_2, leave_{d,2}) = \gamma(enter_2, leave_{d,2}) = d,
\end{aligned}$$

and γ is undefined otherwise; and we consider the $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A}, \gamma)$ expressions

$$\begin{aligned}
&s_0 = \partial_C(p'_0 \parallel p_1 \parallel p_2),\\
&s_1 = \partial_C(p_0 \parallel p'_1 \parallel p_2), \text{ and}\\
&s_2 = \partial_C(p_0 \parallel p_1 \parallel p_2),
\end{aligned}$$

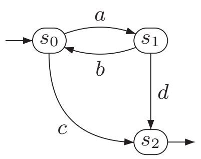


Fig. 8. An $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$-expressible automaton that is not expressible in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$.

in which

$$\begin{aligned}
p_0 &= \mathbf{1}\cdot(\mathit{enter}_0\cdot(\mathit{leave}_{a,1}+\mathit{leave}_{c,2}))^*,\\
p_0' &= (\mathit{leave}_{a,1}+\mathit{leave}_{c,2})\cdot(\mathit{enter}_0\cdot(\mathit{leave}_{a,1}+\mathit{leave}_{c,2}))^*,\\
p_1 &= \mathbf{1}\cdot(\mathit{enter}_1\cdot(\mathit{leave}_{b,0}+\mathit{leave}_{d,1}))^*,\\
p_1' &= (\mathit{leave}_{b,0}+\mathit{leave}_{d,1})\cdot(\mathit{enter}_1\cdot(\mathit{leave}_{b,0}+\mathit{leave}_{d,1}))^*,\ \text{and}\\
p_2 &= \mathbf{1}\cdot(\mathit{enter}_2\cdot\mathbf{1})^*.
\end{aligned}$$

The automaton associated with s_0 is shown in Figure 8, with the states labelled with the above abbreviations.

To establish that there is no $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression bisimilar to s_0, we assume that s is such a $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression and derive a contradiction. Note that the set $C=\{s_0,s_1\}$ is a strongly connected component in $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. Hence, since $s_0 \mathrel{\underline{\leftrightarrow}} s$, by Lemma 3.2 there is a strongly connected component C' in $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ reachable from s satisfying the condition that there exist $s_0', s_1' \in C'$ such that $s_0 \mathrel{\underline{\leftrightarrow}} s_0'$ and $s_1 \mathrel{\underline{\leftrightarrow}} s_1'$. By Proposition 6.3, C' has a dominating alive exit state. From $s_0 \mathrel{\underline{\leftrightarrow}} s_0'$ it follows that there exists $s_2' \notin C'$ such that $s_0' \xrightarrow{c} s_2'$ and $s_2 \mathrel{\underline{\leftrightarrow}} s_2'$. From $s_1 \mathrel{\underline{\leftrightarrow}} s_1'$ it follows that there exists $s_2'' \notin C'$ such that $s_1' \xrightarrow{d} s_2''$ and $s_2 \mathrel{\underline{\leftrightarrow}} s_2''$. So, on the one hand, both s_0' and s_1' are alive exit states with normed exit transitions. Now, on the one hand, s_0' has a normed exit transition labelled with c, but it does not have a normed exit transition labelled with d, and, on the other hand, s_1' does have a normed exit transition labelled with d, but it does not have a normed exit transition labelled with c. It follows that s_0' does not dominate s_1', and, on the same grounds, s_1' does not dominate s_0'. Since all states in C' are bisimilar to either s_0' or s_1', we conclude that C' does not have a dominating exit state, and hence s_0 is not $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$-expressible. □

7. Every finite automaton is $\mathrm{ACP}^{*}_{\mathbf{0,1}}(\mathcal{A},\gamma)$-expressible

Milner (1984) observed that there exist finite automata that are not bisimilar to a finite automaton associated with a $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression. Our proof of Theorem 4.11 has Milner's observation as an immediate consequence: the finite automaton associated with the $\mathrm{PA}^{*}_{\mathbf{0,1}}(\mathcal{A})$ expression used in the proof (see Figure 3) is not $\mathrm{BPA}^{*}_{\mathbf{0,1}}(\mathcal{A})$-expressible. Similarly, by Theorem 5.10, there are finite automata that are not expressible

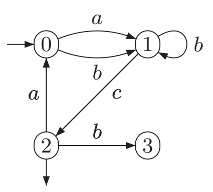


Fig. 9. A finite automaton.

in $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$, and by Theorem 6.4 there are finite automata that are not expressible in $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$.

In this section, we shall prove that every finite automaton *is* expressible in $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, for suitable choices of $\mathcal{A}$ and γ, even up to isomorphism. Before we formally prove the result, let us first explain the idea informally and illustrate it with an example. The $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression that we shall associate with a finite automaton will have one parallel component per state of the automaton, representing the behaviour in that state (i.e., which outgoing transitions it has to which other states and whether it is terminating). At any time, one of those parallel components, the one corresponding with the 'current state,' has control. An a-transition from that current state to a next state corresponds with a communication between two components. We will make essential use of $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$'s facility to let the action a be the result of communication.

Example 7.1. Consider the finite automaton in Figure 9.

We associate with every state $i \in \{0,1,2,3\}$ an $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression p_i as follows:

$$\begin{aligned}
p_0 &= (\mathit{enter}_0 \cdot (\mathit{leave}_{a,1} + \mathit{leave}_{b,1}))^*,\\
p_1 &= (\mathit{enter}_1 \cdot b^* \cdot (\mathit{leave}_{c,2}))^*,\\
p_2 &= (\mathit{enter}_2 \cdot (\mathit{leave}_{a,0} + \mathit{leave}_{b,3} + \mathbf{1}))^*,\\
p_3 &= (\mathit{enter}_3 \cdot \mathbf{0})^*.
\end{aligned}$$

Every p_i has an enter_i transition to gain control, and by executing a $\mathit{leave}_{\alpha,j}$ it may then release control to p_j with action α ($\alpha \in \{a,b,c\}$) as effect. We define the communication function so that an enter_i action communicates with a $\mathit{leave}_{\alpha,i}$ action, resulting in the action α. That is, in this particular case,

$$\begin{aligned}
\gamma(\mathit{enter}_0, \mathit{leave}_{a,0}) &= \gamma(\mathit{leave}_{a,0}, \mathit{enter}_0) = a\\
\gamma(\mathit{enter}_1, \mathit{leave}_{a,1}) &= \gamma(\mathit{leave}_{a,1}, \mathit{enter}_1) = a\\
\gamma(\mathit{enter}_1, \mathit{leave}_{b,1}) &= \gamma(\mathit{leave}_{b,1}, \mathit{enter}_1) = b\\
\gamma(\mathit{enter}_2, \mathit{leave}_{c,2}) &= \gamma(\mathit{leave}_{c,2}, \mathit{enter}_2) = c\\
\gamma(\mathit{enter}_3, \mathit{leave}_{b,3}) &= \gamma(\mathit{leave}_{b,3}, \mathit{enter}_3) = b,
\end{aligned}$$

and γ is undefined otherwise. Loops in the automaton (such as the loop on state 1) require special treatment as they should not have the effect of releasing control to some other state.

Let p_0' be the result of executing the $enter_0$-transition from p_0, i.e.,

$$p_0' = \mathbf{1} \cdot (leave_{a,1} + leave_{b,1}) \cdot p_0.$$

We define the $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ expression that simulates the finite automaton in Figure 9 as the parallel composition of p_0', p_1, p_2 and p_3, encapsulating the control actions $enter_i$ and $leave_{\alpha,i}$, i.e., as

$$\partial_{\{enter_i, leave_{\alpha,i} \mid i \in \{0,1,2,3\},\ \alpha \in \{a,b,c\}\}}(p_0' \parallel p_1 \parallel p_2 \parallel p_3).$$

We now present the technique illustrated in the preceding example in full generality. Let $\mathcal{F} = (S, \longrightarrow, \uparrow, \downarrow)$ be a finite automaton. We shall associate with $\mathcal{F}$ an $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ expression $p_{\mathcal{F}}$ that has one parallel component p_s for every state s in S. To allow a parallel component to gain and release control, we use a collection of *control actions* C, assumed to be disjoint from A, and defined as

$$C = \{enter_s \mid s \in S\} \cup \{leave_{\alpha,t} \mid s \in S,\ \alpha \in A\}.$$

Gaining and releasing control is modelled by the communication function γ on $A \cup C$ satisfying:

$$\gamma(enter_s, leave_{\alpha,t}) = \gamma(leave_{\alpha,t}, enter_s) = \begin{cases} \alpha & \text{if } s = t\text{; and} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

For the specification of the $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ expressions p_s we need one more definition: for $s, t \in S$ we denote by $A_{s,t}$ the set of actions occurring as the label on a transition from s to t, i.e.,

$$A_{s,t} = \{\alpha \mid s \xrightarrow{\alpha} t\}.$$

Now we can specify the $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ expressions p_s ($s \in S$) by

$$p_s = \mathbf{1} \cdot \Big(enter_s \cdot \Big(\sum_{\alpha \in A_{s,s}} \alpha \Big)^* \cdot \Big(\sum_{\substack{t \in S \\ t \neq s}} \sum_{\alpha \in A_{s,t}} leave_{\alpha,s}\ [+\ \mathbf{1}]_{s\downarrow} \Big) \Big)^* .$$

(We use the notation $\sum_{P(x)} p(x)$, with $P(x)$ a unary predicate and $p(x)$ a process expression parametrized in x, to abbreviate the alternative composition of the elements of the set $\{p(x) \mid P(x)\}$, implicitly assuming that this set is finite. Let $\{p(x) \mid P(x)\} = \{p_1, \dots, p_n\}$; then $\sum_{P(x)} p(x) = p_1 + \cdots + p_n$, adopting the convention that if the set $\{p(x) \mid P(x)\}$ is empty, then $\sum_{P(x)} p(x) = \mathbf{0}$.) By $[+\ \mathbf{1}]_{s\downarrow}$ we mean that the summand $+\ \mathbf{1}$ is optional; it is only included if $s\downarrow$. The empty summation denotes $\mathbf{0}$. (We let p_s start with $\mathbf{1}$ to later get that the finite automaton associated with the $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A}, \gamma)$ expression for $\mathcal{F}$ is isomorphic and not just bisimilar with $\mathcal{F}$.)

Note that, in $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, every p_s has a unique outgoing transition; specifically $p_s \xrightarrow{enter_s} p'_s$, where p'_s denotes:

$$p'_s = \left(\mathbf{1}\cdot\left(\sum_{\alpha\in A_{s,s}}\alpha\right)^*\cdot\left(\sum_{\substack{t\in S\\ t\neq s}}\sum_{\alpha\in A_{s,t}} leave_{\alpha,s}\ [+\ \mathbf{1}]_{s\downarrow}\right)\right)\cdot p_s\ .$$

Now suppose that $S = \{s_0,\ldots,s_n\}$ with $\uparrow = s_0$; then we define $p_{\mathcal{F}}$ by

$$p_{\mathcal{F}} = \partial_C(p'_{s_0} \parallel p_{s_1} \parallel \cdots \parallel p_{s_n}).$$

Clearly, the construction of $p_{\mathcal{F}}$ works for every finite automaton $\mathcal{F}$. The bijection defined by $s_i \mapsto \partial_C(p_{s_0} \parallel \cdots \parallel p_{s_{i-1}} \parallel p'_{s_i} \parallel p_{s_{i+1}} \parallel \cdots \parallel p_{s_n})$ is an isomorphism from $\mathcal{F}$ to the automaton associated with $p_{\mathcal{F}}$ by the operational semantics. We shall refer to $p_{\mathcal{F}}$ as the $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression associated with $\mathcal{F}$.

Theorem 7.2. Let $\mathcal{F}$ be a finite automaton, and let $p_{\mathcal{F}}$ be its associated $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression. The automaton associated with $p_{\mathcal{F}}$ by the operational rules for $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ is isomorphic to $\mathcal{F}$.

Proof. The bijection defined by $s_i \mapsto \partial_C(p_{s_0} \parallel \cdots \parallel p_{s_{i-1}} \parallel p'_{s_i} \parallel p_{s_{i+1}} \parallel \cdots \parallel p_{s_n})$ is an isomorphism from $\mathcal{F}$ to the automaton associated to $p_{\mathcal{F}}$ by the operational semantics in $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$. □

Corollary 7.3. For every finite automaton $\mathcal{F}$ there exists an instance of $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ with a suitable finite set of actions $\mathcal{A}$ and a handshaking communication function γ such that $\mathcal{F}$ is $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$-expressible up to isomorphism.

8. Conclusion

In this paper, we have investigated the effect on the expressiveness of regular expressions modulo bisimilarity if different forms of parallel composition are added. We have established an expressiveness hierarchy that can be briefly summarized as:

$$\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A}) \prec \mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A}) \prec \bigcup_{\gamma}\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma) \prec \bigcup_{\gamma}\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma).$$

Furthermore, while not every finite automaton can be expressed modulo bisimilarity with a regular expression, it suffices to add a form of $\mathrm{ACP}(\mathcal{A},\gamma)$-style parallel composition, with handshaking communication and encapsulation, to get a language that is sufficiently expressive to express all finite automata modulo bisimilarity. These results should be contrasted with the well-known result from automata theory that every non-deterministic finite automaton can be expressed with a regular expression modulo language equivalence.

As an important tool in our proof, we have characterized the strongly connected components in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$, $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$, and $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$. An interesting open question is whether the given characterizations are complete, in the sense that a finite automaton is expressible in $\mathrm{BPA}^*_{\mathbf{0,1}}(\mathcal{A})$, $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$, or $\mathrm{ACP}^{*,\neg\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ iff all its strongly connected components satisfy the respective characterizations. If so, then our characterizations would constitute a useful complement to the characterization of Baeten *et al.* (2007)

and perhaps lead to a more efficient algorithm for deciding whether a non-deterministic automaton is expressible.

Bergstra *et al.* (2001) prove that every finite transition system without intermediate termination can be denoted in $\mathrm{ACP}^*_{\mathbf{0},\tau}(\mathcal{A},\gamma)$ up to *branching* bisimilarity (Glabbeek and Weijland 1996), and that $\mathrm{ACP}^*_{\mathbf{0}}(\mathcal{A},\gamma)$ modulo (strong) bisimilarity is strictly less expressive than $\mathrm{ACP}^*_{\mathbf{0},\tau}(\mathcal{A},\gamma)$. In contrast, we have established that every finite automaton (i.e., every finite transition system not excluding intermediate termination) is denoted by an $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expression. It follows that $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ and $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1},\tau}(\mathcal{A},\gamma)$ are equally expressive.

Acknowledgement

We thank Leonardo Vito and the other participants of the Formal Methods seminar of 2007 for their contributions in an early stage of the research for this paper.

Appendix A. Left Merge and Communication Merge do not Add Expressiveness to $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$

Our process theory $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ is grounded in Bergstra and Klop (1984) $\mathrm{ACP}(\mathcal{A},\gamma)$, but we have omitted the operations $⌊\!⌊$ and $|$ introduced by Bergstra and Klop to facilitate a finite axiomatization of the theory. We establish below that they do not add expressiveness in our setting of $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ with Kleene star instead of general recursion.

We shall in this appendix assume that the binary operations $⌊\!⌊$ and $|$ are included in the syntax of $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ and that the operational rules in Table 2 are added to those in Table 1 on page 937.

Table 2. *Operational rules for* $⌊\!⌊$ *and* $|$, *with* $a \in \mathcal{A}$.

$$18\ \frac{p \xrightarrow{a} p'}{p \mathbin{⌊\!⌊} q \xrightarrow{a} p' \parallel q} \qquad 19\ \frac{p \xrightarrow{a} p' \quad q \xrightarrow{b} q' \quad \gamma(a,b) \text{ is defined}}{p \mid q \xrightarrow{\gamma(a,b)} p' \parallel q'} \qquad 20\ \frac{p{\downarrow} \quad q{\downarrow}}{p \mid q{\downarrow}}$$

We shall prove that the process theories $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ with and without $⌊\!⌊$ and $|$ are equally expressive in a strong sense: modulo bisimilarity the operations $⌊\!⌊$ and $|$ can be eliminated from $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions. We proceed as follows: first we establish that $⌊\!⌊$ and $|$ can be eliminated from $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions of the form $p \mathbin{⌊\!⌊} q$ or $p \mid q$ provided that $⌊\!⌊$ and $|$ do not occur in p and q. Then we recall that bisimilarity is a congruence. Finally, we combine these facts to conclude that $⌊\!⌊$ and $|$ can be eliminated from all $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions.

According to the following lemma, occurrences of $⌊\!⌊$ can be eliminated from $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions.

Lemma A.1. Let p and q be $\mathrm{ACP}^*_{\mathbf{0},\mathbf{1}}(\mathcal{A},\gamma)$ expressions. If p and q do not contain occurrences of $⌊\!⌊$, then there exists a process expression r without occurrences of $⌊\!⌊$ such that $p \mathbin{⌊\!⌊} q \leftrightarrows r$.

Proof. If $\mathbin{\|\!\!\_}$ and $|$ are added to the syntax, then the transition system space $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ is still regular (cf. Lemma 2.1 on page 938). Hence, the set

$$Der(p) = \{(a,p') \mid p \xrightarrow{a} p'\}$$

is finite. We define r by

$$r = \sum_{(a,p')\in Der(p)} a\cdot(p' \parallel q).$$

Clearly, r has no occurrences of $\mathbin{\|\!\!\_}$, and it is straightforward to verify that the binary relation

$$\{(p \mathbin{\|\!\!\_} q, r), (r, p \mathbin{\|\!\!\_} q)\} \cup \{(p,p) \mid p \text{ a process expression}\}$$

is a bisimulation relation from $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ to $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, so $p \mathbin{\|\!\!\_} q \leftrightarrows r$. □

According to the following lemma, occurrences of $|$ can be eliminated from $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions too.

Lemma A.2. Let p and q be process expressions. If p and q do not contain occurrences of $|$, then there exists a process expression r without occurrences of $|$ such that $p \mid q \leftrightarrows r$.

Proof. If $\mathbin{\|\!\!\_}$ and $|$ are added to the syntax, then the transition system space $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ is still regular (cf. Lemma 2.1 on page 938). Hence, the sets

$$Der(p) = \{(a,p') \mid p \xrightarrow{a} p'\}$$

and

$$Der(q) = \{(b,q') \mid q \xrightarrow{b} q'\}$$

are finite. Define the set $Der(r)$ by

$$Der(r) = \{(\gamma(a,b), p' \parallel q') \mid (a,p') \in Der(p), (b,q') \in Der(q),\ \text{and } \gamma(a,b) \text{ is defined}\},$$

and define r by

$$r = \sum_{(c,r')\in Der(r)} c\cdot r'.$$

Clearly, r has no occurrences of $|$, and it is straightforward to verify that the binary relation

$$\{(p \mid q, r), (r, p \mid q)\} \cup \{(p,p) \mid p \text{ a process expression}\}$$

is a bisimulation relation from $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ to $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$, so $p \mid q \leftrightarrows r$. □

To be able to apply the previous two lemmas in the context of a larger $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression, we need to know that bisimilarity is a congruence on $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions.

Lemma A.3. The relation $\leftrightarrows$ is a congruence on the set of $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions.

Proof. It is well known that $\leftrightarroweq$ is an equivalence. Moreover, the operational rules in Tables 1 and 2 are all in the *path format*, so by Theorem 5.4 of Baeten and Verhoef (1993) bisimilarity is compatible with the syntax of $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$. □

Now we are in a position to establish the elimination $\mathbin{||\!\_}$ and $|$ from all $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expressions.

Proposition A.4. For every $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression p there exists an $\mathrm{ACP}^*_{\mathbf{0,1}}(\mathcal{A},\gamma)$ expression q such that $p \leftrightarroweq q$.

Proof. By a straightforward induction on the structure of p, using Lemma A.3 for applying the induction hypothesis, Lemma A.1 for the case that p is of the form $p_1 \mathbin{||\!\_} p_2$, and Lemma A.2 for the case that p is of the form $p_1 \mid p_2$. □

Similarly, it can be shown that adding $\mathbin{||\!\_}$ to $\mathrm{PA}^*_{\mathbf{0,1}}(\mathcal{A})$ or $\mathbin{||\!\_}$ and $|$ to $\mathrm{ACP}^{*,-\partial}_{\mathbf{0,1}}(\mathcal{A},\gamma)$ does not add expressiveness to the respective process theories.

References

Baeten, J. C. M., Basten, T. and Reniers, M. A. (2010) *Process Algebra: Equational Theories of Communicating Processes*, Cambridge Tracts in Theoretical Computer Science volume 50, Cambridge University Press.

Baeten, J. C. M., Corradini, F. and Grabmayer, C. A. (2007) A characterization of regular expressions under bisimulation. *Journal of the ACM* **54** (2) 6.1–28.

Baeten, J. C. M. and van Glabbeek, R. J. (1987) Merge and termination in process algebra. In: Nori, K. V. (ed.) Foundations of Software Technology and Theoretical Computer Science. *Springer Lecture Notes in Computer Science* **287** 153–172.

Baeten, J. C. M. and Verhoef, C. (1993) A congruence theorem for structured operational semantics with predicates. In: Best, E. (ed.) Conference on Concurrency Theory. *Springer Lecture Notes in Computer Science* **715** 477–492.

Bergstra, J. A., Bethke, I. and Ponse, A. (1994) Process algebra with iteration and nesting. *The Computer Journal* **37** (4) 243–258.

Bergstra, J. A., Fokkink, W. and Ponse, A. (2001) Process algebra with recursive operations. In: Bergstra, J. A., Ponse, A. J. and Smolka, S. A. (eds.) *Handbook of Process Algebra*, Elsevier 333–389.

Bergstra, J. A. and Klop, J. W. (1984) Process algebra for synchronous communication. *Information and Control* **60** (1–3) 109–137.

Fröschle, S. B. and Valencia, F. D. (eds.) (2010) Proceedings 17th International Workshop on Expressiveness in Concurrency. *Electronic Proceedings in Theoretical Computer Science* **41** 1–15.

Glabbeek, R. J. V. and Weijland, W. P. (1996) Branching time and abstraction in bisimulation semantics. *Journal of the ACM* **43** (3) 555–600.

Gorla, D. (2010) Towards a unified approach to encodability and separation results for process calculi. *Information and Computation* **208** (9) 1031–1053.

Hopcroft, J. E., Motwani, R. and Ullman, J. D. (2006) *Introduction to Automata Theory, Languages, and Computation*, Pearson.

Koymans, C. J. P. and Vrancken, J. L. M. (1985) Extending process algebra with the empty process ε, Logic Group Preprint Series 1, State University of Utrecht.

Milner, R. (1984) A complete inference system for a class of regular behaviours. *Journal of Computer System Science* **28** (3), 439–466.

Milner, R. (1989) *Communication and Concurrency*, Prentice-Hall International, Englewood Cliffs.

Park, D. (1981) Concurrency and automata on infinite sequences. In: Deussen, P. (ed.) Proceedings of the 5th GI Conference. *Springer-Verlag Lecture Notes in Computer Science* **104**, Karlsruhe, Germany 167–183.

Plotkin, G. D. (2004) A structural approach to operational semantics. *Journal of Logic and Algebraic Programming* **60–61** 17–139.

Vrancken, J. L. M. (1997) The algebra of communicating processes with empty process. *Theoretical Computer Science* **177** (2) 287–328.