

Infinite Product of Traces Represented by Projections*

Roman R. Redziejowski

Abstract

The construction of an associative ω -product of traces from an earlier paper is revisited using projection representation of traces. Using projections instead of trace prefixes results in very simple definitions and proofs.

1 Introduction

One way of describing behavior of concurrent systems is by means of "traces" introduced by Mazurkiewicz [10–12]. The analysis of systems that do not stop is naturally reduced to considering their infinite behavior. This leads to the notion of an infinite trace. An intimately related notion is that of an infinite concatenation, or " ω -product", of finite traces, which describes behavior of an infinite sequence of processes.

Neither of these two notions has an immediately obvious definition. Finite trace is a set of finite words that can be transformed into each other by commuting certain "independent" letters. An infinite trace involves an infinite number of such commutations. When two finite traces are concatenated, independent letters can "migrate" between the two components as the result of commutations. In an infinite concatenation we may have infinite such migrations.

The two notions, however, have some intuitively obvious properties that must be satisfied by their definition. First, an infinite trace should in some way be the limit of its initial portions. Second, the infinite product should similarly be the limit of partial products. Third, the result of an ω -product of finite traces should be an infinite trace. Finally, the ω -product should be associative, that means, the result should not change if some factors are concatenated before applying the infinite operation.

The idea of infinite trace being the limit of its prefixes has been exploited in the definitions of infinite trace given by Mazurkiewicz [11], Kwiatkowska [7, 8], and Gastin [4, 5], the last two being essentially identical. In [4], Gastin has also defined the ω -product of traces as the limit of partial products.

In [15], the present author attempted to construct an ω -product of traces starting from the requirement of associativity. It was a rather complicated construction in terms of trace prefixes, using ideas from [14]. The associativity itself does not determine the ω -product. An additional requirement that the product be "maximal" fixes it up to isomorphism. The paper suggested four different (but isomorphic) versions of such maximal product. Defining infinite trace as the result of the ω -product gave four different definitions of infinite trace, two of them identical to those from [11] and [4, 5, 7, 8], respectively.

The present paper is inspired by an observation that both the infinite trace and infinite concatenation of traces have simple definitions in terms of projections. An infinite trace is an infinite set of infinite words, while its projection is only a finite set of infinite words, which is much easier to handle. In particular, the concatenation of traces corresponds to concatenation of projections, and thereby to an ordinary concatenation of words.

The representation of traces by projections was used by, for example, Cori and Perrin [1], Diekert [2], Gastin [4, 5], Kwiatkowska [7–9], and Shields [16, 17]. It is very naturally extended to yield the definition of an infinite trace. In fact, infinite trace was first defined in this way by Flé and Roucairol [3]. The definition was recently used by Mikulski [13].

We begin, in Section 2, by recalling the basic definitions and facts concerning traces. We proceed then, in Section 3, to projections, and to projection representation of traces in Section 4. In Section 5, we recall how infinite trace is defined in terms of projections. In Section 6, we define the ω -product of traces, and

*Appeared in *Fundamenta Informaticae* 101, 1-2 (2010) 105–113.

in subsequent two sections check that it is associative and maximal. The Appendix contains proofs of two, rather technical, facts about projections.

2 Words and traces

An *alphabet* A is a finite nonempty set of *letters*. A *word* is a finite or infinite string of letters. The number of letters in a finite word x is called its *length* and is denoted by $|x|$. The word of length 0 (the "empty word") is denoted by ε . The set of all finite words is denoted by A^* , and the set of all words by A^∞ .

Concatenation of words $x \in A^*$ and $y \in A^\infty$ is the word obtained by appending y at the end of x . It is denoted by xy . The word obtained by joining an infinite sequence of finite words x_1, x_2, x_3, \dots one after another is denoted by $x_1x_2x_3\dots$. Word $y \in A^*$ is a *prefix* of word $x \in A^\infty$, denoted $y \leq x$, if $x = yz$ for some $z \in A^\infty$.

Traces are equivalence classes of certain congruence relation \equiv on the semigroup (A^*, \cdot) , where \cdot is concatenation of words. This congruence is defined by a symmetric and irreflexive *independence relation* $I \subseteq A \times A$. It is the smallest congruence \equiv such that $(a, b) \in I \Rightarrow ab \equiv ba$. A trace consists thus of words that can be transformed into each other by commuting independent letters.

The trace containing word $x \in A^*$ is denoted by $[x]$. The set of all traces is denoted by \mathbb{T} .

Relation \equiv being a congruence means that there exists quotient operation \cdot , the *trace product*, defined by $[x] \cdot [y] = [xy]$ for $[x], [y] \in \mathbb{T}$. The trace product is associative and cancellative.

3 Projections

A *projection* of word $x \in A^\infty$ on a sub-alphabet $\alpha \subseteq A$, denoted by $p_\alpha(x)$, is the word obtained by deleting from x all letters not in α . One can easily see that

$$x \leq y \Rightarrow p_\alpha(x) \leq p_\alpha(y), \quad (1)$$

$$p_\alpha(x)p_\alpha(y) = p_\alpha(xy), \quad (2)$$

$$p_\alpha(x_1)p_\alpha(x_2)p_\alpha(x_3)\dots = p_\alpha(x_1x_2x_3\dots), \quad (3)$$

$$p_\alpha(y) = p_\alpha(p_\beta(y)), \quad (4)$$

for any $\alpha \subseteq \beta \subseteq A$, $x \in A^*$, $y \in A^\infty$, and $x_i \in A^*$.

For a family $\mathbf{C} \subseteq 2^A$ of sub-alphabets, the projection of $x \in A^\infty$ on \mathbf{C} , denoted by $P_{\mathbf{C}}(x)$, is the set of projections of x on all members of \mathbf{C} . More precisely, it is a mapping that assigns $p_\alpha(x)$ to each $\alpha \in \mathbf{C}$. We view it as a vector of words indexed by \mathbf{C} . We write $P_{\mathbf{C}}(x) \leq P_{\mathbf{C}}(y)$ and $P_{\mathbf{C}}(x) = P_{\mathbf{C}}(y)$ to mean that, respectively, $p_\alpha(x) \leq p_\alpha(y)$ and $p_\alpha(x) = p_\alpha(y)$ for each $\alpha \in \mathbf{C}$.

In the following, we consider projections on a family \mathbf{C} of "cliques".

Define D to be the complement $(A \times A) - I$ of the independence relation. A *clique* is a subset $\alpha \subseteq A$ such that $\alpha \times \alpha \subseteq D$, that is, $(a, b) \notin I$ for all $a, b \in \alpha$.

A *clique covering* is a family of cliques such that $\bigcup_{\alpha \in \mathbf{C}} (\alpha \times \alpha) = D$, that is:

$$(a, b) \notin I \Leftrightarrow \exists \alpha \in \mathbf{C} (\{a, b\} \subseteq \alpha). \quad (5)$$

Because I is irreflexive, each $\{a\}$ for $a \in A$ is included in some $\alpha \in \mathbf{C}$, so \mathbf{C} is a covering of the alphabet A . Notice that, in particular, $\{a\}$ is a clique for each letter a .

In general, there exist several different clique coverings for a given alphabet A and independence relation I . For our purpose, they are all equivalent in the following sense:

Proposition 1. *Let \mathbf{C} be any clique covering, and $\beta \subseteq A$ any clique, not necessarily a member of \mathbf{C} . For any $x, y \in A^\infty$ holds $P_{\mathbf{C}}(x) = P_{\mathbf{C}}(y) \Rightarrow p_\beta(x) = p_\beta(y)$.*

(Proof is found in the Appendix.)

From now on, we consider an arbitrary clique covering \mathbf{C} that will remain fixed for the rest of the discussion. We omit the index \mathbf{C} and write $P(x)$ to mean $P_{\mathbf{C}}(x)$.

4 Projection representation of traces

A clique consists of letters that are all mutually dependent. As such, they do not commute, and must appear in the same sequence in all words belonging to the same trace. This sequence, obtained as projection on the clique, is thus an invariant of the trace. One may expect that the trace is fully defined by a set of such invariants. Indeed, the following fundamental result was published at about the same time by Shields [17] (Theorem 3.2.8) and Cori-Perrin [1] (Proposition 1.1):

Proposition 2. *For any $x, y \in A^*$, $P(x) = P(y) \Leftrightarrow x \equiv y$.*

It means that trace $[x]$ consists of all words having the same projection vector as x . A trace is thus identified by a single projection vector. This is the projection representation of trace.

With traces representing concurrency, cliques represent sequential processes, and projection on a clique is the sequence of operations (the "history") of such process. In [17], Shields starts with subalphabets representing processes and uses formula (5) to derive independence relation that turns them into cliques (Lemma 3.2.2). Kwiatkowska [8] uses a similar construction for what she calls "alphabet structure", and remarks that different alphabet structures represent different decompositions into sequential processes.

5 Infinite trace

Proposition 2 can be taken as an alternative definition of the equivalence \equiv . It extends naturally to infinite words by defining:

$$x \equiv^\infty y \Leftrightarrow P(x) = P(y) \text{ for } x, y \in A^\infty. \quad (6)$$

Proposition 1 ensures that this definition does not depend on the clique covering used.

The notion of trace can be now extended to mean an equivalence class of \equiv^∞ in A^∞ . We keep the notation $[x]$ for the trace containing $x \in A^\infty$:

$$[x] = \{y \in A^\infty \mid P(y) = P(x)\} \text{ for } x \in A^\infty. \quad (7)$$

One can easily see that for an infinite word x , $[x]$ is a set of infinite words. This is the infinite trace. The set of all traces, finite and infinite, is in the following denoted by \mathbb{T}_∞ .

The trace product is naturally extended to infinite traces by defining

$$[x] \cdot [y] = [xy] \text{ for } [x] \in \mathbb{T} \text{ and } [y] \in \mathbb{T}_\infty. \quad (8)$$

We note that this definition does not depend on the choice of words x and y used to represent the traces $[x]$ and $[y]$. According to (7), $[xy]$ consists of words having projection vector identical to $P(xy)$. According to (2), each component $p_\alpha(xy)$ of that vector is equal to $p_\alpha(x)p_\alpha(y)$, which, according to (7), does not change if x is replaced by $x' \in [x]$ and y by $y' \in [y]$.

The definition (7) of infinite trace has been given by a number of authors [3–9, 13]. They handle its uniqueness in two different ways. Some [3, 13] use a unique clique covering that consists of singletons and pairs. Other define the infinite trace first in terms of prefixes, and then prove it is identical to one defined from projections. Gastin and Petit [6] remark that otherwise one would have to prove independence of (6) from the clique covering used. This is provided here in the form of Proposition 1.

6 Infinite product of traces

An infinite concatenation of finite traces $[x_1], [x_2], [x_3], \dots$ can be seen as an infinite application of trace product: $[x_1] \cdot [x_2] \cdot [x_3] \cdot \dots$. The problem is, we do not know how to apply the binary operator \cdot infinitely many times. As mentioned in the Introduction, one possible approach is to take the limit of partial products $[x_1] \cdot \dots \cdot [x_n]$ for $n \rightarrow \infty$.

We take here a different approach and view infinite concatenation as application of a new operator π that maps infinite sequences of traces into traces. According to this view, $[x_1] \cdot [x_2] \cdot [x_3] \cdot \dots$ is an informal way to write $\pi(\mathbf{x})$ where \mathbf{x} is the sequence $[x_1], [x_2], [x_3], \dots$. We refer to the operator π as the *infinite product*, or the ω -*product* of traces. It remains to define the mapping π .

By analogy to $[x] \cdot [y] = [xy]$, it is natural to have $[x_1] \cdot [x_2] \cdot [x_3] \cdot \dots = [x_1x_2x_3 \dots]$. We define thus:

$$\pi([x_1], [x_2], [x_3], \dots) = [x_1x_2x_3 \dots] \quad \text{for } [x_1], [x_2], [x_3], \dots \in \mathbb{T}^{\mathbb{N}}, \quad (9)$$

where $\mathbb{T}^{\mathbb{N}}$ is the set of all infinite sequences of finite traces. We note that this definition does not depend on the words x_i used to represent the traces $[x_i]$. Written more explicitly, (9) is:

$$\pi([x_1], [x_2], [x_3], \dots) = \{x \in A^\infty \mid P(x) = P(x_1x_2x_3 \dots)\}. \quad (10)$$

According to (3), each component of the vector $P(x_1x_2x_3 \dots)$ is equal to $p_\alpha(x_1)p_\alpha(x_2)p_\alpha(x_3) \dots$, which, according to (7), does not change if x_i is replaced by $x'_i \in [x_i]$ for $i > 1$.

If infinitely many among x_i are not empty, $x_1x_2x_3 \dots$ is an infinite word, and $\pi([x_1], [x_2], [x_3], \dots)$ is an infinite trace. Otherwise $x_1x_2x_3 \dots$ is a finite word and $\pi([x_1], [x_2], [x_3], \dots)$ is a finite trace. We have thus $\pi : \mathbb{T}^{\mathbb{N}} \rightarrow \mathbb{T}_\infty$.

7 The infinite product π is associative

Associativity of an operator has the effect that the result is not affected by different groupings of operations. We exploit this property to define the associativity of π . To speak about associativity of an infinitary operator, we must consider infinite grouping. We are thus going to say that π is *associative* if it has these properties:

$$[x_1] \cdot [x_2] \cdot [x_3] \cdot \dots = [x_1 \dots x_{n_1}] \cdot [x_{n_1+1} \dots x_{n_2}] \cdot [x_{n_2+1} \dots x_{n_3}] \cdot \dots, \quad (11)$$

$$[x_1] \cdot [x_2] \cdot [x_3] \cdot \dots = [x_1 \dots x_n] \cdot [x_{n+1}x_{n+2}x_{n+3} \dots], \quad (12)$$

for any $[x_1], [x_2], [x_3], \dots \in \mathbb{T}^{\mathbb{N}}$, $n \geq 1$, and ascending sequence n_1, n_2, n_3, \dots of natural numbers.

By applying (9), we find that the right-hand side of (11) is equal to:

$$[(x_1 \dots x_{n_1})(x_{n_1+1} \dots x_{n_2})(x_{n_2+1} \dots x_{n_3}) \dots] = [x_1x_2x_3 \dots],$$

which is identical to the result of $[x_1] \cdot [x_2] \cdot [x_3] \cdot \dots$. This shows that the ω -product π satisfies (11). According to (8), the right-hand side of (12) is equal to:

$$[(x_1 \dots x_{n_1})(x_{n_1+1}x_{n_1+2}x_{n_1+3} \dots)] = [x_1x_2x_3 \dots],$$

which is identical to the result of $[x_1] \cdot [x_2] \cdot [x_3] \cdot \dots$. The ω -product π satisfies thus also (12) and is associative according to our definition.

8 The infinite product π is maximal

We shall say that an associative ω -product is *maximal* if two sequences have the same product only when this is required by associativity.

We used above the informal representation of π to define associativity. For a more formal statement, let us say that sequence $\mathbf{y} = [y_1], [y_2], [y_3], \dots \in \mathbb{T}^{\mathbb{N}}$ is a *contraction* of sequence $\mathbf{x} = [x_1], [x_2], [x_3], \dots \in \mathbb{T}^{\mathbb{N}}$, denoted by $\mathbf{x} \triangleright \mathbf{y}$, if there exists an ascending sequence of natural numbers n_1, n_2, n_3, \dots such that

$$[y_1] = [x_1 \dots x_{n_1}] \quad \text{and} \quad [y_i] = [x_{n_{i-1}+1} \dots x_{n_i}] \quad \text{for } i > 1.$$

The property (11) can be now stated as

$$\mathbf{x} \triangleright \mathbf{y} \Rightarrow \pi(\mathbf{x}) = \pi(\mathbf{y}) \quad \text{for } \mathbf{x}, \mathbf{y} \in \mathbb{T}^{\mathbb{N}}.$$

Define $\mathbf{x} \sim \mathbf{y}$ to mean that sequences $\mathbf{x}, \mathbf{y} \in \mathbb{T}^{\mathbb{N}}$ can be transformed into each other in a finite number of steps using contraction or its opposite. More precisely, there exist sequences $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k$ such that $\mathbf{x} = \mathbf{z}_1$, $\mathbf{y} = \mathbf{z}_k$, and either $\mathbf{z}_i \triangleright \mathbf{z}_{i+1}$ or $\mathbf{z}_{i+1} \triangleright \mathbf{z}_i$ for $1 \leq i < k$.

From associativity follows $\mathbf{x} \sim \mathbf{y} \Rightarrow \pi(\mathbf{x}) = \pi(\mathbf{y})$. The ω -product π is maximal if it also satisfies the opposite implication, that is,

$$\mathbf{x} \sim \mathbf{y} \Leftrightarrow \pi(\mathbf{x}) = \pi(\mathbf{y}) \quad \text{for } \mathbf{x}, \mathbf{y} \in \mathbb{T}^{\mathbb{N}}. \quad (13)$$

To demonstrate that π is maximal, we need an auxiliary result about projections:

Lemma 1. *Let $u, v \in A^\infty$ be such that $P(u) = P(v)$. For any $x, y \in A^*$ such that $x \leq u$, $y \leq v$, $|x| < |y|$, there exists $z \in A^*$ such that $[x] \cdot [z] = [y]$.*

(Proof is found in the Appendix.)

Proposition 3. *The ω -product π defined by (9) is maximal.*

Proof. Let $\mathbf{x} = [x_1], [x_2], [x_3], \dots \in \mathbb{T}^{\mathbb{N}}$ and $\mathbf{y} = [y_1], [y_2], [y_3], \dots \in \mathbb{T}^{\mathbb{N}}$ be such that $\pi(\mathbf{x}) = \pi(\mathbf{y})$.

If $\pi(\mathbf{x}) = \pi(\mathbf{y})$ is a finite trace, only finitely many among x_i , respectively y_i , are different from ε , and we have $\mathbf{x} \triangleright [u], [\varepsilon], [\varepsilon], [\varepsilon], \dots$ and $\mathbf{y} \triangleright [w], [\varepsilon], [\varepsilon], [\varepsilon], \dots$. From (9) we have $[u] = [w]$, showing that $\mathbf{x} \sim \mathbf{y}$.

Suppose now that $\pi(\mathbf{x}) = \pi(\mathbf{y})$ is an infinite trace, meaning that both $x_1x_2x_3\dots$ and $y_1y_2y_3\dots$ are infinite words. According to (10) we have $P(x_1x_2x_3\dots) = P(y_1y_2y_3\dots)$.

Take any $n_1 \geq 1$ and denote $x_1 \dots x_{n_1} = z_1$.

Take any m_1 such that $|y_1 \dots y_{m_1}| > |x_1 \dots x_{n_1}|$. This is always possible because $y_1y_2y_3\dots$ is infinite. We have $x_1 \dots x_{n_1} \leq x_1x_2x_3\dots$ and $y_1 \dots y_{m_1} \leq y_1y_2y_3\dots$. By Lemma 1, there exists $z_2 \in A^*$ such that $[x_1 \dots x_{n_1}] \cdot [z_2] = [y_1 \dots y_{m_1}]$.

Take now any n_2 such that $|x_1 \dots x_{n_2}| > |y_1 \dots y_{m_1}|$. By the same argument as before, there exists z_3 such that $[y_1 \dots y_{m_1}] \cdot [z_3] = [x_1 \dots x_{n_2}]$.

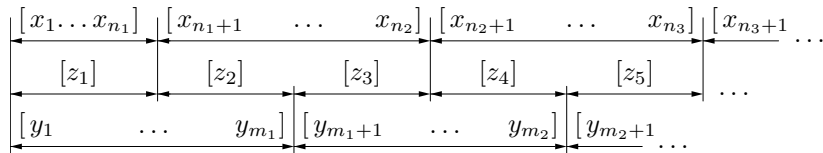
This can be repeated indefinitely to produce sequences n_1, n_2, n_3, \dots , m_1, m_2, m_3, \dots , and z_1, z_2, z_3, \dots such that

$$\begin{aligned} [x_1 \dots x_{n_i}] \cdot [z_{2i}] &= [y_1 \dots y_{m_i}], \\ [y_1 \dots y_{m_i}] \cdot [z_{2i+1}] &= [x_1 \dots x_{n_{i+1}}], \end{aligned}$$

for all $i \geq 1$. By left-cancellation, we have, for all $i > 1$:

$$\begin{aligned} [x_{n_{i-1}+1} \dots x_{n_i}] &= [z_{2i-2} z_{2i-1}], \\ [y_{m_{i-1}+1} \dots y_{m_i}] &= [z_{2i-1} z_{2i}]. \end{aligned}$$

This can be illustrated as follows:



Denote $\mathbf{z} = z_1, z_2, z_3, \dots$, $\mathbf{p} = [z_1], [z_2z_3], [z_4z_5], \dots$, and $\mathbf{q} = [z_1z_2], [z_3z_4], [z_5z_6], \dots$. One can easily see that $\mathbf{x} \triangleright \mathbf{p} \triangleleft \mathbf{z} \triangleright \mathbf{q} \triangleleft \mathbf{y}$, showing that $\mathbf{x} \sim \mathbf{y}$. \square

9 The infinite product π is unique up to isomorphism

As stated in the introduction, a maximal ω -product is unique up to isomorphism. To explain what we mean by this, consider an ω -product as a pair (V, π) consisting of a set V of *values* and a surjective mapping $\pi : \mathbb{T}^{\mathbb{N}} \rightarrow V$. We shall say that ω -products (V_1, π_1) and (V_2, π_2) are *isomorphic* to mean that there exists a bijection $\varphi : V_1 \rightarrow V_2$ such that $\varphi(\pi_1(\mathbf{x})) = \pi_2(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{T}^{\mathbb{N}}$.

One can easily verify that any two maximal ω -products are isomorphic. Suppose (V_1, π_1) and (V_2, π_2) are maximal. For $v \in V_1$ define $\varphi(v) = \pi_2(\mathbf{x})$, where $\mathbf{x} \in \mathbb{T}^{\mathbb{N}}$ is any sequence such that $\pi_1(\mathbf{x}) = v$. Because π_1 is surjective, such \mathbf{x} always exists. The result does not depend on the choice of \mathbf{x} : if $\pi_1(\mathbf{x}) = \pi_1(\mathbf{y})$, we have, by (13), $\mathbf{x} \sim \mathbf{y}$ and $\pi_2(\mathbf{x}) = \pi_2(\mathbf{y})$. The function φ is thus uniquely defined and satisfies $\varphi(\pi_1(\mathbf{x})) = \pi_2(\mathbf{x})$. Suppose $\varphi(v) = \varphi(v')$. That means $\pi_2(\mathbf{x}) = \pi_2(\mathbf{y})$ for some \mathbf{x}, \mathbf{y} such that $v = \pi_1(\mathbf{x})$, $v' = \pi_1(\mathbf{y})$. By (13), we have $\mathbf{x} \sim \mathbf{y}$ and $v = v'$, showing that φ is a bijection.

The ω -product π being maximal means that it is isomorphic to all ω -products defined in [15]. It is, in fact, identical to one of them.

10 Conclusions

The definition of an ω -product of traces in terms of projections is simple and intuitive. The associativity follows trivially from associativity of the infinite concatenation of words.

The constructions in [15] using trace prefixes, and the definition as upper bound of partial products in [4] are very complex in comparison.

Appendix

Proof of Proposition 1

Let \mathbf{C} be any clique covering, and $\beta \subseteq A$ any clique, not necessarily a member of \mathbf{C} . Let $x, y \in A^\infty$ be such that $P_{\mathbf{C}}(x) = P_{\mathbf{C}}(y)$, meaning that $p_\alpha(x) = p_\alpha(y)$ for all $\alpha \in \mathbf{C}$.

We show first that x and y have identical projections on any $\{a\} \subseteq \beta$ and $\{a, b\} \subseteq \beta$.

As found before, each $\{a\} \subseteq \alpha$ for some $\alpha \in \mathbf{C}$. By β being a clique, we have $(a, b) \notin I$ for each $\{a, b\} \subseteq \beta$, and, by (5), $\{a, b\} \subseteq \alpha$ for some $\alpha \in \mathbf{C}$. Using the respective α and (4), we find

$$p_{\{a\}}(x) = p_{\{a\}}(p_\alpha(x)) = p_{\{a\}}(p_\alpha(y)) = p_{\{a\}}(y), \quad (14)$$

$$p_{\{a,b\}}(x) = p_{\{a,b\}}(p_\alpha(x)) = p_{\{a,b\}}(p_\alpha(y)) = p_{\{a,b\}}(y). \quad (15)$$

Suppose now that $p_\beta(x)$ and $p_\beta(y)$ have identical prefix z of length n . This is true for $n = 0$. Consider any $n \geq 0$. There are four possibilities:

- (a) $p_\beta(x) = z = p_\beta(y)$.
- (b) $p_\beta(x) = z$ and $p_\beta(y) = z\mathbf{a}u$ for some $\mathbf{a} \in A$ and $u \in A^\infty$. (Exchange x and y if the inverse holds.)
Clearly, $\{\mathbf{a}\} \subseteq \beta$. From (2) and (4) we have $p_{\{\mathbf{a}\}}(y) = p_{\{\mathbf{a}\}}(z)\mathbf{a}p_{\{\mathbf{a}\}}(u) \neq p_{\{\mathbf{a}\}}(z) = p_{\{\mathbf{a}\}}(x)$, which contradicts (14).
- (c) $p_\beta(x) = z\mathbf{a}u$ and $p_\beta(y) = z\mathbf{b}v$ for some $\mathbf{a}, \mathbf{b} \in A$, $\mathbf{a} \neq \mathbf{b}$ and $u, v \in A^\infty$.
Clearly, $\{\mathbf{a}, \mathbf{b}\} \subseteq \beta$. From (2) and (4) we have $p_{\{\mathbf{a}, \mathbf{b}\}}(x) = p_{\{\mathbf{a}, \mathbf{b}\}}(z)\mathbf{a}p_{\{\mathbf{a}, \mathbf{b}\}}(u) \neq p_{\{\mathbf{a}, \mathbf{b}\}}(z)\mathbf{b}p_{\{\mathbf{a}, \mathbf{b}\}}(v) = p_{\{\mathbf{a}, \mathbf{b}\}}(y)$, which contradicts (15).
- (d) $p_\beta(x) = z\mathbf{a}u$ and $p_\beta(y) = z\mathbf{a}v$ for some $\mathbf{a} \in A$ and $u, v \in A^\infty$. That means $p_\beta(x)$ and $p_\beta(y)$ have identical prefix of length $n + 1$.

Thus, only (a) and (d) are possible, and $p_\beta(x) = p_\beta(y)$ follows by induction on n .

Proof of Lemma 1

Let $u, v \in A^\infty$ be such that $P(u) = P(v)$. Let $x, y \in A^*$ be such that $x \leq u$, $y \leq v$, and $|x| < |y|$.

Consider any $\alpha \in \mathbf{C}$. We have $p_\alpha(u) = p_\alpha(v) = w$. Applying (1) to $x \leq u$, $y \leq v$, we obtain $p_\alpha(x) \leq w$, $p_\alpha(y) \leq w$. That means either $p_\alpha(x) \leq p_\alpha(y) \leq w$ or $p_\alpha(y) \leq p_\alpha(x) \leq w$. This second possibility is excluded by $|x| < |y|$, so we have $p_\alpha(x) \leq p_\alpha(y) \leq w$. As this holds for all $\alpha \in \mathbf{C}$, we have $P(x) \leq P(y)$.

The following fact appears in slightly different forms as Lemma 3.2.9 in [17], Proposition 3.4.12 in [8], and Proposition 1.3 in [4]:

”For any $x, y \in A^*$ such that $P(x) \leq P(y)$ exists $z \in A^*$ such that $P(xz) = P(y)$ ”.

As $P(xz) = P(y)$ means $[x] \cdot [z] = [y]$, this ensures that the required z exists.

References

- [1] CORI, R., AND PERRIN, D. Automates et commutations partielles. *RAIRO Informatique Théorique et Applications* 19, 1 (1985), 21–32.
- [2] DIEKERT, V. *Combinatorics on Traces*. No. 454 in Lecture Notes in Comp. Sci. Springer-Verlag, 1987.
- [3] FLÉ, M., AND ROUCAIROL, G. Maximal serializability of iterated transactions. *Theoretical Comp. Sci.* 38 (1985), 1–16.
- [4] GASTIN, P. Infinite traces. In *Semantics of Systems of Concurrent Processes*, I. Guessarian, Ed., no. 469 in Lecture Notes in Comp. Sci. Springer-Verlag, 1990, pp. 277–308.
- [5] GASTIN, P. Un modèle asynchrone pour les systèmes distribués. *Theoretical Comp. Sci.* 74 (1990), 121–162.
- [6] GASTIN, P., AND PETIT, A. Infinite traces. In *The Book of Traces*, V. Diekert and G. Rozenberg, Eds. World Scientific, 1995, pp. 393–486.
- [7] KWIATKOWSKA, M. Z. Event fairness and non-interleaving concurrency. *Formal Aspects of Computing* 1, 3 (1989), 213–228.
- [8] KWIATKOWSKA, M. Z. Fairness for non-interleaving concurrency (PhD Thesis). Tech. Rep. 22, University of Leicester, Department of Computing Studies, May 1989.
- [9] KWIATKOWSKA, M. Z. Defining process fairness for non-interleaving concurrency. In *FSTTCS* (1990), K. V. Nori and C. E. V. Madhavan, Eds., vol. 472 of *Lecture Notes in Computer Science*, Springer, pp. 286–300.
- [10] MAZURKIEWICZ, A. Concurrent program schemes and their interpretations. Tech. Rep. DAIMI PB-78, Comp. Sci. Dep., Århus Universitet, 1977.
- [11] MAZURKIEWICZ, A. Trace theory. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, W. Brauer, W. Reisig, and G. Rozenberg, Eds., no. 255 in Lecture Notes in Comp. Sci. Springer-Verlag, 1987, pp. 371–388.
- [12] MAZURKIEWICZ, A. Introduction to trace theory. In *The Book of Traces*, V. Diekert and G. Rozenberg, Eds. World Scientific, 1995, pp. 3–41.
- [13] MIKULSKI, L. Projection representation of Mazurkiewicz traces. *Fundamenta Informaticae* 85, 1-4 (2008), 399–408.
- [14] REDZIEJOWSKI, R. R. On associative omega-products. *Fundamenta Informaticae* 60, 1-4 (2004), 333–350.
- [15] REDZIEJOWSKI, R. R. Associative omega-products of traces. *Fundamenta Informaticae* 67, 1–3 (Aug. 2005), 175–185.
- [16] SHIELDS, M. W. Adequate path expressions. In *Semantics of Concurrent Computation* (1979), G. Kahn, Ed., vol. 70 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 249–265.
- [17] SHIELDS, M. W. Concurrent machines. *Comput. J.* 28, 5 (1985), 449–465.