On the concatenation of infinite traces*

Volker Diekert

Institut für Informatik, Technische Universität München, Arcisstr. 21, D-8000 München 2, Germany

Abstract

Diekert, V., On the concatenation of infinite traces, Theoretical Computer Science 113 (1993) 35–54. There is a straightforward generalization of traces to infinite traces as dependence graphs where every vertex has finitely many predecessors, or what is the same, as a backward closed and directed set of traces with respect to prefix ordering. However, this direct approach has a drawback since it does not allow one to describe some basic phenomena which are related to concatenation. We solve this problem by adding to an infinite trace a second component. This second component is a finite alphabetic information which is called the alphabet at infinity.

We obtain a compact and complete ultra-metric space where the concatenation is uniformly continuous and where the set of finite traces is an open, discrete, and dense subset. Our objects arise in a natural way from the consideration of dependence graphs where the induced partial order is well-founded. Such a graph splits into a so-called real part and a transfinite part. From the transfinite part only its alphabet is of importance.

Our approach is a nontrivial generalization of the well-known construction for words and yields a convenient semantics for infinite concurrent processes.

0. Introduction

Trace theory has been recognized as an important tool for investigations of concurrent systems. This dates back to the work of Mazurkiewicz [15], who used traces as a suitable partial-order semantics for elementary systems. Since then a systematic study of traces under various aspects has begun; see [1, 6, 16, 17] for overviews. A theory of infinite traces started only recently. But the interest in this theory has grown quickly. Some recent works are [2, 7, 8, 10, 12, 14]. The general background is to study concurrent processes which never terminate. In fact, ideas to use traces for infinite processes can already be found in [16]. Another early example is [9], where infinite traces are used to solve serializability questions of iterated transactions.

Correspondence to: V. Diekert, Institut für Informatik der Universität Stuttgart, Breitwiesenstr. 20–22, D-7000 Stuttgart 80, Germany. Email: diekert@informatik.uni-stuttgart.de.

*Supported by the ESPRIT Basic Research Actions No. 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS) and No. 3148: Design Methods Based on Nets (DEMON)
The abstract model is an infinite labelled acyclic graph where arcs are between dependent actions and every vertex has only finitely many predecessors. The semantics is that an execution has to respect the induced partial order, only. In particular, if two actions are not ordered then they can be performed concurrently. Equivalently, we could also consider prefix-closed, directed subsets of finite traces. From a certain viewpoint this is a convenient notion. But it is unsatisfactory with respect to some specific phenomena on infinite concurrent processes. For example, let $P$ and $Q$ be the following procedures:

$$
P = \text{begin } \ y := 0; \ x := y; \text{ while true do } x := x + 1 \text{ od; } \ x := y; \ y := y + 1; \text{ end}
$$

$$
Q = \text{begin } \ y := 0; \ x := y; \text{ while true do } x := x + 1 \text{ od; } \ y := y + 1; \text{ end}
$$

We have four different instructions $a = (y := 0)$, $b = (x := y)$, $c = (x := x + 1)$, and $d = (y := y + 1)$. In an abstraction we replace $P$ and $Q$ as

$$
P = abc^\omega bd,
$$

$$
Q = abc^\omega d.
$$

In a sequential run, we will have $P = Q = abc^\omega$ and both $P$ and $Q$ will terminate with overflow on $x$ and value zero for $y$. However, on a parallel machine $P$ and $Q$ may behave differently and we stress that the difference can be observed. Indeed, the compiler can check by syntax (in particular, without evaluating the condition in the \texttt{while} loop) that the instruction $d$ can be performed independently of $c$. Hence, in a parallel execution of $Q$ a (sequential) observer may see $d$ before the overflow on $x$ and, therefore, the value one for $y$ at the end of $Q$. Since this is impossible for $P$, the equality $P = Q$ is not adequate. We see that concatenating $bd$ to $abc^\omega$ yields something different from concatenating $d$ to $abc^\omega$. Therefore, we need a mathematical interpretation where $abc^\omega bd \neq abc^\omega d$ and a notion of concatenation for infinite concurrent processes.

A possible solution based on partial orders seems to be to describe $P$ and $Q$ not by sequences but by infinite graphs (Fig. 1). It is clear that $P$ and $Q$ are different and this difference can be observed since $abd$ is no prefix of $P$ but of $Q$. Besides the difficulty to deal with graphs which may have vertices with infinitely many predecessors, the question of observability shows another weak point of this solution. We would obtain different graphs for programs such as $P = abc^\omega bd$ and $P' = abc^\omega bdb$ although $Q_1PQ_2$ and $Q_1P'Q_2$ coincide on all finite prefixes for all $Q_1, Q_2$. So, the solution
presented in this paper abstracts from graphs as above in the following way. Given a graph, we split it into its (real) part of vertices with finitely many predecessors and its transfinite part of vertices with infinitely many predecessors. The real part can be approximated by finite prefixes. It is the part which can be observed directly within a finite amount of time. From its transfinite part, however, only its alphabet is of importance. More precisely, we remember only those actions which are dependent on actions occurring in the transfinite part. This is some finite additional information.

We will see that such an information is indeed necessary to obtain a mathematically and semantically sound theory. In fact, our theory yields a good interpretation for infinite concurrent processes.

In the special case for full dependencies, we reobtain the classical theory of infinite words where the concatenation is right-absorbent. Furthermore, our theory also includes the parallel composition of independent processes.

In the first section we introduce some notations used throughout. The second section presents the calculus from an informal viewpoint only. It may be skipped, if the reader prefers to go straight into the technical details.

Then we develop our theory which leads to the notion of complex trace from a metric viewpoint. This is a question of personal taste and could be solved differently. Readers who are interested mainly in discrete aspects of the theory may start with the section on dependence graphs. One can then define complex traces as a quotient by the equivalence relation which takes the real part and the letters which depend on the transfinite part into account. This relation turns out to be a congruence. Such an approach leads more directly to the characterization of Theorem 6.6.

1. Preliminaries

A dependence alphabet is a pair \((X, D)\), where \(X\) is a finite alphabet and \(D \subseteq X \times X\) is a reflexive, symmetric dependence relation. We identify \((X, D)\) with a finite undirected graph where \(X\) is the set of vertices and edges are between different dependent letters.
The complement \( I = X \times X \setminus D \) is called the independence relation and the quotient monoid \( M = M(X, D) = X^*/\{ab = ba|(a, b) \in I\} \) is the associated free partially commutative monoid. An element of \( M \) is called a (finite) trace. For a trace \( t \in M \) and \( a \in X \) we denote by \( |t| \) its length, i.e., the number of vertices in the dependence graph of \( t \), and by \( |t|_a \) its \( a \)-length, i.e., the number of occurrences of \( a \) in \( t \). By \( D(a) = \{ b \in X \mid (a, b) \in D \} \) we denote the set of letters dependent on \( a \). This notation is extended to subsets \( A \subseteq X \) by \( D(A) = \bigcup_{a \in A} D(a) \) and to traces \( t \in M \) by \( D(t) = D(\text{alph}(t)) \). If \( p, t \in M \) are traces then \( p^{-1}t \) is defined if and only if \( p \) is a prefix of \( t \), written as \( p \preceq t \). In this case \( p^{-1}t = t' \) for the unique trace \( t' \) such that \( t = pt' \). The following convention of partially defined functions is used. For traces \( p, s, t \in M \) an equation such as \( D(p^{-1}s) = D(p^{-1}t) \) means that either both \( p^{-1}s \) and \( p^{-1}t \) are undefined or both are defined and then the values \( D(p^{-1}s) \) and \( D(p^{-1}t) \) are equal.

2. Informal calculus

The aim of this paper is to extend the notion of finite trace to infinite trace in such a way that a reasonable concatenation of infinite trace is defined.

To see the problem, it suffices to consider very simple special cases of infinite traces \( a^\omega \) with \( a \in X \). As a first idea, one would probably like to start with two laws:

\[
\begin{align*}
\text{(First idea)} & \quad (a^\omega)(b^\omega) = a^\omega \quad \text{if } a \text{ and } b \text{ are dependent,} \\
& \quad (a^\omega)(b^\omega) \neq a^\omega \quad \text{if } a \text{ and } b \text{ are independent.}
\end{align*}
\]

Note that these are very weak assumptions since for independent letters \( a, b \) we want to have something like \((a^\omega)(b^\omega) = (ab)^\omega\). Nevertheless, an approach based on laws I and II above must fail, since then the concatenation cannot be associative.

Consider a dependence alphabet \((X, D)\) where \( D \) is not transitive, i.e., \( M(X, D) \) is not a direct product of free monoids. Then we find \( a, b, c \in X \) such that \((a, b) \in D, (b, c) \in D\), but \((a, c) \notin D\). Using laws I and II, we obtain: \((a^\omega b^\omega)c^\omega = (a^\omega c^\omega) \neq a^\omega = a^\omega b^\omega = (b^\omega c^\omega)\). Thus, \((a^\omega b^\omega)c^\omega \neq a^\omega (b^\omega c^\omega)\) and the concatenation cannot be associative.

If \( D \subseteq X \times X \) is transitive then there is a concatenation satisfying I and II above. In fact, then we have \( D = \bigcup_{i=1}^{k} (X_1 \times X_i) \) for some partition \( X = \bigcup_{i=1}^{k} X_i \) and we have the usual extension of the concatenation to the direct product \( \prod_{i=1}^{k} X_i^\omega \). However, what is the necessary modification if \( D \) is not transitive? Surely, we have to insist on II. (Saying that an \( \omega \)-expression is always right-absorbent is not adequate for concurrent systems.) So, the only solution is to modify I. We have to allow \( a^\omega b^\omega \neq a^\omega \) at least for some dependent letters \( a, b \). In \([1, 2]\) an additional error element \( \bot \) is introduced with \( a^\omega b^\omega = \bot \) and \( \bot \) behaves like a zero. This is a feasible way, but not really convenient, since, for example, \( a^\omega a^\omega = \bot \) in this formalism, instead of \( a^\omega a^\omega = a^\omega \), which is more natural and necessary if we want to generalize the theory of infinite words. Therefore, we will follow another strategy and we will come to the discussion of the error element later.
Let $M = M(X, D)$ be a monoid of finite traces. The prefix relation defines a partial order $\leq$ on $M$ and we may consider limits of strictly increasing sequences of (finite) traces. Later we will call these objects real infinite traces. Now, it turns out that there is no concatenation on real traces which is compatible with taking limits. Therefore, let us consider pairs $(s, A)$, where $s$ is a trace (finite or real infinite) and $A$ is a subset of $X$. The intended semantics is that $s$ is a prefix of some process and $A$ contains possible future actions. Therefore, every new action depending on $A$ is blocked. For a moment the only restriction on $A$ is that it contains all letters which occur in $s$ infinitely often. Thus, for finite $s$ there is no restriction at all. Now, for any $A \subseteq X$ and trace $t$ we can define the maximal prefix of $t$ such that all letters in this prefix are independent of $A$. We denote this prefix by $\mu_A(t)$ and by $\mu_A(t)^{-1}t$ the rest of the trace $t$ which does not belong to this prefix. (This notation is justified by left-cancellativity; see below.) Then, using the abbreviation $\mu$ for $\mu_A(t)$ and $\text{alph}(\mu^{-1}t)$ for the set of letters occurring in the suffix $\mu^{-1}t$, we can define a binary operation on these pairs as follows:

$$(s, A)(t, B) = (s\mu, \text{alph}(\mu^{-1}t) \cup A \cup B).$$

Note that, because of the restriction on $A$, it is no problem to define $s\mu$, say by shuffling. The reader may convince himself that this operation is associative. Furthermore, the pair $(1, \emptyset)$ is the neutral element for this operation and $t \mapsto (t, \emptyset)$ for $t \in M$ defines an embedding of $M$ into this structure. This embedding can be extended to (real) infinite traces by $t \mapsto (t, \text{alphinf}(t))$, where $\text{alphinf}(t)$ denotes the set of letters occurring infinitely often in $t$. Note that $\text{alphinf}(t)$ is the minimal set which is possible by the restriction on the second component. The submonoid which is generated by these objects is called later the set of $\alpha$-complex traces $C_\alpha(X, D)$. Hence, our theory differs from the consideration of ordinary real traces only in the enlargement of the second component.

However, for many purposes the monoid $C_\alpha(X, D)$ is still too large; in particular, it is different from $X^\infty$ for $M = X^*$. The solution is to replace the second component by $D(A)$ which denotes the set of letters dependent on some letter of $A$. We could also view $D(A)$ as the set of letters blocked by $A$. The concatenation then becomes

$$(s, D(A))(t, D(B)) = (s\mu, D(\text{alph}(\mu^{-1}t) \cup A \cup B)),$$

with the same abbreviations as above. We use the notation $D(A)$ for the second component although $A$ need not to be known explicitly. We can determine the set $I(A)$ and the prefix $\mu$ from the knowledge of the set $D(A)$.

In order to explain more about the semantics behind our calculations, let us consider the following example.

Let $s_1, \ldots, s_n \in M$ be finite traces with alphabets $\text{alph}(s_i) = A_i$. Assume that we want to run the infinite process $s = s_1 \cdots s_n$ and another process $t$ concurrently, but the priority for dependent actions is always given to the process $s$. So, what we have to do is to split $t$ into $t = t_1 \cdots t_n t'$ such that in the prefix $t_1 \cdots t_i$ there is no action dependent on any action in $\bigcup_{i=1}^n A_i$. Then we can run $st$ according to Fig. 2. In particular, we can run $s_i$ in parallel to $t_i$ for $i = 1, \ldots, n$. The part $t'$ is delayed at infinity.
It is blocked by the infinite repetition of $s_n$. Now, if there is a third process $u$ with priority behind $t$ then it is clear that, also in a concurrent run, the part of $u$ depending on $A_u \cup \text{alph}(t')$ has to be blocked. Thus, we see the enlargement of the second component and why it is necessary. The next sections will provide us with a mathematical machinery for a rigorous treatment of this situation.

In the example from the introduction the actions $c$ and $d$ are independent. For the procedures $P$ and $Q$ we obtain in “complex abstraction” $P = (abc^n, D(c)) \cdot (bd, 0) = (abc^n, D(b, c, d)) = (abc^n, X)$ and $Q = (abc^n d, D(c)) = (abde^n, X \setminus \{d\})$. In particular, we have $Pd = P$ and $Qd^* \neq Qd^* + 1$ for all $n \geq 1$.

3. Ultra-metric spaces

As in any theory on infinite objects, we have to deal with some topology. We assume that the reader is familiar with the basic concepts of metric spaces, which can be found in any textbook on topology (and, of course, in [3]). Recall that an ultra-metric space is a set $S$ with a distance function $d: S \times S \to \mathbb{R}^+$ such that for all $s, t, u \in S$ we have

(i) $d(s, t) = 0 \iff s = t,$
(ii) $d(s, t) = d(t, s),$ 
(iii) $d(s, u) \leq \max \{d(s, t), d(t, u)\}.$

If we replace (i) by (i'): $d(s, s) = 0$, then $S$ is called a pseudo ultra-metric space.

In this paper we will define the distance by

$$d(s, t) = 2^{-l(s, t)},$$

where, by convention, $2^{-\infty} = 0$ and $l: S \times S \to \mathbb{N} \cup \{\infty\}$ is some function which verifies the following:

(i) $l(s, t) = \infty \iff s = t,$
(ii) $l(s, t) = l(t, s),$ 
(iii) $l(s, u) \geq \min \{l(s, t), l(t, u)\},$

and (i') $l(s, s) = \infty$ instead of (i) for pseudo ultra-metrics. In fact, here (and in many other cases) it is much more convenient to work with the function $l: S \times S \to \mathbb{N} \cup \{\infty\}$.
only; and we call it a log-distance since \( l \) is the logarithm of the distance to the base \( \frac{1}{l} \). An ultra-metric space is complete if every Cauchy sequence converges and compact if every covering by open balls has a finite subcovering. Every metric space can be viewed as a dense subspace of a complete metric space, its Cauchy-completion. A compact metric space is necessarily complete.

4. The bottom-up solution

In this section we define an ultra-metric of \( M(X, D) \) such that the concatenation of traces is uniformly continuous. It follows that the concatenation extends uniquely to the completion of \( M(X, D) \). The objects of interest are, therefore, Cauchy sequence of finite traces. In the next section we will give a concrete interpretation of these objects as abstractions of (infinite) dependence graphs. In particular, the semantics for infinite concurrent processes will be clear from this interpretation.

Let \( s, t \in M \) be finite traces, \( M = M(X, D) \). Then we define

\[
\ell(s, t) = \sup \{ n \in \mathbb{N} \mid \forall p \in M, \ |p| \leq n; \ p \leq s \iff p \leq t \}.
\]

This is the well-known approach of a metric based on prefixes (see [14]). Up to some constant, \( \ell \) could also be defined by using the length of the maximal common prefix in the Foata normal form (see [2]). However, this metric has a serious drawback with respect to the concatenation. Consider the example \((X, D) = \{a, b, c\}\) and two traces \( s = a^nb \) and \( t = a^n \). View \( s \) and \( t \) as concurrent processes and assume we want to run \( sc \) or \( tc \), respectively. Then the action \( b \) at the end of \( s = a^nb \) prevents \( c \) from being started before \( s \) has stopped. The situation with \( t = a^n \) is completely different. We can start \( t \) and the action \( c \) concurrently. It is clear that the behaviour of \( sc = a^nbca \) and \( tc = a^nca = ca^n \) may be different from the very beginning of the execution. Thus, we should not use a metric where the processes \( a^nb \) and \( a^n \) come close together with growing \( n \). The idea is that for the distance between two processes we need alphabetic information on at which point in the executions which actions may start concurrently. Therefore, we define a new log-distance by

\[
\ell(s, t) = \sup \{ n \in \mathbb{N} \mid \forall p \in M, \ |p| < n; \ D(p^{-1}s) = D(p^{-1}t) \}.
\]

(The reason that we switched from \( |p| \leq n \) to \( |p| < n \) is that then we never take the sup over the empty set; so, for finite traces it may be replaced by max.)

**Remark 4.1.** (i) The functions above define ultra-metrics by \( d_{\text{pref}}(s, t) = 2^{-\ell_{\text{pref}}(s, t)} \) and \( d(s, t) = 2^{-\ell(s, t)} \).

(ii) We have \( \ell(s, t) \leq \ell_{\text{pref}}(s, t) + 1 \). In particular, the identity of \( M \) induces a uniformly continuous mapping \((M, d) \to (M, d_{\text{pref}})\) of ultra-metric spaces. Furthermore, for \( s \neq t \) we have \( \ell_{\text{pref}}(s, t) \leq \min \{|s|, |t|\} \). Therefore both metrics induce the discrete topology on \( M \).

(iii) The log-distances \( \ell_{\text{pref}} \) and \( \ell \) define equivalent metrics if and only if \( D \subseteq X \times X \) is an equivalence relation, i.e., if and only if the monoid \( M \) is a direct product of free monoids.
Proof. (i) and (ii) are straightforward.

(iii) Let \( D \) be an equivalence relation, i.e., \( M = \prod_{i=1}^{k} X_i^* \) for some partition \( X = \bigcup_{i=1}^{k} X_i \). It is enough to show that \( l_{\text{pref}}(s, t) \leq l(s, t) + 1 \). Let \( p, s, t \in M \) such that \( |p| < l_{\text{pref}}(s, t) \), and assume that we would have \( D(p^{-1}s) \neq D(p^{-1}t) \). Then for some component \( i \in \{1, \ldots, k\} \) we may assume that \( X_i \subseteq D(p^{-1}s) \) and \( X_i \cap D(p^{-1}t) = \emptyset \). This implies \( pa \leq s \) for some \( a \in X_i \), but \( pa \) is no prefix of \( t \), contradicting \( l_{\text{pref}}(s, t) \geq |p| + 1 \).

Now, consider the (more interesting) case where \( D \) is not an equivalence relation. Then there are letters \( a, b, c \in X \) such that \( (a, b) \in D \), \( (b, c) \in D \), but \( (a, c) \notin D \). We have \( \lim_{n \to \infty} l_{\text{pref}}(a^n b, a^n) = \infty \) whereas \( \lim_{n \to \infty} l((a^n b, a^n)) = 0 \), since \( c \in D(a^n b) \) and \( c \notin D(a^n) \). \( \square \)

It follows from (iii) of the remark above that if \( M \) is a direct product of free monoids then our theory does not yield anything new. But this is exactly our intention. There is a good notion of concatenating tuples of infinite words and we are going to generalize this concept without changing this case.

Let us now state some of our basic results. The proofs are deferred to the next sections.

**Theorem 4.2.** The ultra-metric completion of \( M \) with respect to the log-distance \( 1 \) (or with respect to \( l_{\text{pref}} \)) is compact.

**Theorem 4.3.** The concatenation of \( M \) is uniformly continuous with respect to the log-distance \( 1 \) (but not with respect to \( l_{\text{pref}} \)).

**Corollary 4.4.** The concatenation of \( M \) extends uniquely to a uniformly continuous concatenation of the ultra-metric completion of \( M \) with respect to the log-distance \( 1 \).

5. Dependence graphs revisited

Let \( (X, D) \) be a dependence alphabet. A dependence graph is (an isomorphism class of) a directed acyclic labelled graph \([V, E, \lambda] \) such that \( V \) is a (countable) set of vertices, \( E \subseteq V \times V \) is the set of arcs, \( \lambda : V \to X \) is the labelling and it holds that \( E \cup E^{-1} \cup \text{id}_V = \lambda^{-1}(D) \).

We also identify such an acyclic graph with a labelled partial order \([V, \leq, \lambda] \). In what follows we impose the restriction that this partial order is well-founded. This means that any nonempty subset of vertices has minimal elements. It follows that \( V_a = \{ x \in V | \lambda(x) = a \} \) is a (countable) well-ordered subset of \( V \) for any \( a \in X \); hence, it is uniquely isomorphic to some (countable) ordinal. In particular, if \([V', E', \lambda'] = [V, E, \lambda] \) then there is a unique isomorphism between the underlying concrete graphs. It allows one also to think of dependence graphs in terms of
a standard representation where vertices are of the form \((a, i)\), with \(a \in X\) and a (countable) ordinal \(i\). The set \(G(X, D)\) is a monoid with the concatenation

\[
[V_1, E_1, \lambda_1] [V_2, E_2, \lambda_2] = [V_1 \cup V_2, E_1 \cup E_2 \cup \{(x_1, x_2) \in V_1 \times V_2 | \\
(\lambda_1(x_1), \lambda_2(x_2)) \in D\}, \lambda_1 \cup \lambda_2].
\]

Thus, the concatenation is basically the disjoint union with new arcs from \(V_1\) to \(V_2\) between vertices with dependent labels. The neutral element is the empty graph \(1 = [\emptyset, \emptyset, \emptyset]\). Note that the concatenation is well-defined and associative. It can be visualized as in Fig. 3.

In what follows, we identify the submonoid of finite dependence graphs with the monoid of finite traces. Thus, we assume \(M(X, D) \subseteq G(X, D)\).

**Proposition 5.1.** The monoid \(G(X, D)\) is left-cancellative, i.e., \(st = st' \in G(X, D)\) implies \(t = t'\).

**Proof.** This is a slightly generalized standard argument from ordinal arithmetic: Let \(s, s', t, t'\) be concrete dependence graphs such that \(s \geq s', st \geq s't'\), i.e., \(s = s'\) and \(st = s't'\) in \(G(X, D)\). We have to show that this implies \(t \geq t'\). We may assume that \(s = s'\) as concrete graphs. Let \(h : st \to s't'\) be the isomorphism which induces equality in \(G(X, D)\).

It is enough to show that \(h|_s = id_s\). So, assume \(h|_s \neq id_s\) and let \(x \in s\) be a minimal element with \(h(x) \neq y\). Clearly, \(\lambda(y) = \lambda(x)\); hence, \(x < y\). Now, assume \(x\) would be in the image of \(h\). Then \(x = h(z)\) for some \(x < z\), and we obtain \(y = h(x) < h(z) = x\), contradicting \(x < y\). Hence, the result. \(\square\)

The monoid of dependence graphs has another characteristic which allows one to define operations such as \(L^\rho\) for \(L \subseteq G(X, D)\) directly and without involving limits. Indeed, let \(L \subseteq G(X, D)\) and \(\rho\) be any (countable) ordinal, i.e., any (countable) well-ordered set. Consider the set of mappings from \(\rho\) to \(L\). Any \(\tau : \rho \to L\) yields a dependence graph in the following way.

First, we take the disjoint union \(\bigcup_{i \in \rho} \tau(i)\) and then introduce additional arcs from a vertex \(x \in \tau(i)\) to a vertex \(y \in \tau(j)\) whenever \(i < j\) and \(x, y\) have dependent labels. (Recall the similarity to the definition of a dependence graph of a finite trace starting from its representation as a finite word.) It is clear that this construction yields a well-defined dependence graph and, hence, we can view \(L^\rho \subseteq G(X, D)\). (This approach shows also

\[
\begin{array}{c}
\text{\(x_1\)}
\end{array}
\begin{array}{c}
\text{\(x_2\)}
\end{array}
\begin{array}{c}
\text{[\(V_1, E_1, \lambda_1\)]}
\end{array}
\begin{array}{c}
\text{[\(V_2, E_2, \lambda_2\)]}
\end{array}
\]

Fig. 3. Arc\((x_1, x_2)\) exists if and only if \((\lambda_1(x_1), \lambda_2(x_2)) \in D\).
that it is most natural to define the \( \omega \)-iteration by \( L^\omega = L^* \cup (L \setminus \{1\})^\omega \) whenever the empty trace (word) is in \( L \).

Note that in \( G = G(X, D) \) we have \( aa^\omega = a^\omega \); hence, \( G \) is not right-cancellative. On the other hand, since \( a^\omega a \neq a^\omega \) in \( G \), this monoid cannot be the ultra-metric completion of \( M \) we are looking for. But we are very close. The trick is to define pseudo ultra-metrics.

In fact, we define three different pseudo metrics using straightforward generalizations of the notations defined for finite traces above.

**Definition.** Let \( s, t \in \mathbb{G} \) be dependence graphs, \( \mathbb{G} = \mathbb{G}(X, D) \). Then define the following pseudo log-distances:

1. \( l_s(s, t) = \sup\{n \in \mathbb{N} \mid \forall p \in M, |p| < n: \alpha(p - s) = \alpha(p - t)\} \).
2. \( l(s, t) = \sup\{n \in \mathbb{N} \mid \forall p \in M, |p| < n: D(p - s) = D(p - t)\} \).
3. \( l_{\text{pref}}(s, t) = \sup\{n \in \mathbb{N} \mid \forall p \in M, |p| \leq n: p \leq s \Leftrightarrow p \leq t\} \).

The pseudo log-distance \( l_s \) was not introduced earlier because we are mainly interested in the second pseudo log-distance \( I: \mathbb{G} \times \mathbb{G} \to \mathbb{N} \cup \{\infty\} \). We use the first one since some results are more general and not more difficult to obtain. The use of the third one is to compare our approach with the one of pure prefixes. Observe also that no pseudo distance above can distinguish \( a^\omega a \) from \( a^\omega \).

The pseudo log-distances above define pseudo ultra-metric spaces \( \mathbb{G}_s = (\mathbb{G}(X, D), 2^{-l_s}) \), \( \mathbb{G} = (\mathbb{G}(X, D), 2^{-l}) \), and \( \mathbb{G}_{\text{pref}} = (\mathbb{G}(X, D), 2^{-l_{\text{pref}}}) \) and the set-identity of \( \mathbb{G}(X, D) \) defines uniformly continuous bijections

\[
\mathbb{G}_s \xrightarrow{id} \mathbb{G} \xrightarrow{id} \mathbb{G}_{\text{pref}}.
\]

However, none of these are isomorphisms since neither \( \mathbb{G}_{\text{pref}} \xrightarrow{id} \mathbb{G} \) nor \( \mathbb{G} \xrightarrow{id} \mathbb{G}_s \) are continuous, in general. Indeed, for \( (X, D) = a \longleftarrow b \longleftarrow c \) we have \( l_{\text{pref}}(a^\omega b, a^\omega) = n \), \( l(a^\omega b, a^\omega) = l_s(a^\omega b, a^\omega) = 0 \), \( l(a^\omega bc, a^\omega b) = n + 1 \), \( l_s(a^\omega bc, a^\omega b) = 0 \).

The main theorem of this section states that the concatenation is uniformly continuous for \( \mathbb{G}_s \) and \( \mathbb{G} \) (but not for \( \mathbb{G}_{\text{pref}} \)). In the proof we need Levi's Lemma for the monoid \( \mathbb{G}(X, D) \). Its proof is similar to that of the finite case [5]; see also [6, Sect. 1.3].

**Lemma 5.2** (Levi’s lemma). Let \( x, y, p, q \in \mathbb{G}(X, D) \) be dependence graphs such that \( xy = pq \). Then there are (uniquely determined) dependence graphs \( r, u, v, s \in \mathbb{G}(X, D) \) such that

\[
x = ru, \quad y = sv, \quad p = rv, \quad q = us,
\]

\[
\alpha(u) \times \alpha(v) \subseteq I = X \times X \setminus D.
\]

**Proof.** We may view \( x, y, p, q \) as labelled induced subgraphs of the dependence graph \( xy = pq \). Define \( r = x \cap p, u = x \cap q, v = y \cap p \) and \( s = y \cap q \). The formulae \( ru = x, sv = y \),
Concatenation of infinite traces

rv = p, us = q follow. Since no vertex in u precedes any vertex in v by $u \subseteq q, v \subseteq p$ nor vice versa by $v \subseteq y, u \subseteq x$, we have $\text{alph}(u) \times \text{alph}(v) \subseteq I$. See also Fig. 4. The uniqueness of $r, u, v, s$ is not used below and left to the reader. One obtains the uniqueness by a simple argument from ordinal arithmetic. 

Theorem 5.3. The concatenation is uniformly continuous for $G_s$ and for $G$.

Proof. We show this for $G$ only. The case for $G_s$ is similar (and notational even simpler). Let $x, y, z, t \in G(X, D)$ be dependence graphs, $n \in \mathbb{N}$ such that $l(x, z) \geq n$ and $l(y, t) \geq n$. We show

$$l(xy, zt) = \sup\{m \in \mathbb{N} | \forall p \in M, |p| < m : D(p^{-1}xy) = D(p^{-1}zt) \geq n.$$ 

Let $p \in M, |p| < n$. We may assume that $p^{-1}xy$ is defined. Then $xy = pq$ and by Levi's lemma there are $r, u, v, s \in G(X, D)$ such that $x = ru, y = vs, p = rv, q = us$ and $\text{alph}(u) \times \text{alph}(v) \subseteq I$. Since $p = rv$, we have $|r| < n$ and $|v| < n$. Hence, $D(u) = D(r^{-1}z)$ and $D(s) = D(v^{-1}t)$. Write $u' = r^{-1}z, s' = v^{-1}t$, i.e., $z = ru', t = vs'$. Then $\text{alph}(v) \cap D(u') = 0$ and $u'v = vu'$. Hence, $zt = ru'vs' = ru'vs' = pu's'$. Hence, $p^{-1}zt$ is defined. Furthermore, $D(p^{-1}zt) = D(u's') = D(u') \cup D(s') = D(u) \cup D(s) = D(p^{-1}xy)$. 

Remark 5.4. View $M$ as a subspace of $G_{pref}$. Then, in general, the concatenation of $G_{pref}$ is not continuous and the concatenation of $M$ is not uniformly continuous. (It is trivially continuous since $M$ is discrete.)

Indeed, consider $(X, D) = a \longrightarrow b \longrightarrow c$. Then it holds that $l_{pref}(a^n b, a^n) = n, l_{pref}(a^n bc, a^n c) = l_{pref}(a^n bc, ca^n) = 0$. The concatenation of $G_{pref}$ is not continuous since $l_{pref}(a^n b, a^n) = \infty, l_{pref}(a^n bc, a^n c) = l_{pref}(a^n bc, ca^n) = 0$.

The next steps will be to show that $G_s, G, G_{pref}$ are quasi compact spaces where $M$ is an open, discrete and dense subspace. Furthermore, the space $G_s, G, G_{pref}$ are complete in the sense that any Cauchy sequence has a limit. However, for these purposes it is more convenient to work with the ultra-metric quotients of these spaces. Thus, we consider equivalence classes defined by $l_{q}(s, t) = \infty, l(s, t) = \infty$ and $l_{pref}(s, t) = \infty$, respectively.
6. Real traces, a complex world and the error element

For a dependence graph \( t \in G(X, D) \) and a vertex \( x \in t \) let \( x \downarrow \) be the downward-closed subset of \( t \) with maximal vertex \( x \). We view \( x \downarrow \) as a dependence graph which is a prefix of \( t \). Hence, \( x \downarrow \in G(X, D) \) and \( x \downarrow \leqslant t \).

**Definition.** A real trace is a dependence graph \( t \in G(X, D) \) such that \( x \downarrow \) is finite for all \( x \in t \).

The set of real traces is denoted by \( R(X, D) \). Note that it contains finite and infinite traces. Every dependence graph \( t \in G(X, D) \) has a maximal real prefix denoted by \( \text{Re}(t) = \{x \in t \mid x \downarrow \text{ is finite}\} \). Since \( \text{Re}(t) \) is a prefix of \( t \), we may write \( t = \text{Re}(t) \cdot \text{Tr}(t) \) and call the dependence graph \( \text{Tr}(t) \) the *transfinite* part of \( t \). Obviously, the complement of \( R(X, D) \) in \( G(X, D) \) forms an ideal; hence, we may take the Rees quotient which smashes everything outside \( R(X, D) \) to some (new) zero element. This zero element corresponds to the element called *error* in [11] and is denoted by \( \bot \). Thus, \( R(X, D) \cup \{\bot\} \) is a monoid. It is easy to see that for real traces \( s, t \in R(X, D) \) we have \( st = \bot \) if and only if there are \( (a, b) \in D \) such that \( a \) occurs in \( s \) infinitely often and \( b \) occurs in \( t \) at least once. Note that, in particular, we have \( d^\infty a = \bot \) in \( R(X, D) \cup \{\bot\} \).

Since in no pseudo metrics above there is any distance between \( d^\infty a \) and \( a^\infty \), we cannot define a distance between \( a^\infty \) and \( a^\infty \) through a metric quotient of \( G(X, D) \). This quotient is obtained by considering the *alphabet at infinity* of a dependence graph. It is defined by

\[
\text{alphinf}(t) = \{a \in X^\infty \mid |t_a^\infty = \infty\} \cup \text{alph} (\text{Tr}(t))
\]

Of course, we have \( \text{alphinf}(t) = \emptyset \) if and only if \( t \) is a finite trace. Furthermore, for all dependence graphs \( t \in G(X, D) \) we have \( \text{alphinf}(t) = \bigcap_{p \in t, p \in M} \text{alph}(p^{-1} t) \).

**Theorem 6.1.** Let \( s, t \in G(X, D) \) be dependence graphs. Then we have

1. \( l_s(s, t) = \infty \) \( \iff \) \( \text{Re}(s) = \text{Re}(t) \) and \( \text{alphinf}(s) = \text{alphinf}(t) \),
2. \( l(s, t) = \infty \) \( \iff \) \( \text{Re}(s) = \text{Re}(t) \) and \( D(\text{alphinf}(s)) = D(\text{alphinf}(s)) \),
3. \( l_{\text{pref}}(s, t) = \infty \) \( \iff \) \( \text{Re}(s) = \text{Re}(t) \).

**Proof.** This follows directly from the definition of the metrics and the observation that \( \text{alphinf}(t) = \bigcap_{p \in t, p \in M} \text{alph}(p^{-1} t) \).

**Definition.** An \( \alpha \)-complex (complex) trace is a pair \( (s, A) \), where \( s = \text{Re}(t) \in R(X, D) \) and \( A = \text{alphinf}(t) (A = D(\text{alphinf}(t))) \) for some dependence graph \( t \in G(X, D) \). The set of \( \alpha \)-complex traces is denoted by \( C_\alpha(X, D) \), and the set of complex traces is denoted by \( C(X, D) \).

There is now an obvious way to define log-distances for the sets \( C_\alpha(X, D), C(X, D) \) and \( R(X, D) \) such that we may view them as metric spaces. The explicit formula for
complex traces is given by
\[ l((s, D(A)), (t, D(B))) = \sup \{ n \in \mathbb{N} \mid \forall p \in M, |p| < n \} \]
\[ D(p^{-1}s) \cup D(A) = D(p^{-1}t) \cup D(B) \].

The (similar) formulae for \( C_a(X, D) \) and \( R(X, D) \) are left to the reader. Using these definitions, Theorems 5.3 and 6.1 imply the following corollary.

**Corollary 6.2.** The set of \( \alpha \)-complex traces \( C_\alpha(X, D) \) is the metric quotient of the pseudo ultra-metric space \( \mathbb{G}_\alpha \). The same holds for the set of complex traces \( C(X, D) \) and \( \mathbb{G} \), and for the set of real traces \( \mathbb{R}(X, D) \) and \( \mathbb{G}_{\text{pref}} \), respectively. The spaces \( C_\alpha(X, D) \) and \( C(X, D) \) are monoids where the concatenation is uniformly continuous.

We need a few more notations. For a dependence graph \( t \in \mathbb{G}(X, D) \) we denote by \( \min(t) \) the set of labels of the minimal elements of \( t \). Since these elements commute, we view this set also as a trace \( \min(t) \in M(X, D) \).

By \( \alpha(t) \) we denote the trace which is obtained as follows. For each \( a \in \alpha(t) \) we take the minimal vertex with label \( a \). The induced subgraph of these vertices is a finite dependence graph. Hence, \( \alpha(t) \) is also a finite trace. The following formulae are obvious:

\[ \min(\text{Re}(t)) = \min(t), \]
\[ \min(\min(t)) = \min(t), \]
\[ \min(\alpha(t)) = \min(t), \]
\[ \alpha(\alpha(t)) = \alpha(t), \]
\[ \alpha(\alpha(t)) = \alpha(t). \]

**Remark 6.3.** A pair \( (s, A) \), with \( s \in \mathbb{R}(X, D) \) and \( A \subseteq X \), is an \( \alpha \)-complex (complex) trace if and only if there exists a finite trace \( f \in M \) such that \( A = \alpha(f) \) (\( A = D(\alpha(f)) \)) and \( \min(f) \subseteq \alpha(\infty(s)) \subseteq \alpha(f) \).

**Proof.** Let \( (s, A) \) be such that \( A = \alpha(f) \) for some \( f \in M \) with \( \min(f) \subseteq \alpha(\infty(s)) \subseteq \alpha(f) \). Then \( \text{Re}(sf) = s \) and \( \text{Tr}(sf) = f \). Hence, \( \alpha(\infty(\text{Re}(f))) = \alpha(\infty(f)) \subseteq \alpha(f) \). 

For the other direction let \( t \in \mathbb{G}(X, D) \) be a dependence graph. Write \( t = \text{Re}(t) \text{Tr}(t) \) and \( \text{Re}(t) = \text{pq} \) be any factorization such that \( \alpha(q) = \alpha(\infty(\text{Re}(t))) \). Let \( f = \alpha(\text{qTr}(t)) \); then we have \( \min(f) \subseteq \alpha(\infty(\text{Re}(t))) \subseteq \alpha(f) \) and \( \alpha(\infty(\text{Re}(t))) = \alpha(f) \).

Note that we can demand in the remark above \( f = \alpha(f) \). Hence, \( |f| \leq |X| \). Therefore, it is easy to determine the possible sets \( A \subseteq X \).
The uniformly continuous concatenation of $C_\alpha(X, D)$ and $\mathcal{C}(X, D)$ is inherited from the concatenation of $G(X, D)$. Next we give an explicit formula how to concatenate $\alpha$-complex and complex traces. This formula also shows the semantics of the alphabetic information in the second components. It contains the necessary information for running two infinite processes concurrently.

For $A \subseteq X$ and a real trace $t \in \mathbb{R}(X, D)$ let $\mu_A(t) \leq t$ be the maximal prefix of $t$ such that $\text{alph}(\mu_A(t)) \subseteq I(A) = X \setminus D(A)$. Note that $\mu_A(t)$ is a well-defined real trace. (Again, the notation $\mu_A(t)$ is a slight abuse of language since below only $D(A)$, and hence $I(A)$, will be known, in general.)

Now assume that we want to concatenate $\alpha$-complex traces $(s, A)$ and $(t, B)$. The idea is to split $t$ as $t = \mu_A(t) \cdot t'$ and then to concatenate the real traces $s$ and $\mu_A(t)$. This is possible since for some finite prefix $p \leq s$ we have $\text{alph}(p^{-1}s) \subseteq A$. Hence, $sp\mu_A(t) - p(p^{-1}s)\mu_A(t) = p\mu_A(t)(p^{-1}s)$ and the processes $p^{-1}s$ and $\mu_A(t)$ can run really concurrently after the execution of $p$.

**Theorem 6.4.** Let $(s, A), (t, B) \in C_\alpha(X, D)$ be $\alpha$-complex traces. Then we have

$$(s, A) \cdot (t, B) = (s\mu_A(t), \text{alph}(\mu_A(t)^{-1}t) \cup A \cup B).$$

**Proof.** Replace $A$ by $\alpha \in M$ and $B$ by $\beta \in M$ such that the image of $s\alpha, t\beta \in \mathbb{G}(X, D)$ in $C_\alpha(X, D)$ is $(s, A)$ and $(t, B)$, respectively. Then one verifies that $\text{Re}(s\alpha t\beta) = s\mu_A(t)$ and $\text{alph}(s\alpha t\beta) = \text{alph}(\mu_A(t)^{-1}t) \cup A \cup B)$. □

**Corollary 6.5.** Let $(s, D(A)), (t, D(B))$ be complex traces. Then we have

$$(s, D(A)) \cdot (t, D(B)) = (s\mu_A(t), D(\text{alph}(\mu_A(t)^{-1}t)) \cup D(A) \cup D(B)).$$

**Proof.** The homomorphism $C_\alpha \rightarrow C$ is defined by $(s, A) \mapsto (s, D(A))$. The result follows since $D(A \cup B) = D(A) \cup D(B)$. □

**Example.** Let $(X, D) = a \rightarrow b \rightarrow c \rightarrow d$ and $t = abc(d^\omega)$. Then the complex traces $t$, $t^2$, $t^3$ and $t^4$ in $C_\alpha(X, D)$ and in $C(X, D)$ are visualized in Fig. 5. We have $t^3 = t^4$ in $C(X, D)$, but $t^3 \neq t^4, t^4 = t^5$ in $C_\alpha(X, D)$.

So far, we have always considered $C_\alpha(X, D)$ and $C(X, D)$ without pointing out which is the better model to embed in concurrent processes. However, if the full alphabet $(X, D)$ is known then the preference should be given to $C(X, D)$. The reason is that $C_\alpha(X, D)$ contains some unnecessary information. The role of the alphabet at infinity is restricted to prevent dependent actions from being started. Thus, the $D$ of the alphabet at infinity provides this information. This transfers to the idea of approximating an infinite (real) trace by a finite trace as follows. Let $s \in M(X, D)$ be a finite trace and $t \in \mathbb{R}(X, D)$ be infinite such that $l(s, t) \geq n$. Consider the union of prefixes $p = \bigcup_{1 \leq |p| \leq n, p \leq s} p'$. Then the finite trace $p$ approximates the infinite trace in the
prefix ordering since \( p \preceq t \), and we have the additional information that a letter is independent of \( p^{-1}s \) if and only if it is independent of \( p^{-1}t \). Thus, speaking less formally, we could say that \( s \) represents the pair \((p, D(p^{-1}s))\), which is a good approximation of \( t \). On the other hand, if we know only some finite prefix \( p \) of \( t \) without any alphabetic information about \( p^{-1}t \) then this can never be a good approximation as far as concatenation is concerned. The necessary alphabetic information is reflected exactly by \( C(X, D) \).

Another approach to complex traces has been suggested by Andrzej Tarlecki. Let us call dependence graphs \( s, t \in \mathbb{G}(X, D) \) practically distinguishable if they have a different real part or (recursively) if they become practically distinguishable after concatenation with a (finite) trace.

For example, let \((X, D) = a \rightarrow b \rightarrow c\). Then the dependence graphs \( aw \) and \( a^2b \) are practically distinguishable by concatenating \( c \). But \( a^2b \) is not practically distinguishable from \( a^2b^2, a^2bc \) or, more generally, from any \( a^mbs \) with \( s \in \mathbb{G}(X, D) \). It is easy to see that \( s \) and \( t \) are practically undistinguishable if and only if they yield the same complex trace. This is formally stated and shown in the following theorem, which gives another intrinsic characterization of the set of complex traces.

**Theorem 6.6.** The monoid \( C(X, D) \) of complex traces is the quotient of \( \mathbb{G}(X, D) \) by the largest congruence \( \sim \) such that \( s \sim t \) implies \( \text{Re}(s) = \text{Re}(t) \) for all dependence graphs \( s, t \in \mathbb{G}(X, D) \).

**Proof.** Note first that \( \sim \) exists since having the same real part is an equivalence relation of \( \mathbb{G}(X, D) \). Of course, two dependence graphs denoting the same complex trace are in the relation \( \sim \). So, assume that for some \( s, t \in \mathbb{G}(X, D) \) we would have \( s \sim t \) but \( s \neq t \) in \( C(X, D) \). Then \( \text{Re}(s) = \text{Re}(t) \) and we may assume that for some \( a \in X \) we have \( a \in D(\text{alphinf}(s)) \setminus D(\text{alphinf}(t)) \). Since \( \sim \) is a congruence, we have \( sa \sim ta \). Hence, \( \text{Re}(sa) = \text{Re}(ta) \). But this is a contradiction since \( \text{Re}(s) = \text{Re}(sa) = \text{Re}(ta) \neq \text{Re}(t) = \text{Re}(s) \). \( \square \)
The discussion above shows that the notion of complex trace is the correct level of abstraction if \((X, D)\) is known. If the dependence alphabet is only partly known, then it might be necessary to use the finer model \(C_a(X, D)\) of \(\alpha\)-complex traces. The reason is that it behaves better with respect to embeddings into larger alphabets.

**Remark 6.7.** Let \((X, D), (X', D')\) be dependence alphabets such that \(X \subseteq X'\) and \(D = D' \cap (X \times X)\). Then the inclusion \(\mathbb{G}(X, D) \subseteq \mathbb{G}(X', D')\) induces a topological (metric) injective homomorphism \(C_a(X, D) \to C_a(X', D')\).

Remark 6.7 is not true for complex traces. This can be seen, e.g., from the inclusion of the dependence alphabet \((a \prec b)\) into \((a \prec b \prec c)\). However, as a special case of Theorem 6.6, we obtain that the notion of complex trace generalizes the classical case of a tuple of \(\infty\)-words.

This can also be verified directly. Consider first the case of \((X, D) = (X, X \times X)\), i.e., \(M = M(X, D) = X^\ast\) is a free monoid. Then for \((s, D(A)) \in \mathbb{C}(X, D)\) there are only two possibilities and these are characterized by the real part \(s\). If \(s \in M\) is a finite word, then \(D(A) = \emptyset\). If \(s \in X^\omega\) is an infinite word, then \(D(A) = X\). Hence, in this special case the second component is redundant. We can identify \(C(X, D)\) with the usual construct \(X \odot = X^\ast \cup X^\omega\). More precisely, the canonical mapping \(C(X, X \times X) \to \mathbb{R}(X, X \times X) = X^\omega\) is an isomorphism. Of course, this generalizes to the case of a direct product of free monoids. Thus, we can state the following theorem.

**Theorem 6.8.** Let \((X, D) = \bigcup_{i=1}^k (X_i, X_i \times X_i)\) be a disjoint union, i.e., \(M(X, D) = \prod_{i=1}^k X_i^\ast\). Then the topological monoids \(C(X, D)\) and \(\prod_{i=1}^k X_i^\omega\) are canonically isomorphic.

It is well-known that \(\prod_{i=1}^k X_i^\omega\) is a compact ultra-metric space, where \(\prod_{i=1}^k X_i^\ast\) is an open, discrete and dense subspace; see, for example, [13, 18]. The same results hold for \(C_a(X, D), C(X, D)\) and \(\mathbb{R}(X, D)\) as will be shown next.

### 7. Topological properties

The following results concerning \(\mathbb{R}(X, D)\) with the metric \(d_{\text{pref}}\) were known before (see [2, 14]). They are mentioned here for sake of completeness.

**Theorem 7.1.** Let \((X, D)\) be a dependence alphabet and \(M = M(X, D)\). Then \(M\) is an open, discrete and dense subspace of \(C_a(X, D), C(X, D)\) and \(\mathbb{R}(X, D)\).

**Proof.** Since \(M\) is open and discrete in \(\mathbb{R}(X, D)\), the same holds for \(C(X, D)\) and \(C_a(X, D)\). For density it is enough to consider \(C_a(X, D)\). The following arguments are very close to those used in Remark 6.3. So, let \((s, A)\) be an \(\alpha\)-complex trace and \(f \in M\) such that \(\min(f) \subseteq \text{alphinf}(s) \subseteq \text{alph}(f) = A\). (Note that if \(s \in M\) is finite then \(f = 1\) and...
A = ∅. Now, let \( n \in \mathbb{N} \). We find a factorization \( s = pq \) such that for all \( p' \in M, |p'| < n \) we have \( p' \preceq s \iff p' \preceq p \). In fact, take for \( p \) a prefix of \( s \) which is large enough such that it contains the union of all finite prefixes of length smaller than \( n \) in \( s \). Furthermore, we may assume that \( \text{alph}(q) = \text{alphinf}(s) \). Let \( r = \alpha(q) \) and consider \( \text{pref} \in M \). Since \( \min(r) = \min(q) \) and \( \text{alph}(r) = \text{alph}(q) \), one easily verifies that \( l_q(\text{pref}, sf) \geq n \) in \( G_\alpha \); hence, \( l_q((\text{pref}, \emptyset), (s, A)) \geq n \in C_\alpha(X, D) \) and the result follows.

**Theorem 7.2.** The ultra-metric space \( C_\alpha(X, D) \) is complete.

**Proof.** Let \( (s_i)_{i \geq 1} = (r_i, A_i)_{i \geq 1} \) be a Cauchy sequence in \( C_\alpha(X, D) \). Consider the set of finite traces which are prefixes of all \( r_i \) where \( i \) is large enough, i.e., \( R = \{ p \in M(X, D) \mid |p| \leq r_i \text{ for almost all } i \geq 1 \} \). Clearly, \( R \) is prefix-closed and directed. Thus, \( R \) defines a real trace \( r \in \mathbb{R}(X, D) \) by the union of its elements, \( r = \bigcup_{p \in R} p \). (In the spirit of Mazurkiewicz [16], we could say that \( R \) is the real trace \( r \).) The trace \( r \) is the limit of \( (r_i)_{i \geq 1} \) with respect to the metric \( d_{\text{pref}} \) and for each \( p \in R \) we have \( \min(p^{-1}r) = \min(p^{-1}r_i) \) for almost all \( i \geq 1 \). In general, \( (r, \text{alphinf}(r)) \) is not the limit of \( (s_i)_{i \geq 1} \) in \( C_\alpha(X, D) \). But since \( (s_i)_{i \geq 1} \) is a Cauchy sequence of \( C_\alpha(X, D) \), we see that for each \( p \in R \) there exists some (unique) \( A_p \subseteq X \) such that \( A_p = (\text{alph}(p^{-1}r_i) \cup A_i) \) for almost all \( i \geq 1 \). Define \( A = \bigcap_{p \in R} A_p \). We will show that \( s = (r, A) \) is the limit of \( (s_i)_{i \geq 1} \). First, observe that \( \text{alphinf}(r) \subseteq A \). Since \( A_p \cap A_p^c \subseteq A_p \cap A_p^c \), we find some fixed \( p_0 \in R \) such that \( A = A^c_p \) for all \( p_0 \leq p \in R \). Additionally, we may assume that \( \text{alph}(p_0^{-1}r) = \text{alphinf}(r) \). Now, fix some \( i \geq 1 \) such that \( \min(p_0^{-1}r) = \min(p_0^{-1}r_i) \) and \( A = \text{alph}(p_0^{-1}r_i) \cup A_i \). Then it is easy to see that \( (r, \text{alphinf}(r))(p_0^{-1}r_i, A_i) = (r, A) \). Hence, \( (r, A) \) is an \( x \)-complex trace. The final step is now easy: Let \( n \geq 1 \) and \( q \in M(X, D) \), with \( |q| < n \). Since \( q \preceq r \) if and only if \( q \preceq r_i \) for almost all \( i \geq 1 \), we may assume \( q < r_i \); in fact, we may assume \( p_0 < q \in R \) for \( p_0 \) as above. Thus, \( A = \text{alph}(q^{-1}r) \cup A = A = (\text{alph}(p_0^{-1}r_i) \cup A_i) \) for almost all \( i \geq 1 \); hence, \( l_q(s, s_i) \geq n \) for almost all \( i \geq 1 \), which concludes the proof.

**Corollary 7.3.** The space \( C_\alpha(X, D) \) is the completion of the metric space \( (M, d_\alpha) \).

**Proof.** This is clear since \( M \subseteq C_\alpha(X, D) \) is dense and \( C_\alpha(X, D) \) is complete.

**Theorem 7.4.** The complete ultra-metric space \( C_\alpha(X, D) \) is compact.

**Proof.** Write \( (X, D) = (\bigcup_{i=1}^k X, X_i \times X_i) \) as a covering by cliques such that for all \( a \in X \) there exists some \( i \in \{1, \ldots, k\} \), with \( X_i = \{a\} \). (This will be used below.) For \( t \in M = M(X, D) \) let \( t_i \) denote the canonical image of \( t \) in \( X_i^* \), which is obtained by erasing those letters of \( t \) which do not belong to \( X_i \). The well-known embedding theorem [19, 4, 5] states that \( \varphi : M \to \prod_{i=1}^k X_i^* \), \( \varphi(t) = (t_i)_{i=1, \ldots, k} \) is injective. On the direct product we have the usual log-distance \( l_{\text{pref}} \) given by prefixes, \( l_{\text{pref}}(s_i, t_i) = \min \{ l_{\text{pref}}(s_i, t_i) \mid 1 \leq i \leq k \} \). It should be noted that \( \varphi : (M, l_q) \to (\prod_{i=1}^k X_i^*, l_{\text{pref}}) \) is not continuous. However, we are interested in the inverse of \( \varphi \) restricted to the
image of \( M \). So, let \( \phi: \varphi(M) \to M \) be the inverse of \( \varphi \). Let us show that \( \phi: (\varphi(M), l_{\text{pref}}) \to (M, l_2) \) is uniformly continuous: Consider \( s, t \in M \) and \( n \in \mathbb{N} \) such that \( \min \{ l_{\text{pref}}(s_i, t_i) \mid 1 \leq i \leq k \} \geq n \). Let \( p \in M \) such that \( |p| < n \). We show that \( \alpha(p^{-1} s) = \alpha(p^{-1} t) \). We may assume that \( p \leq s \). Hence, \( p_i \leq s_i \) for all \( 1 \leq i \leq k \) and, therefore, \( p_i \leq t_i \) for all \( 1 \leq i \leq k \). It follows that \( p^{-1} t \) is defined, too. By symmetry, it is enough to show that \( a \in \alpha(p^{-1} s) \) implies \( a \in \alpha(p^{-1} t) \).

Consider the index \( i \) such that \( X_i = \{ a \} \). Then \( p_i = a^m \) for some \( m < n \) and \( a^{m+1} \leq s_i \). Since \( m + 1 < n \) and \( l_{\text{pref}}(s_i, t_i) \geq n \), we have \( a^{m+1} \leq t_i \) and \( a \leq (p^{-1} t)_i \). Hence, \( a \in \alpha(p^{-1} t) \) and the claim follows. Now, we know that \( \prod_{i=1}^k X_i^p \) is compact and the completion of \( \prod_{i=1}^k X_i^p \). Hence, the closure \( \bar{M} \) of \( \varphi(M) \) in \( \prod_{i=1}^k X_i^p \) is compact, too. The uniformly continuous mapping \( \phi: \varphi(M) \to M \) extends uniquely to the completions of \( \varphi(M) \) by \( l_{\text{pref}} \), which is \( \bar{M} \), and \( M \) by \( l_2 \). By Corollary 7.3, the completion of \( (M, d_2) \) is exactly \( C_\alpha(X, D) \). Hence, \( C_\alpha(X, D) \) is compact, being the continuous image of the compact space \( \bar{M} \).

**Corollary 7.5.** The spaces \( C(X, D) \) and \( \mathbb{R}(X, D) \) are complete and compact. The space \( C(X, D) \) is the completion of \( M \) by the log-distance \( l \) and \( \mathbb{R}(X, D) \) is the completion of \( M \) by the log-distance \( l_{\text{pref}} \).

**Proof.** Recall that the identity of dependence graphs induces continuous surjective mappings:

\[
C_\alpha(X, D) \to C(X, D) \to \mathbb{R}(X, D),
\]

\[
(t, A) \mapsto (t, D(A)) \mapsto t.
\]

(For a dependence graph \( t \in \mathbb{G}(X, D) \) the pair \( (s, A) \) is \( (\text{Re}(t), \alpha\text{phinf}(t)) \)). Thus, the compactness of \( C_\alpha(X, D) \) transfers to \( C(X, D) \) and \( \mathbb{R}(X, D) \). If a metric space is compact then it is complete. It is the completion of \( M \) since \( M \) is dense. □

**Remark 7.6.** (i) Since every real trace is a dependence graph, we may view \( \mathbb{R}(X, D) \) as a subset of \( \mathbb{G}(X, D) \). Moreover, \( \mathbb{R}(X, D) \) can be viewed as a subset of \( C_\alpha(X, D) \) and of \( C(X, D) \). Since \( (M, d_{\text{pref}}) \) is dense in \( \mathbb{G}_{\text{pref}} = (\mathbb{G}(X, D), d_{\text{pref}}) \), the compact space \( (\mathbb{R}(X, D), d_{\text{pref}}) \) is dense but not closed in the quasi-compact space \( \mathbb{G}_{\text{pref}} \). The metric spaces \( (\mathbb{R}(X, D), d) \) and \( (\mathbb{R}(X, D), d_j) \) are not compact.

(ii) Let \( t_1, t_2, t_3, \ldots \) be any sequence of \( (\ast) \)-complex traces. Then it is easy to see that the sequence of products \( (s_i)_{i \geq 1} \), with \( s_i = t_1 \cdots t_i \) for \( i \geq 1 \) is a Cauchy sequence. Hence, if we can define the infinite product by the limit of the sequence \( (s_i)_{i \geq 1} \). This allows one to define \( L^w \) for \( (\ast) \)-complex trace language by taking limits. But this is not really necessary. We can also use the \( \omega \)-iteration defined on dependence graphs earlier and then consider the projection to \( (\ast) \)-complex traces. This yields the same definition for \( L^w \) and shows that the \( \omega \)-iteration also commutes with projections.

(iii) Finally, let us briefly discuss why we did not always work with the direct product \( \prod_{i=1}^k X_i^{p_i} \). Say, we start with a covering by cliques of the dependence
Concatenation of infinite traces

alphabet, i.e., we write \( (X, D) = (\bigcup_{i=1}^{k} X_i, \bigcup_{i=1}^{k} (X_i \times X_i)) \). Then the embedding theorem mentioned above yields an embedding

\[
\varphi: M(X, D) \rightarrow \prod_{i=1}^{k} X_i^* \subseteq \prod_{i=1}^{k} X_i^*.
\]

Since \( \prod_{i=1}^{k} X_i^* \) is a compact, ultra-metric space where the concatenation is uniformly continuous, we might think that the closure \( \hat{M} \) of \( \varphi(M(X, D)) \) in \( \prod_{i=1}^{k} X_i^* \) is a good model for infinite traces. However, unfortunately, there are several objections.

First of all, the topology of \( \hat{M} \) depends on the chosen covering by cliques. For example, it may happen that a dependence alphabet has more than one covering by maximal cliques and that these coverings lead to incomparable topologies. Thus, a sequence of (finite) traces may be a Cauchy sequence with respect to one covering by cliques (and, therefore, may have a limit), whereas it is not a Cauchy sequence with respect to another covering.

Second, the “difference” between the set \( R(X, D) \) and \( \hat{M} \) is very large. Note that \( \varphi \) extends uniquely to an injective mapping \( \varphi: R(X, D) \rightarrow \hat{M} \) which may be seen as an inclusion. In \( \hat{M} \) we distinguish between many “nonreal traces”, i.e., between many elements in \( \hat{M} \setminus R(X, D) \), whereas from our viewpoint such a distinction is misleading. For example, consider \( (X, D) = a \rightarrow b \rightarrow c \rightarrow d \) with \( X_1 = \{a, b\} \), \( X_2 = \{b, c\} \), \( X_3 = \{c, d\} \). Then for different \( k \geq 1 \) the Cauchy sequences \( (a^k b^k)_{n \geq 1} \) have different limits \( (a^k b^k, 1) \) in \( X_1^* \times X_2^* \times X_3^* \), whereas we obtain for all \( k \geq 1 \) the same limit \( (a^k, D(\{a, b\})) \) in \( C(X, D) \).

Furthermore, if we concatenate \( a^n b = (a^n, D(\{a, b\})) \) with \( d^n = (d^n, D(\{d\})) \), we obtain the real infinite trace \( (a^n d^n, X) = ((ad)^n, D(\{a, d\})) \). Note that here the concatenation of a nonreal infinite trace with something else yields back a real trace. However, this fits into our viewpoint of approximation. The complex infinite trace \( (a^n, D(\{a, b\})) \) is approximated by some \( a^n \), but with the additional constraint that \( c \)'s are not allowed to run concurrently. Only a concurrent run of \( d \)'s is possible. Now, after the concatenation of \( a^n \) with \( d^n \), this constraint is subsumed. Thus, we obtain the usual trace \( (ad)^n \). Contrary to this, the complement of \( R(X, D) \) in \( X_1^* \times X_2^* \times X_3^* \) is a right-ideal (but, of course, not an ideal); thus, concatenating something to a nonreal trace never yields a real trace. This leads to the following asymmetric situation in the direct product. The sequence \( (a^n a^n b)_{n \geq 1} \) has a limit \( (a^n, b, a^n) \) which is outside \( R(X, D) \) in \( X_1^* \times X_2^* \times X_3^* \) and the sequence \( (a^n c^n b)_{n \geq 1} \) has a limit \( (a^n, c^n, 1) \) which is inside \( R(X, D) \). We think that either both sequences or none of them should have a real limit. In our approach, using \( C(X, D) \), the sequences above have the real infinite limits \( ((ad)^n, X) \) and \( ((ac)^n, X) \), respectively.

Acknowledgments

I thank Paul Gastin, Hendrik-Jan Hoogeboom, Edward Ochmanski, Andrzej Tarlecki and one of the anonymous referees for many detailed comments which helped
to improve the paper. This is greatly acknowledged here. Helpful comments were also
given by Lutz Priese, who has developed similar ideas independently (personal
communication), and by some other participants of the ASMICS-workshop Logic and
Recognizable Sets organized by Wolfgang Thomas at Dersau, October 1990.

Last but not least I thank Harald Hadwiger and Dieter Stein for having typed the
different versions of the manuscript.

References

workshop on Free Partially Commutative Monoids, Kochel am See, October 1989, Report TUM-19002,
Technical University of Munich (1990) 1–10.
21–32.
1990).
Theoretical Aspects of Computer Science (STACS 91), Hamburg, 1991, Lecture Notes in Computer
16th Symp. on Mathematical Foundations of Computer Science (MFCS 91), Kazimierz Dolny, 1991,
de Recherche 640, Université de Paris Sud 1991).
(1985) 1–16.
Semantics of Systems of Concurrent Processes, Lecture Notes in Computer Science, Vol. 469 (Springer
eds., Proc. 8th Ann. Symp. on Theoretical Aspects of Computer Science (STACS 91), Hamburg, 1991,
systems, in: J.W. Bakker et al., eds., Current Trends in Concurrency, Lecture Notes in Computer
[15] A. Mazurkiewicz, Concurrent program schemes and their interpretations, DAIMI Report PB 78,
Aarhus University, Aarhus, 1977.
[16] A. Mazurkiewicz, Trace theory, in: W. Brauer et al., eds., Petri Nets, Applications and Relationship to
279–324.
Programming (ICALP 89), Stresa, 1989, Lecture Notes in Computer Science, Vol. 372 (Springer,