Resource Traces: A Domain for Processes
Sharing Exclusive Resources *

Paul Gastin and Dan Teodosiu

LIAFA, Université Paris 7
2, place Jussieu, F-75252 Paris Cedex 05
{gastin,teodosiu}@liafa.jussieu.fr

Abstract

The domain of explicitly terminated finite and infinite words is commonly used to define denotational semantics for process algebras such as CSP. In this well-known framework the denotational semantics of concurrency is derived via power-domains from that of non-deterministic choice and interleaving to the effect that the denotational semantics of a concurrent process is equal to the set of all its possible finite and infinite sequential behaviours.

In this paper we define a more versatile domain of so called finite and infinite resource traces which allow to capture the concurrent behaviour of a process and encode the static concurrency of a system directly into the domains definition. The approach we present refines previous work of Diekert and Gastin [DG95] on \( \alpha \) - and \( \delta \)-traces.

We start with an alphabet of atomic actions, a set of resources, and a resource map assigning to each action the non-empty subset of resources it uses. Actions that do not share common resources are called independent and considered to be able to execute concurrently. A partially terminated concurrent process is specified by a resource trace which consists of two components: an already observed part represented as an action-labeled partial order (Mazurkiewicz trace), and a guard set containing the resources granted to the process for its further development. A process concatenation is then defined which allows independent actions to execute concurrently. Specification refinement leads to a natural approximation ordering between processes. It confers to the set of all processes the structure of a coherently complete prime algebraic Scott domain whereby process concatenation is Scott-continuous in both arguments. Furthermore, we define a natural ultrametric on processes based on prefix information. The induced topology is shown to be equivalent to the compact Lawson topology induced by the approximation ordering. Process concatenation is moreover shown to be uniformly continuous with respect to the defined ultrametric.

The mathematical theory we develop thus extends the central order and metric properties of the domain of explicitly terminated finite and infinite words which are needed in order to devise truly concurrent semantics for process algebras much in the style of classical CSP semantics.

*This research has been supported by the PROCOPE programme No 95143 ASTRA
1 Introduction

The theory of Mazurkiewicz traces has been recognized over the last decade as an expressive tool for investigating concurrent systems according to the non-interleaving viewpoint (overviews can be found in [Maz87, AR88, Per89, Die90] and in the recent monograph [DR95]). The idea is to start with a finite alphabet $\Sigma$ of actions and an explicit symmetric and irreflexive independence relation $I \subseteq \Sigma \times \Sigma$ specifying those pairs of actions that can execute concurrently. The complement $D = \Sigma \times \Sigma \setminus I$ is called dependence relation. The intended semantics is that performing two independent actions in any order should lead to the same result.

Natural opportunities for defining dependence alphabets arise when considering actions that share exclusive resources from some fixed set $R$. This can be expressed by a resource map $\text{res} : \Sigma \to P(R) \setminus \{\emptyset\}$ assigning to each action the non-empty set of resources it uses. An induced dependence relation is then implicitly defined by setting two actions dependent iff they share some common resource, i.e. $a \; D \; b$ iff $\text{res}(a) \cap \text{res}(b) \neq \emptyset$. Moreover, any dependence relation $D$ can be induced on $\Sigma$ by choosing the set of dependencies as resource set $R = D$ and by defining the resources of an action $a \in \Sigma$ to be precisely the incident dependencies $\text{res}(a) = \{(b, c) \in D \mid b = a \text{ or } c = a\} \in P(R) \setminus \{\emptyset\}$.

Finite sequences of actions which can be mutually transformed into each other by commuting consecutive independent actions are considered to represent different sequential behaviours of one and the same concurrent process. Thus, finite concurrent processes are mathematically described as elements of the quotient monoid $(\text{I} \; \text{M}(\Sigma, D), \cdot) = (\Sigma^*, \cdot)/\{ab = ba \mid (a, b) \in I\}$, called the monoid of finite Mazurkiewicz traces over the dependence alphabet $(\Sigma, D)$. The concatenation operation on finite traces is inherited from $\Sigma^*$ and can be regarded as a weak sequential process composition: only dependent actions are synchronized whereas at the process level no explicit synchronization is performed, thus retaining the maximal possible concurrency for the composed process. It models in fact very well both extreme cases namely that of sequential composition, in the word case $(\Sigma^*, \cdot)$ of full dependency between all actions ($D = \Sigma \times \Sigma$), and that of parallel composition, in the vector-addition case $(\Sigma^N, +)$ of total independency between all actions ($I = \Sigma \times \Sigma \setminus 1\_\Sigma$), thereby offering a unifying language for generalizing various results previously established in either finite automata or Petri-net theory.

This versatile nature of trace theory accounts for its main attractiveness as a model for concurrency and has led over the last decade to increasing interest in using traces as a denotational domain for defining truly concurrent semantics of CSP-similar process algebras. However, apart from the monoidal structure modeling process concatenation, additional properties must be satisfied by a denotational domain in order to be able to complete such a task.

To make the picture more precise let us consider a minimal process term algebra build up from the set of actions $\Sigma$ and allowing for concatenation of processes, process variables from a set $V$ and recursive definitions, as expressed by the BNF-like syntax $p ::= a \mid p \cdot p \mid x \mid \text{rec } x.p$ where $a \in \Sigma$ and $x \in V$ denote actions and variables respectively. A valid process term would be for instance $p = \text{rec } x.(a \cdot x) \cdot y$ whose single free variable is $y$. Finitary process terms are those which contain no recursion such as $p = (a \cdot x) \cdot b$. Closed process terms are those which contain no free variables such as $p = \text{rec } x.(a \cdot x) \cdot b$.

A denotational semantics for this process term algebra requires to fix a ground domain.
\( \mathcal{D} \) and to specify a denotation \([p] : \mathcal{D}^V \rightarrow \mathcal{D} \) for every term \( p \) of the language. Elements \( \sigma \in \mathcal{D}^V = V \rightarrow \mathcal{D} \) which supply for every variable \( x \in V \) a value \( \sigma(x) \in \mathcal{D} \) from the ground domain are called environments. The semantics of a process term \( p \) is therefore a function which to every environment \( \sigma \in \mathcal{D}^V \) assigns a value from the ground domain \([p](\sigma) \in \mathcal{D} \). Choosing for instance for every \( \sigma \) \( \mathcal{D} \) denotes the environment obtained from \( \sigma \) by changing its value in \( x \) to \( y \). In order to make sure that such fixed points always exist further structure of either order or metric nature has to be imposed on the ground domain and the mappings involved leading to two well-known formally similar denotational approaches.

In the historically first, order-based approach, originally laid out by D. Scott for the purpose of modeling the \( \lambda \)-calculus, the ground domain \( \mathcal{D} \) is endowed with the structure of a complete algebraic partial order also called a Scott domain. Choosing \([\_] : \mathcal{D}^2 \rightarrow \mathcal{D} \) to be a Scott-continuous operator makes all finitary process terms define Scott-continuous evaluation mappings. The theorem of Tarski, Knaster and Scott stating that every Scott-continuous self-map of a Scott domain has a least fixed point and that this functional is in turn Scott-continuous, allows then to define the denotation of a recursive process term as the least fixed point of its evaluation mapping.

The second, metric-based approach, which stemmed from classical functional analysis, consists in endowing the ground domain with the structure of a complete metric space. Choosing now \([\_] : \mathcal{D}^2 \rightarrow \mathcal{D} \) to be a contractive operator makes all guarded finitary process terms define contractive evaluation mappings. Banach’s fixed point theorem stating that every contractive self-map of a complete metric space has a unique fixed point, allows then to define the denotation of a guarded recursive process term as the unique fixed point of its evaluation mapping.

Unfortunately, neither of the two denotational frameworks just sketched readily applies to finite traces and their concatenation defined above. The main difficulty resides in defining an order/metric structure on the set of finite traces such that the concatenation operation be Scott-continuous/contractive. Surmounting this obstacle called for a number of gradual adjustments of the original trace model, which we next briefly review.

A natural partial order on finite traces over \((\Sigma, D)\) generalizing the one used for words is the prefix ordering derived from the concatenation operation which is defined for all \( x, y \in \mathbb{M}(\Sigma, D) \) by \( x \leq y \) iff there exists \( z \in \mathbb{M}(\Sigma, D) \) such that \( x \cdot z = y \). As in the well-known word case, the prefix order structure \((\mathbb{M}(\Sigma, D), \leq)\) is algebraic but not complete and, likewise, the natural prefix ultrametric structure \((\mathbb{M}(\Sigma, D), d_{\text{pref}})\) fails to be complete. Applying the standard ideal completion device to \((\mathbb{M}(\Sigma, D), \leq)\) leads to the Scott-domain of real traces over \((\Sigma, D)\) denoted by \((\mathbb{R}(\Sigma, D), \leq)\). Similarly, the standard Cauchy-completion of \((\mathbb{M}(\Sigma, D), d_{\text{pref}})\) leads to a complete prefix ultrametric structure \((\mathbb{R}(\Sigma, D), d_{\text{pref}})\) on real traces. Unfortunately, though having the right order and metric structure, the set of real traces lacks the essential monoidal structure since a concatenation operation extending that for finite traces can only be partially defined. The monoidal structure can be mended by considering the monoid of complex traces \([\text{Die93b}]\), but then, in exchange, the order structure defined by the prefix relation, while being complete, is
no more algebraic [GP92, Die93a, Teo93]. Moreover, a natural prefix-based ultrametric
structure \((\mathcal{C}(\Sigma, D), d_{\text{pref}})\) can be defined on complex traces, such that the concatenation
operation is uniformly continuous, yet, non-contractive.

In addition to all these difficulties in defining a trace domain having convenient order
and monoid structures comes the fact that in all cases the concatenation operation is not
even monotone, let alone Scott-continuous, with respect to the prefix ordering. This comes
indeed as no surprise, since it is already the case for the classical domain of finite words
\(\Sigma^*\), as illustrated by the simple example \(\varepsilon \leq a, b \leq b\) but \(\varepsilon \cdot b = b \not\leq ab = a \cdot b\). The, by
now, classical solution employed in order to induce a Scott-domain structure on words and
to render word concatenation Scott-continuous is to go over to the domain of explicitly
terminated finite and infinite words \(\Sigma^\infty \bot \cup \Sigma^* \top\), whereby the trailing symbols \(\top\) and \(\bot\) respectively signal word termination and non-termination.

This technique of explicit termination has been extended to the setting of trace theory
in [Die93a, DG95], where the domains of \(\alpha\)- and \(\delta\)-traces have been defined. Since our aim
is to describe concurrent systems, we are not only concerned with full termination but
also with partial termination. A partially terminated trace consist of an already observed
trace and some alphabetical restriction on the continuation, which is operationally similar
to the above termination symbols and guarantees the Scott-continuity of the defined con-
catenation operation. The approach is in fact similar to the failure (or ready) semantics
used for process algebras like CCS and CSP [Hoa85, Mil80]. The difference is that whereas
in failure semantics the alphabetical restriction applies merely to the next process step, in
the trace models it persists on all future steps of the process.

The domain of resource traces proposed in this paper follows the general line of thought
developed in [DG95]. However, it uses in an essential way the representation of dependence
alphabets in terms of resource mappings in order to define appropriate monoidal and or-
der/metric structures on the set of resource traces. The concatenation operation we define
is associative and has a neutral element thus conferring to the set of resource traces the
desired monoidal structure. Also, an approximation ordering is defined which exhibits on
the set of traces the structure of a coherently complete prime algebraic Scott-domain. The
concatenation operation is then proved to be Scott-continuous with respect to the order
structure. Furthermore, an ultrametric distance is defined which turns the set of traces
into a compact (hence complete) separable ultrametric space. The concatenation opera-
tion is subsequently proved to be uniformly continuous with respect to the ultrametric
structure. In addition, the topology induced by the ultrametric is shown to coincide with
the compact Lawson topology [Law88] associated with the approximation ordering, which
allows to easily transfer convergence results between the order and metric structures.

The body of order and metric properties thus established for the domain of resource
traces opens the way to devising truly concurrent semantics for processes algebras similar
in style and spirit to CSP. A full development of such an approach to true concurrency,
containing a denotational and a matching operational process semantics, is presented
in [GM99].

We use a rich mathematical arsenal comprising techniques from algebra, domain theory
and topology in order to present a generalization of the domain of explicitly terminated
finite and infinite words which promises to be adequate from an automata theory as well
as from a process algebra perspective.
2 Domains

In this section we briefly introduce some standard terminology of domain theory in order to lay down the context. Subsequently, we prove two general domain-theoretic propositions which will later on ease our investigations.

A partial order \((X, \leq)\) is a set \(X\) equipped with a reflexive, antisymmetric and transitive binary relation \(\leq\). If \(x \leq y\), we shall say that \(x\) is less than or below \(y\) and that \(y\) is greater than or above \(x\). The lower set (upper set resp.) of a subset \(Y \subseteq X\) is \(\downarrow Y = \{x \in X \mid x \leq y\text{ for some }y \in Y\}\) \((\uparrow Y = \{x \in X \mid x \geq y\text{ for some }y \in Y\}\text{ resp.})\). A subset \(Y \subseteq X\) is downward closed (upward closed resp.) iff \(\downarrow Y = Y\) \((\uparrow Y = Y\text{ resp.})\).

A subset \(Y \subseteq X\) is upper bounded (lower bounded resp.) by an element \(x \in X\) iff \(y \leq x\) \((x \leq y\text{ resp.})\) for all \(y \in Y\). It is directed (coherent resp.) iff it is non-empty and for all \(x, y \in Y\) there exists \(z \in Y\) \((z \in X\text{ resp.})\) such that \(x \leq z\) and \(y \leq z\). Whenever they exist the least upper bound and the greatest lower bound of a subset \(Y \subseteq X\) will be denoted by \(\bigvee Y\) and \(\bigwedge Y\), respectively.

The partial order \((X, \leq)\) is a coherently complete partial order (CCPO), directed complete partial order (DCPO), bounded complete partial order (BCPO) iff every coherent subset, directed subset, bounded subset, respectively, has a least upper bound.

A BCPO \((X, \leq)\) is bounded distributive iff for all bounded subsets \(Y \cup \{x\} \subseteq X\), we have \(x \land (\bigvee Y) = \bigvee(x \land Y)\) where \(x \land Y = \{x \land y \mid y \in Y\}\).

An element \(x \in X\) is irreducible (prime resp.) iff for all subsets \(Y \subseteq X\) having a least upper bound, \(x = \bigvee Y\) \((x \leq \bigvee Y\text{ resp.})\) implies \(x \leq y\) for some \(y \in Y\). It is compact, iff for all directed subsets \(Y \subseteq X\) having a least upper bound, \(x \leq \bigvee Y\) implies \(x \leq y\) for some \(y \in Y\). The set of all irreducible, prime, compact elements in \(X\) will be denoted by \(\text{Irr}(x)\), \(\text{Prm}(x)\), \(\text{Kmp}(x)\), respectively, the sets of those below \(x \in X\) by \(\text{Irr}(x)\), \(\text{Prm}(x)\), \(\text{Kmp}(x)\), respectively.

The partial order \((X, \leq)\) is \(i\)-algebraic \((p\)-algebraic resp.) iff \(x = \bigvee \text{Irr}(x)\) \((x = \bigvee \text{Prm}(x)\text{ resp.})\) for all \(x \in X\). It is \(k\)-algebraic iff \(\text{Kmp}(x)\) is directed and \(x = \bigvee \text{Kmp}(x)\) for all \(x \in X\). A \(k\)-algebraic DCPO is called a Scott domain.

A mapping \(F : (X, \leq) \to (X', \leq')\) between two partial orders is monotone iff for all \(x, y \in X\), \(x \leq y\) implies \(F(x) \leq' F(y)\). It is continuous iff for all directed sets \(Y \subseteq X\), such that \(\bigvee Y\) exists, \(\bigvee F(Y)\) exists, and \(F(\bigvee Y) = \bigvee F(Y)\). A pair of mappings \(E : (X, \leq) \to (X', \leq')\), \(P : (X', \leq') \to (X, \leq)\) between two partial orders is an embedding-projection pair iff both mappings are monotone and \(P \circ E = \mathbb{I}_X\), \(E \circ P \leq' \mathbb{I}_{X'}\). \(E\) is then called the embedding and \(P\) the projection.

Let us stop for an instant and comment on the above definitions. It follows directly from the definitions that any directed or bounded set is coherent. Therefore, CCPOs are DCPOs and BCPOs at the same time. Furthermore, we see that primes are irreducible and compact, so \(p\)-algebraicity implies \(i\)-algebraicity and, in the case of BCPOs, \(k\)-algebraicity. The later statement is a consequence of the fact that in a \(p\)-algebraic BCPO, compacts are precisely finite suprema of primes, i.e. for all \(x \in X\) \(\text{Kmp}(x) = \{\bigvee P \mid P\text{ finite and }P \subseteq \text{Prm}(x)\}\), which is indeed directed.

Our models turn out to be \(p\)-algebraic coherently complete partial orders (CCPOs), hence enjoy the most powerful properties of completeness and algebraicity stated above. Whereas coherently completeness is fairly easy to prove, \(p\)-algebraicity is rather difficult.
We shall, therefore, exploit to this end a further property shared by all our models through their very construction, namely well-foundedness. A partial order is well-founded iff every descending sequence of elements is finite or, equivalently, iff every non-empty subset has a minimal element. As we shall see in a moment, i-algebraicity is automatically satisfied under this assumption (Proposition 2.1), so we are left with showing that all irreducibles are prime in order to conclude p-algebraicity. This implication is then further reduced to checking bounded distributivity (Proposition 2.2). The latter task can finally be easily accomplished within our models, by using the form of the \( \wedge \) and \( \vee \) operators on bounded subsets. We now proceed to the details.

**Proposition 2.1** Every well-founded partial order is i-algebraic.

**Proof:** Let \( (X, \leq) \) be a well-founded partial order. We have to prove that every element \( x \in X \) is a supremum of irreducible elements. We use well-founded induction. Assume that every element strictly below \( x \) is a supremum of irreducibles. We distinguish two cases. Either \( x \) itself is irreducible, in which case the one-element set \( \{x\} \) will do, or otherwise \( x \) is not irreducible. Then, there must exist by definition a subset \( Y \subseteq X \) having as supremum \( \bigvee Y = x \) such that \( x \notin Y \). But then all elements \( y \in Y \) are necessarily strictly below \( x \). Applying now the induction hypothesis, we obtain for every \( y \in Y \) a subset of irreducibles having as supremum \( \bigvee I_y = y \). Taking the union of all these subsets ranging over \( y \in Y \) yields a subset of irreducibles whose supremum is \( x \), thus concluding the induction proof.

It is well-known that a BCPO is always bounded distributive if it is p-algebraic. The next proposition shows that additionally assuming well-foundedness makes the two conditions equivalent.

**Proposition 2.2** A well-founded BCPO is p-algebraic iff it is bounded distributive.

**Proof:** Let \( (X, \leq) \) be a well-founded BCPO.

We first prove the direct implication. Consider any \( x \in X, Y \subseteq X \) such that \( \{x\} \cup Y \) is bounded. First, for all \( y \in Y \) we have \( x \wedge y \leq x \wedge (\bigvee Y) \), which implies \( \bigvee (x \wedge Y) \leq x \wedge (\bigvee Y) \). Secondly, for any prime \( p \leq x \wedge (\bigvee Y) \) we have \( p \leq x \) and \( p \leq \bigvee Y \), hence \( p \leq y \) for some \( y \in Y \). Therefore, \( p \leq x \wedge y \leq \bigvee (x \wedge Y) \). Since \( (X, \leq) \) is p-algebraic it follows that \( x \wedge (\bigvee Y) = \bigvee \Prm(x \wedge (\bigvee Y)) \leq \bigvee (x \wedge Y) \).

We now prove the converse implication. In view of the preceding proposition, all we have to show is that irreducibles are prime. Let, therefore, \( x \in X \) be irreducible and suppose that \( x \leq \bigvee Y \) for some set \( Y \subseteq X \). Then \( \bigvee Y \) is a common upper bound to \( x \) and all elements of \( Y \). Hence, \( x = x \wedge (\bigvee Y) = \bigvee (x \wedge Y) \) by bounded distributivity. Since \( x \) is irreducible, it follows that there exists \( y \in Y \), such that \( x = x \wedge y \), which in turn implies \( x \leq y \in Y \), showing thus that \( x \) is prime.

### 3 Traces

In this section we review the basics of trace theory and stress upon some results necessary to the subsequent presentation of our model. We assume the reader to be familiar with
the notions involved and, therefore, only sketch the general ideas. Extensive treatments of
the subject can be found in [Maz87, AR88, Per89, Die90, DR95].

We fix in the following a finite alphabet of actions \( \Sigma \) and a symmetric and irreducible
relation \( I \) on \( \Sigma \) called independence relation. The symmetric and reflexive complement
of \( I \) in \( \Sigma \times \Sigma \) is denoted by \( D \) and called dependence relation. The pairs \( (\Sigma, D) \) and
\( (\Sigma, I) \) are called dependence alphabet and independence alphabet, respectively. For \( A \subseteq \Sigma \) we denote
by \( D(A) \) the set of actions dependent on some action in \( A \) and by \( I(A) \) the set of actions
independent of all actions in \( A \).

A trace over \( (\Sigma, D) \) is (the isomorphism class of) a labeled directed graph \( x = [V, E, \lambda] \),
where \( V \) is a countable set of events, \( E \) is a well-founded synchronization relation on \( V \),
and \( \lambda : V \rightarrow \Sigma \) is an event-labeling satisfying the well-synchronization condition
\[
(\lambda(p), \lambda(q)) \in D \iff (p, q) \in E \text{ or } (q, p) \in E \text{ or } p = q
\]
for any \( p, q \in V \). The length of the trace \( x \) is \( \vert x \vert = \vert V \vert \). A trace is finite iff its length is
finite. The set of traces over \( (\Sigma, D) \) is denoted by \( G(\Sigma, D) \) or simply \( G \), the set of finite
traces by \( \mathbf{M}(\Sigma, D) \) or simply \( \mathbf{M} \).

As \( E \) is assumed to be well-founded, it will necessarily be acyclic. Hence, its reflexive-
transitive closure \( E^* \) defines a well-founded partial ordering on the events of the trace \( x \).
The past \( \downarrow v \) of the event \( v \in V \) is the set of all events below \( v \) with respect to the ordering
\( E^* \). The set of maximal actions of the trace \( x \), denoted by \( \max(x) \), is the set of labels of
maximal events in \( x \) with respect to the ordering \( E^* \). We define the alphabet of the trace \( x \)
by \( \text{alph}(x) = \lambda(V) \) and set \( D(x) = D(\text{alph}(x)) \) and \( I(x) = I(\text{alph}(x)) \).

The set of traces \( G \) forms a monoid together with the concatenation operation defined
by \( [V_1, E_1, \lambda_1] \cdot [V_2, E_2, \lambda_2] = [V, E, \lambda] \), where \( V = V_1 \cup V_2 \), \( \lambda = \lambda_1 \cup \lambda_2 \), \( E = E_1 \cup E_2 \cup (V_1 \times V_2 \cap \lambda^{-1}(D)) \), and the empty trace \( 1 = [\emptyset, \emptyset, \emptyset] \) as neutral element. The set of finite
traces \( \mathbf{N} \) is a countable submonoid of \( G \). It is actually isomorphic to the factor monoid
obtained from \( \Sigma^* \) by setting two words equivalent, if they can be mutually transformed into
each other by repeatedly commuting consecutive independent actions, that is \( \mathbf{M}(\Sigma, D) = \Sigma^*/\{ab = ba \mid (a, b) \in I\} \).

The monoid \( G \) is left-cancellative (i.e. \( xy = xz \) implies \( y = z \)) and group free (i.e. \( xy = 1 \)
implies \( x = y = 1 \)). The reflexive and transitive prefix relation, defined by \( x \leq y \) iff there
exists \( z \) such that \( xz = y \), is therefore antisymmetric, hence defines a partial order on
\( G \). Moreover, for any traces \( x \leq y \) we can define \( x^{-1}y \) by the equation \( x(x^{-1}y) = y \).
For all \( x \leq y \leq z \) we then have \( x^{-1}z = (x^{-1}y)(y^{-1}z) \), which is the standard usage of
this notation. Furthermore, as a consequence of the well-known Levi Lemma, we have
\( (x \wedge y)^{-1}y = x^{-1}(x \vee y) \) for any traces \( x \) and \( y \) having a common upper bound.

One can easily see that the prefix ordering is well-founded. The properties of \( \mathbf{M} \) and \( G \)
with respect to the prefix ordering have been investigated in [Maz87, Gas90, Kw90, GR93,
GP95], where it has been shown that \( (\mathbf{M}, \leq) \) and \( (G, \leq) \) are p-algebraic BCPOs. Since
any prefix of a finite trace is finite, \( \mathbf{M} \) is a downward closed subset of \( G \). Moreover, all
finite traces can be shown to be compact. Neither of the two monoids is directed complete
with respect to the prefix ordering. The reason is in fact purely set-theoretical: in both
cases the set of ordinal numbers corresponding to the ordered sets of underlying events
does not contain its limit points.

We can nevertheless consider for any trace \( x \in G \) the (bounded) set of all its finite
prefixes and extract its real part

\[ \text{Re}(x) = \bigvee \{ k \mid k \in \mathbb{M}, k \leq x \}. \]

This defines a mapping \( \text{Re} : \mathcal{G} \to \mathcal{G}, x \mapsto \text{Re}(x) \). A trace \( x \in \mathcal{G} \) is real iff it equals its real part. This is equivalent to demanding of all events in the trace to have finite past. The set of real traces, denoted by \( \mathbb{R}(\Sigma, D) \) or simply \( \mathbb{R} \), is a directed complete, downward closed subset of \( \mathcal{G} \) containing all finite traces. The compact real traces are precisely the finite ones whereas the prime real traces are the non-empty finite traces having exactly one maximal event. Moreover, \( \mathbb{R} \) is a p-algebraic CCPO isomorphic to the ideal completion of \( \mathbb{M} \), which shows that \( \mathbb{R} \) is a good denotational domain from an order-theoretic perspective. Unfortunately, we completely lose the monoidal structure in passing over to \( \mathbb{R} \), as the concatenation operation is no longer an internal operation on \( \mathbb{R} \).

An even more penalizing deficiency, from a denotational point of view, is caused by the fact that the monoidal and order structures of \( \mathbb{M} \) and \( \mathcal{G} \) are not compatible, since the concatenation is not even monotone, let alone continuous, with respect to the prefix ordering. This comes indeed as no surprise, since it is already the case for the classical domain of finite words \( \Sigma^* \), when all actions are considered to be dependent, as illustrated by the simple example \( \varepsilon \leq a, b \leq b \) but \( \varepsilon \cdot b = b \not\leq ab = a \cdot b \). The direct consequence of this circumstance is that we can not use a compositional approach to defining the denotational semantics of recursive process terms using trace concatenation, as was sketched in the introduction.

The model we present in the next sections provides a solution to these problems. It refines the approach proposed in [Die93a, DG95], meanwhile avoiding some of its inconveniences which we discuss in the last section.

4 The Calculus of Resource Traces

A solution for making the concatenation monotonous (continuous) in the word case has been known for quite a while. It consists in augmenting the alphabet \( \Sigma \) with an explicit termination symbol \( \top \) and non-termination symbol \( \bot \) and considering explicitly terminated finite and infinite words, defined as elements of \( \Sigma^\infty \bot \cup \Sigma^* \top \), where \( \Sigma^\infty = \Sigma^* \cup \Sigma^\omega \). Words ending with \( \top \) are finite and called terminated, those ending with \( \bot \) are possibly infinite and called non-terminated. The concatenation is defined by \( u \bot \cdot w = u \bot \) and \( v \top \cdot w = vw \) for all \( u \in \Sigma^\infty, v \in \Sigma^* \) and \( w \in \Sigma^\infty \bot \cup \Sigma^* \top \). This is formally equivalent to setting the non-termination symbol \( \bot \) to be a left zero and the termination symbol \( \top \) to be a left unit. \( \Sigma^\infty \bot \cup \Sigma^* \top \) is a monoid with respect to the concatenation and \( \top \) as unit. An approximation order can be introduced by putting \( u \bot \sqsubseteq uv \) for any \( u \in \Sigma^* \) and \( v \in \Sigma^\infty \bot \cup \Sigma^* \top \), which means that only finite non-terminated words can grow. \( \Sigma^\infty \bot \cup \Sigma^* \top \) is a p-algebraic coherently complete domain with respect to the approximation order. The compact elements are exactly the finite words, that is the elements of \( \Sigma^* \{ \bot, \top \} \). Moreover, concatenation can be shown to be continuous with respect to the approximation order. This formalism has been used as a foundation for specifying denotational semantics for process calculi such as CCS and CSP. The model we present tries to extend this line of thought to the trace model of computation.

Let us fix in the following a finite alphabet of actions \( \Sigma \), a finite set of resources (registers) \( \mathcal{R} \) and a resource mapping \( \text{res} : \Sigma \to \mathcal{P}(\mathcal{R}) \setminus \{ \emptyset \} \) assigning to every action in
\( \Sigma \) a non-empty set of resources from \( \mathcal{R} \) of which it makes use. The resource mapping \( \text{res} \) induces an implicit dependence relation \( D \) on \( \Sigma \) by setting two actions dependent iff they use some common resource, i.e. \( a \cdot b \) iff \( \text{res}(a) \cap \text{res}(b) \neq \emptyset \) for all \( a, b \in \Sigma \). The resource mapping \( \text{res} \) will be called a representation of the induced dependence relation \( D \).

We extend the resource mapping to \( \mathcal{P}(\Sigma) \) and \( \mathcal{G}(\Sigma, D) \) by \( \text{res} : \mathcal{P}(\Sigma) \rightarrow \mathcal{P}(\mathcal{R}), A \mapsto \text{res}(A) = \bigcup_{a \in A} \text{res}(a) \) and \( \text{res} : \mathcal{G}(\Sigma, D) \rightarrow \mathcal{P}(\mathcal{R}), x \mapsto \text{res}(x) = \text{res}(\alpha(x)) \). Furthermore, we define the resources at infinity of a trace by

\[
\text{resinf}(x) = \bigcap \{ \text{res}(k^{-1}x) \mid k \in \mathbb{M}, k \leq x \}
\]

This defines a mapping \( \mathcal{G} \rightarrow \mathcal{P}(\mathcal{R}), x \mapsto \text{resinf}(x) \). For real traces, the resources at infinity of a trace can be identified as being the set of resources used infinitely often by action labelings of events in the trace. For non-real traces, it additionally contains the resources of the non-real part, hence \( \text{resinf}(x) = \text{resinf}(\text{Re}(x)) \cup \text{res}(\text{Re}(x)^{-1}x) \). Furthermore, as can be easily shown, we have \( \text{resinf}(x) \subseteq \text{resinf}(y) \subseteq \text{resinf}(x) \cup \text{res}(x^{-1}y) \) whenever \( x \leq y \).

**Definition 4.1** A resource trace, \( \rho \)-trace for short, over \( (\Sigma, \mathcal{R}, \text{res}) \) is a pair \( x = (r, R) \), where \( r \in \mathcal{R}(\Sigma, D) \) is a real trace over \( (\Sigma, D) \), and \( R \subseteq \mathcal{R} \) is a subset of resources, such that \( \text{resinf}(r) \subseteq R \). The trace \( r \) is denoted by \( \text{Re}(x) \) and called the real part of \( x \). The set \( R \) is denoted by \( \text{Im}(x) \) and called the imaginary part of \( x \). The length \( |x| \) of \( x \) is the length \( |r| \) of its real part. The set of resource traces over \( (\Sigma, \mathcal{R}, \text{res}) \) is denoted by \( \mathcal{F}_\rho(\Sigma, \mathcal{R}, \text{res}) \), or simply \( \mathcal{F}_\rho \).

We should think of a \( \rho \)-trace as an approximation of a terminated process. The first component is a real trace describing the part of the process which is certain. The second component is the set of resources granted to the process for its continuation. Thus, \( \rho \)-traces can be viewed as specifications over traces. The \( \rho \)-trace \( (r, A) \) represents in this view the set of all real traces starting with \( r \) and continuing with actions using only the resources belonging to \( R \).

This interpretation of \( \rho \)-traces leads in a natural way to the following approximation ordering, essentially corresponding to specification refinement

\[
(r, R) \sqsubseteq (s, S) \quad \text{iff} \quad r \leq s \quad \text{and} \quad \text{res}(r^{-1}s) \cup S \subseteq R
\]

The underlying idea is that we increase the information about a process by letting grow its real part with actions using only the specified resources \( r \leq s, \text{res}(r^{-1}s) \subseteq R \) and by possibly reducing the set of resources granted to the process for its continuation \( (S \subseteq R) \). The bottom element with respect to \( \sqsubseteq \) is \( \bot = (1, \mathcal{R}) \). It signifies that nothing has been observed and everything is still possible. If the imaginary part is the empty set, the process is called terminated, otherwise non-terminated.

Let us see how a meaningful concatenation of two \( \rho \)-traces \( (r, R) \) and \( (s, S) \) should be defined. Since the information we have about the continuation of the first process restricts to the set of resources which can still be used, we can only let execute those actions of the second process which do not use these resources. This motivates us to define for all \( R \subseteq \mathcal{R} \) and \( s \in \mathcal{G} \), the maximal real prefix of \( s \) independent of \( R \) and the corresponding resource suffix by

\[
\mu_R(s) = \bigvee \{ k \mid k \in \mathbb{M}, k \leq s, \text{res}(k) \cap R = \emptyset \},
\]

\[
\sigma_R(s) = \text{res}(\mu_R(s)^{-1}s)
\]
They define two mappings \( \mu_R : \mathcal{G} \to \mathbb{R} \) and \( \sigma_R : \mathcal{G} \to \mathcal{P}(\mathcal{R}) \). One can easily see that \( \mu_R(s) \) is in fact the restriction of \( s \) to the set of events having a past which is finite and labeled with actions not using resources from \( R \). Note that for \( R' \subseteq R \) and \( s \leq s' \) we have \( \mu_R(s) \leq \mu_{R'}(s') \) and

\[
1 = \mu_R(s) \leq \mu_R(s) \leq \mu_{\emptyset}(s) = \text{Re}(s)
\]

\[
\text{res}(s) = \sigma_R(s) \supseteq \sigma_R(s) \supseteq \sigma_{\emptyset}(s) = \emptyset.
\]

Furthermore, \( \sigma_R(s) \cup \text{resinf}(s) = \bigcap \{ \text{res}(k^{-1}s) | k \in \mathbb{M}, k \leq s, \text{res}(k) \cap R = \emptyset \} \).

Using the notations above, the following concatenation can be defined

\[
(r, R) \cdot (s, S) = (r \mu_R(s), R \cup \sigma_R(s) \cup S).
\]

When concatenating two processes, we let pass into the real component the maximal real prefix of the second process not using resources reserved by the first process for its future and adjoin the resources used by the remaining suffix to the cumulated imaginary parts.

As can be easily checked, \( \mathbb{F}_\rho \) forms a monoid together with the concatenation defined above and \( \top = (1, \emptyset) \) as the neutral element. Note furthermore that the bottom element \( \bot = (1, R) \) is a left zero with respect to concatenation, i.e. \( \bot \cdot x = \bot \) for all \( x \in \mathbb{F}_\rho \).

The concatenation in \( \mathbb{F}_\rho \) determines as usual a reflexive and transitive prefix relation \( \leq \) on \( \mathbb{F}_\rho \). Actually, one can easily check that

\[
(r, R) \leq (s, S) \iff r \leq s, R \subseteq S, \text{res}(r^{-1}s) \cap R = \emptyset.
\]

This shows in particular that the prefix relation is antisymmetric, hence it defines a partial order on \( \mathbb{F}_\rho \). Comparing the prefix ordering \( \leq \) and the approximation ordering \( \subseteq \) on \( \mathbb{F}_\rho \), we see that they are disjoint in the following sense: If \( x \leq y \) and \( x \subseteq y \), then \( x = y \). This means that a partial computation cannot be at the same time the extension and the refinement of another partial computation.

We have a chain of canonical (partial) monoid homomorphisms (note that the concatenation of two real traces is not necessarily a real trace)

\[
\mathbb{M} \hookrightarrow \mathbb{R} \hookrightarrow \mathcal{G} \overset{\varphi_\rho}{\twoheadrightarrow} \mathbb{F}_\rho
\]

where the first two mappings are inclusions and \( \varphi_\rho(g) = (\text{Re}(g), \text{resinf}(g)) \). Since the mappings are homomorphisms, they are also monotone with respect to the associated prefix orderings. The injection \( \mathbb{R} \hookrightarrow \mathbb{F}_\rho, r \mapsto (r, \text{res}(r)) \) is therefore monotone with respect to the prefix ordering \( \leq \). It maps every finite trace to a terminated \( \rho \)-trace. This canonical embedding allows us to identify \( \mathbb{R} \) with its image in \( \mathbb{F}_\rho \) and set \( \mathbb{R} \subseteq \mathbb{F}_\rho \). Likewise, we have a canonical embedding \( \mathcal{P}(\mathcal{R}) \hookrightarrow \mathbb{F}_\rho, R \mapsto (1, R) \), which allows us to set \( \mathcal{P}(\mathcal{R}) \subseteq \mathbb{F}_\rho \). Using these embeddings and the definition of the concatenation, we see that \( x = \text{Re}(x) \cdot \text{Im}(x) \) for all \( x \in \mathbb{F}_\rho \). Furthermore, we also have an injection \( \mathbb{R} \hookrightarrow \mathbb{F}_\rho, r \mapsto (r, R) \) which is monotone with respect to the approximation ordering \( \subseteq \). It maps every finite trace to a \( \rho \)-trace whose continuation is bound to no restriction.

Given an arbitrary dependence alphabet we can conveniently choose a set of resources and a representing resource mapping depending on the structure of the dependence alphabet. In the word case for instance, where we have full dependency between all actions, we recover the classical completion by restricting to a single resource on which all actions
synchronize. We then have on the second component of any ρ-trace either the empty set, corresponding thus to termination, or otherwise the chosen resource as singleton set, acting as non-termination symbol. A similar case can be made for a disjoint union of full dependencies by taking a separate resource for each full dependency. In general, we could use as resources the cliques of any covering of the dependence alphabet and assign to every action those cliques of the covering to which it belongs. This representation of dependence alphabets by resources (registers) is similar to the one used in constructing asynchronous automata recognizing regular trace languages [DR95].

5 Properties of the Approximation Ordering

The approximation order on Fρ is evidently coarser than the product order of (R, ≤) and (P(R), ⊇). Since both orders are well-founded, this implies in turn the well-foundedness of the approximation order. The next proposition shows that for ρ-traces having a common upper bound the approximation order reduces to the product order. Thus, on coherent subsets the approximation order is equivalent to the product order.

Proposition 5.1

(1) (r, R) and (s, S) have a common upper bound in Fρ iff r and s have a common upper bound in R and moreover res((r ∧ s)^{-1}s) = res(r^{-1}(r ∨ s)) ⊆ R and res((r ∧ s)^{-1}r) = res(s^{-1}(r ∨ s)) ⊆ S.

(2) If (r, R) and (s, S) have a common upper bound in Fρ, then (r, R) ⊆ (s, S) iff r ≤ s and R ⊇ S.

Proof:

(1) Let (t, T) be a common upper bound of (r, R) and (s, S). It follows from the definition that r ≤ t and s ≤ t, hence r ∨ s ≤ t. Moreover, res(r^{-1}(r ∨ s)) ⊆ res(r^{-1}t) ⊆ R and similarly, res(s^{-1}(r ∨ s)) ⊆ S. Conversely, resinf(r ∨ s) ⊆ resinf(r) ∪ res(r^{-1}(r ∨ s)) ⊆ R and similarly, resinf(r ∨ s) ⊆ S. Hence (r ∨ s, R ∩ S) is a ρ-trace which is clearly a common upper bound of (r, R) and (s, S).

(2) Clear from (1). □

The following proposition shows that Fρ is a coherently complete partial order (CCPO) and gives the computation formula for the supremum operator. It also implies the continuity of the mappings Re : (Fρ, ⊆) → (R, ≤) and Im : (Fρ, ⊆) → (P(R), ⊇). The proof uses

Lemma 5.2 Let x ∈ R and R_x be the set of real traces coherent with x. The mapping f_x : (R_x, ≤) → (R, ≤) defined by y → x^{-1}(x ∨ y) is monotone. Moreover, for all coherent sets Y ⊆ R_x, f_x(Y) is coherent and f_x(∨ Y) = ∨ f_x(Y).

Proof: From the computation of the least upper bound for real traces, it is easy to see that f_x is monotone. Hence, if Y ⊆ R_x is coherent, so is f_x(Y). Now remark that if Z ⊆ R is coherent then so is x · Z = \{x · z | z ∈ Z\} and x · ∨ Z = ∨(x · Z). Therefore, x · ∨ f_x(Y) = ∨(x · f_x(Y)) = ∨_{y \in Y} (x · x^{-1}(x ∨ y)) = ∨_{y \in Y} (x ∨ y) = x ∨ ∨ Y and, therefore, ∨ f_x(Y) = x^{-1}(x ∨ ∨ Y) = f_x(∨ Y). □
Proposition 5.3 $\mathbb{F}_\rho$ is a CCPO. In particular it is also a DCPO. The supremum of a coherent set $X \subseteq \mathbb{F}_\rho$ is

$$ \bigcup X = \left( \bigvee_{x \in X} \text{Re}(x), \bigcap_{x \in X} \text{Im}(x) \right) $$

Proof: Let $X$ be coherent in $\mathbb{F}_\rho$. Then $\text{Re}(X)$ must be coherent in $\mathbb{R}$, which is a CCPO, so there exists $r = \bigvee_{x \in X} \text{Re}(x)$. Let $R = \bigcap_{x \in X} \text{Im}(x)$.

We first check that $(r, R) \in \mathbb{F}_\rho$ and that $(r, R)$ is an upper bound of $X$. For all $x \in X$ we obtain using Lemma 5.2

$$ \text{Re}(x)^{-1}r = \text{Re}(x)^{-1}(\text{Re}(x) \lor \bigvee_{y \in X} \text{Re}(y)) = \bigvee_{y \in X} \text{Re}(x)^{-1}(\text{Re}(x) \lor \text{Re}(y)). $$

Using Proposition 5.1 (1), we deduce

$$ \text{res}(\text{Re}(x)^{-1}r) = \bigcup_{y \in X} \text{res}(\text{Re}(x)^{-1}(\text{Re}(x) \lor \text{Re}(y))) \subseteq \text{Im}(x). $$

Therefore, $\text{resinf}(r) \subseteq \text{resinf}(\text{Re}(x)) \cup \text{res}(\text{Re}(x)^{-1}r) \subseteq \text{Im}(x)$, so $\text{resinf}(r) \subseteq \bigcap_{x \in X} \text{Im}(x) = R$ which shows that $(r, R) \in \mathbb{F}_\rho$. Moreover, $(r, R)$ is an upper bound to all $x \in X$ since $\text{res}(\text{Re}(x)^{-1}r) \subseteq \text{Im}(x)$ and evidently $\text{Re}(x) \leq r$ and $\text{Im}(x) \geq R$.

We now show that $(r, R)$ is the minimal upper bound of $X$. Let $z$ be any upper bound of $X$. For all $x \in X$ we then have $\text{Re}(x) \leq \text{Re}(z)$ and, therefore, $r = \bigvee \text{Re}(X) \leq \text{Re}(z)$. Furthermore, $\text{Im}(x) \supseteq \text{Im}(z) \cup \text{res}(\text{Re}(x)^{-1} \text{Re}(z)) \supseteq \text{Im}(z) \cup \text{res}(r^{-1} \text{Re}(z))$, hence $R = \bigcap_{x \in X} \text{Im}(x) \supseteq \text{Im}(z) \cup \text{res}(r^{-1} \text{Re}(z))$. This proves $(r, R) \subseteq z$. \hfill $\square$

Analogously, we can prove the following proposition concerning the infimum operator, which is defined on all non-empty subsets. Note the substantial simplification of the formula for the greatest lower bound in the presence of coherence due to

Lemma 5.4 Let $x \in \mathbb{R}$. The mapping $g_x : (\mathbb{R}, \leq) \rightarrow (\mathcal{P}(\Sigma), \supseteq)$ defined by $y \mapsto \text{alph}(y \land x)^{-1}x$ is monotone. Moreover, for all non-empty subsets $Y \subseteq \mathbb{R}$ we have $g_x(\bigwedge Y) = \bigcup_{y \in Y} g_x(y)$.

Proof: If $y \leq z$ then $y \land x \leq z \land x \leq x$. Hence $(z \land x)^{-1}x$ is a suffix of $(y \land x)^{-1}x$ and $g_x(y) \supseteq g_x(z)$ which shows that $g_x$ is monotone.

Let $Y \subseteq \mathbb{R}$ be non-empty. For all $y \in Y$, we have $\bigwedge Y \leq y$. Since $g_x$ is monotone it follows that $g_x(\bigwedge Y) \supseteq \bigcup_{y \in Y} g_x(y)$. Conversely, note that $a \in g_x(y)$ if and only if there exists a prime real trace $z$ such that $z \leq x$, $z \not\leq y$ and $\text{max}(z) = \{a\}$. Hence, if $a \in g_x(\bigwedge Y)$, let $z$ be as above. Since $z \not\leq \bigwedge Y$, there exists $y \in Y$ such that $z \not\leq y$ and it follows that $a \in g_x(y) \subseteq \bigcup_{y \in Y} g_x(y)$. \hfill $\square$

Proposition 5.5 The infimum of a non-empty set $X \subseteq \mathbb{F}_\rho$ is

$$ \bigcap X = (r, R) \text{ where } r = \bigwedge_{x \in X} \text{Re}(x) \text{ and } R = \bigcup_{x \in X} (\text{Im}(x) \cup \text{res}(r^{-1} \text{Re}(x))) $$

In particular, if $X$ is coherent, then

$$ \bigcap X = (\bigwedge \text{Re}(X), \bigcup \text{Im}(X)) $$
Proof: Since $X \neq \emptyset$, there exists $r = \bigwedge \Re(X)$. Let $R = \bigcup_{x \in X} (\Im(x) \cup \res(r^{-1} \Re(x)))$. Choosing some $x \in X$, we have $r \leq \Re(x)$, hence $\res(r) \subseteq \res(\Re(x)) \subseteq \Im(x) \subseteq R$ and, therefore, $(r, R) \in \mathbb{F}_\rho$. Moreover, $(r, R)$ is clearly a lower bound of $X$.

Let now $(s, S)$ be any lower bound of $X$. Then $s \leq \Re(x)$ for all $x \in X$, hence $s \leq r$. Furthermore, for all $x \in X$, $S \supseteq \Im(x) \cup \res(s^{-1} \Re(x)) = \Im(x) \cup \res(s^{-1}r) \cup \res(r^{-1} \Re(x))$. Therefore $S \supseteq R \cup \res(s^{-1}r)$. We conclude that $(s, S) \subseteq (r, R)$. This shows that $(r, R)$ is the greatest lower bound of $X$.

If $X$ is coherent, then we see that for all $x \in X$ we have

\[
\res(r^{-1} \Re(x)) = \res((r \wedge \Re(x))^{-1} \Re(x)) = \bigcup_{y \in X} \res((\Re(y) \wedge \Re(x))^{-1} \Re(x)) \quad \text{by Lemma 5.4}
\]

\[
\subseteq \bigcup_{y \in X} \Im(y) \quad \text{by Prop. 5.1.(1)}.
\]

Therefore, $\bigcup_{x \in X} \res(r^{-1} \Re(x)) \subseteq \bigcup_{x \in X} \Im(x)$, which finally implies $R = \bigcup_{x \in X} (\Im(x) \cup \res(r^{-1} \Re(x))) = \bigcup_{x \in X} \Im(x)$. \hfill \Box

Proposition 5.6 ($\mathbb{F}_\rho, \sqsubseteq$) is bounded distributive.

Proof: By Propositions 5.3 and 5.5 the supremum and infimum of bounded sets can be computed componentwise. Using the bounded distributivity in $\Re$ and $\mathcal{P}(\Re)$ we can then easily check the bounded distributivity in $\mathbb{F}_\rho$. \hfill \Box

As a consequence of Propositions 5.3, 5.6, and the well-foundedness of ($\mathbb{F}_\rho, \sqsubseteq$) we are now able to state, due to Proposition 2.2, that ($\mathbb{F}_\rho, \sqsubseteq$) is a p-algebraic CCPO, thus assuring its adequacy as a denotational domain for modeling processes term algebras.

Theorem 5.7 ($\mathbb{F}_\rho, \sqsubseteq$) is a p-algebraic CCPO. In particular it is also a Scott domain.

The next theorem identifies the compact elements in ($\mathbb{F}_\rho, \sqsubseteq$). They are in fact exactly those having a compact and thus finite real part.

Theorem 5.8 An element $x \in \mathbb{F}_\rho$ is compact iff $\Re(x)$ is compact (finite).

Proof: Let $x$ be a compact element of $\mathbb{F}_\rho$ and consider the set $Y = \{ (k, \res(k^{-1} \Re(x)) \cup \Im(x)) \mid k \in M, k \leq \Re(x) \}$. We will first prove that $Y$ is directed and $\bigcup Y = x$. Recall that $\bigcap \{ \res(k^{-1} \Re(x)) \mid k \in M, k \leq \Re(x) \} = \res(\Re(x)) \subseteq \Im(x)$. Since $\Re$ is k-algebraic and the compacts are precisely the finite traces, we have $\Re(x) = \bigvee \{ k \in M \mid k \leq \Re(x) \}$. It follows from Proposition 5.3 that $\bigcup Y = x$. Furthermore, since for any $k \leq k' \leq \Re(x)$ we have $(k, \res(k^{-1} \Re(x)) \cup \Im(x)) \subseteq (k', \res(k'^{-1} \Re(x)) \cup \Im(x))$ we deduce that $Y$ is directed. Now, since $x$ is compact and $x = \bigcup Y$, we have $x \in Y$ which implies that $\Re(x) \in M$ is compact.

Suppose now, conversely, that $\Re(x) \in \Re$ is compact and $x \subseteq \bigcup Y$ for some directed set $Y \subseteq \mathbb{F}_\rho$. Then $\Re(x) \leq \bigvee \Re(Y)$ and $\Im(x) \supseteq \bigcap \Im(Y)$, whereby both suprema are directed. We can therefore extract from $Y$ two elements $y_1$ and $y_2$, such that $\Re(x) \leq \Re(y_1)$ and $\Im(x) \supseteq \Im(y_2)$. Since $Y$ is directed, there exists moreover an upper bound $y \in Y$ to $y_1$ and $y_2$. Consequently, we have $\Re(x) \leq \Re(y_1) \leq \Re(y)$ and $\Im(x) \supseteq \Im(y_2) \supseteq \Im(y)$.
Im(y). As \( \bigcup Y \) is a common upper bound of \( x \) and \( y \), it follows by Proposition 5.1.(2) that \( x \subseteq y \in Y \) is compact.

Let us now identify the form of the primes in \( (\mathbb{F}_\rho, \subseteq) \). First, one can easily see that the primes in \((\mathcal{P}(\mathcal{R}), \supseteq)\) are the sets missing a single element. Secondly, the primes in \((\mathbb{R}, \leq)\) are so-called pyramids (see [GR93]), that is traces having a single maximal event. Now, the primes of the direct product of any two domains are exactly those pairs having a prime on the first component and the bottom element on the second component or vice versa. Recall that the bottom element of \((\mathbb{I}, \leq)\) is 1 and the bottom element of \((\mathcal{P}(\mathcal{R}), \supseteq)\) is \( \mathcal{R} \). As the order structure of \((\mathbb{F}_\rho, \subseteq)\) is finer than the direct product order of \((\mathbb{R}, \leq)\) and \((\mathcal{P}(\mathcal{R}), \supseteq)\), we expect to have additional primes corresponding to the constraint imposed on actions increasing the real part to use only those resources specified by the imaginary part. We, therefore, obtain three forms of primes, as stated in the following

**Theorem 5.9** An element \((p, R) \in \mathbb{F}_\rho\) is prime in the following three cases

1. \( p \) is prime (i.e. \(|\max(p)| = 1\)) and \( R = \mathcal{R} \)
2. \( p = 1 \) and \( R \) is prime (i.e. \(|\bar{R}| = 1\))
3. \( p \) and \( R \) are both prime and \( \text{res}(\max(p)) \cup R = \mathcal{R} \).

**Proof:** We first show that any prime \((p, R) \in \mathbb{F}_\rho\) has one of the three forms. Let \((p, R)\) be prime, hence necessarily compact and irreducible. Since \((1, \mathcal{R}) = \bigcup \emptyset\) is not irreducible, we must have \(|\max(p)| \geq 1\) or \(|\bar{R}| \geq 1\).

Suppose first that \(|\bar{R}| \geq 2\). Then, choosing \(\alpha, \beta \in \bar{R}, \alpha \neq \beta\) we would have \((p, R) = (p, R \cup \{\alpha\}) \cup (p, R \cup \{\beta\})\). On the other hand, neither \((p, R) = (p, R \cup \{\alpha\})\) nor \((p, R) = (p, R \cup \{\beta\})\) satisfies \(|\bar{R}| \leq 1\).

Suppose now that \(|\max(p)| \geq 2\). Then \(p = qab = qba\) for some \(a, b \in \Sigma\) with \((a, b) \in I\). Consequently, \(\text{res}(a) \cap \text{res}(b) = \emptyset\) and \((p, R) = (qab, R) = (qb, R \cup \text{res}(a)) \cup (qa, R \cup \text{res}(b))\), but again neither \((p, R) = (qb, R \cup \text{res}(a))\) nor \((p, R) = (qa, R \cup \text{res}(b))\). This implies \(|\max(p)| \leq 1\).

Putting these observations together we obtain for \((p, R)\) either one of the first two cases stated or otherwise \(|\max(p)| = 1\) and \(|\bar{R}| = 1\). Let then \(\max(p) = \{a\}\) and \(\bar{R} = \{\alpha\}\). Assuming \(\alpha \notin \text{res}(a)\) we would have \((p, R) = (qa, R) = (q, R) \cup (qa, \mathcal{R})\), but once again neither \((p, R) = (q, R)\) nor \((p, R) = (qa, \mathcal{R})\). We conclude that \(\bar{R} = \{\alpha\} \subseteq \text{res}(a) = \text{res}(\max(p))\), which shows that \((p, A)\) satisfies the third case stated.

Let us now show that, conversely, any \((p, R) \in \mathbb{F}_\rho\) of one of the three forms is prime. Let \(X \subseteq \mathbb{F}_\rho\) be such that \((p, R) \subseteq \bigcup_{x \in X} (p_x, R_x) = (\bigvee_{x \in X} p_x, \bigcap_{x \in X} R_x)\). Then \(p \leq \bigvee_{x \in X} p_x\) and \(R \supseteq \bigcap_{x \in X} R_x\).

In the first case \(R = \mathcal{R}\) and \(p \in \mathcal{R}\) is prime, hence there exists \(x \in X\) such that \(p \leq p_x\) and therefore \((p, R) = (p, \mathcal{R}) \subseteq (p_x, R_x)\).

Similarly, in the second case \(p = 1\) and \(R \subseteq \mathcal{R}\) is prime, hence there exists \(x \in X\) such that \(R \supseteq R_x\). Since \(p = 1 \leq p_x\), we have by Proposition 5.1 (2) that \((p, R) \subseteq (p_x, R_x)\).

Finally, in the third case \(p \in \mathcal{R}\) and \(R \subseteq \mathcal{R}\) are both prime so \(\max(p) = \{a\}, \bar{R} = \{\alpha\}\) and, furthermore, \(\bar{R} \subseteq \text{res}(\max(p))\) so \(\alpha \in \text{res}(a)\). In particular, since \(R\) is prime, there exists \(x \in X\) such that \(R \supseteq R_x\). As \((p, R)\) and \((p_x, R_x)\) are coherent, we have by Proposition 5.1 (1) \(R \supseteq R_x \supseteq \text{res}(p_x^{-1}(p_x \vee p))\). Hence, \(\alpha \notin \text{res}(p_x^{-1}(p_x \vee p))\) and, therefore,
Proof: Let it indeed has this property with respect to the approximation ordering \( \preceq \). Since \( \max(p_x^{-1}(p_x \vee p)) \subseteq \max(p) = \{a\} \) we must have \( \max(p_x^{-1}(p_x \vee p)) = \emptyset \). But then \( p_x^{-1}(p_x \vee p) = 1 \) which implies \( p_x = p_x \vee p \) and finally \( p \leq p_x \). Together with \( R \supseteq R_x \) we again conclude using Proposition 5.11(2) that \( (p, R) \subseteq (p_x, R_x) \).

As expected from the word case, the concatenation operation is not monotone with respect to prefix ordering \( \preceq \) on \( \mathbb{F}_\rho \). On the other hand, as stated in the next proposition, it indeed has this property with respect to the approximation ordering \( \sqsubseteq \).

**Proposition 5.10** The concatenation in \( \mathbb{F}_\rho \) is monotone with respect to the approximation order, i.e. \( x \subseteq x' \) and \( y \subseteq y' \) implies \( x \cdot y \subseteq x' \cdot y' \) for all \( x, x', y, y' \in \mathbb{F}_\rho \).

**Proof:** Let \( x = (r, R) \), \( y = (s, S) \), \( x' = (r', R') \), \( y' = (s', S') \) be such that \( x \subseteq x' \) and \( y \subseteq y' \). Then \( r \leq r' \), \( R \supseteq \res(r^{-1}r') \cup R' \) and \( s \leq s' \), \( S \supseteq \res(s^{-1}s') \cup S' \). By definition, we have \( xy = (r \mu_R(s), R \cup \sigma_R(s) \cup S) \) and \( x'y' = (r' \mu_R(s'), R' \cup \sigma_R(s') \cup S') \). We have to prove the two inequalities

\[
\begin{align*}
r \mu_R(s) \leq & \ r' \mu_R(s') \\
R \cup \sigma_R(s) \cup S \supseteq & \ \res((r \mu_R(s))^{-1}(r' \mu_R(s'))) \cup R' \cup \sigma_R(s') \cup S'.
\end{align*}
\]

Since \( \res(r^{-1}r') \subseteq R \) and \( \res(\mu_R(s)) \cap R = \emptyset \) we have \( (r^{-1}r') \cdot \mu_R(s) \cdot (r^{-1}r') \). Furthermore, since \( R \supseteq R' \) and \( s \leq s' \) we have \( \mu_R(s) \leq \mu_R(s') \).

To check the first inequality, we use

\[
\begin{align*}
r' \cdot \mu_R(s') = & \ r \cdot (r^{-1}r') \cdot \mu_R(s) \cdot (\mu_R(s)^{-1} \mu_R(s')) \\
= & \ r \cdot \mu_R(s) \cdot (r^{-1}r') \cdot (\mu_R(s)^{-1} \mu_R(s')).
\end{align*}
\]

To prove the second inequality we proceed analogously. We have \( \mu_R(s) \leq \mu_R(s') \leq s' \) and \( \mu_R(s) \leq s \leq s' \) hence

\[
\mu_R(s)^{-1} s' = (\mu_R(s)^{-1}(\mu_R(s'))) (\mu_R(s')^{-1} s') = (\mu_R(s)^{-1} s)'(s^{-1}s').
\]

Therefore,

\[
\begin{align*}
\res((r \mu_R(s))^{-1}(r' \mu_R(s'))) \cup \sigma_R(s') = & \ \res(r^{-1}r') \cup \res(\mu_R(s)^{-1} \mu_R(s')) \cup \res(\mu_R(s')^{-1} s') \\
= & \ \res(r^{-1}r') \cup \res(\mu_R(s)^{-1} s) \cup \res(s^{-1}s') \\
\subseteq & \ R \cup \sigma_R(s) \cup S
\end{align*}
\]

which together with \( R' \subseteq R \) and \( S' \subseteq S \) shows the validity of the second condition.

The next theorem states that the concatenation operation is continuous with respect to the approximation order. Together with the Scott-domain structure defined by the approximation ordering, this gives us the main tool necessary for devising compositional denotational semantics using resource traces and their concatenation operation. The proof of the theorem uses the following

**Proposition 5.11** Let \( x, y \in \mathbb{F}_\rho \) be \( \rho \)-traces. Then,

\[
\Kmp(x \cdot y) = \downarrow(\Kmp(x) \cdot \Kmp(y)).
\]

15
Proof: From Proposition 5.10 it is clear that \( \downarrow (\text{Kmp}(x) \cdot \text{Kmp}(y)) \subseteq \text{Kmp}(x \cdot y) \). Conversely, let \( x = (r,R), y = (s,S) \) be \( \rho \)-traces and consider any \( z = (t,T) \in \text{Kmp}(x \cdot y) \). Let \( u = r \land t \). Then \( x' = (u,R \cup \text{res}(u^{-1}r)) \in \text{Kmp}(x) \). Let furthermore \( v = u^{-1}t \). Then \( v \leq \mu_R(s) \leq s \), hence, \( y' = (v,S \cup \text{res}(v^{-1}s)) \in \text{Kmp}(y) \). Now, \( \text{res}(v) \cap (\text{res}(u^{-1}r) \cup R) = \emptyset \), consequently

\[
x' \cdot y' = (uv, R \cup S \cup \text{res}(u^{-1}r) \cup \text{res}(v^{-1}s)) = (t, R \cup S \cup \text{res}(u^{-1}r) \cup \text{res}(v^{-1} \mu_R(s)) \cup \sigma_R(s)) = (t, R \cup S \cup \sigma_R(s)) \cup \text{res}(t^{-1}r \mu_R(s))
\]

Since \( z \subseteq x \cdot y \) we have \( T \supseteq R \cup S \cup \sigma_R(s) \cup \text{res}(t^{-1}r \mu_R(s)) \). Therefore, \( z \subseteq x' \cdot y' \in \text{Kmp}(x) \cdot \text{Kmp}(y) \).

\[\square\]

**Theorem 5.12** The concatenation in \( \mathbb{F}_\rho \) is continuous with respect to the approximation order, i.e. if \( X, Y \subseteq \mathbb{F}_\rho \) are directed, then \( X \cdot Y \) is directed and \( \bigcup (X \cdot Y) = (\bigcup X) \cdot (\bigcup Y) \).

**Proof:** Let \( X \) and \( Y \) be directed in \( \mathbb{F}_\rho \). Using Proposition 5.10 one can easily check that \( X \cdot Y \) is directed. Moreover, since for all \( x \in X \) and \( y \in Y \) we have \( x \subseteq \bigcup X \) and \( y \subseteq \bigcup Y \), again by Proposition 5.10 we see that \( x \cdot y \subseteq (\bigcup X) \cdot (\bigcup Y) \). Therefore, \( \bigcup (X \cdot Y) \subseteq (\bigcup X) \cdot (\bigcup Y) \).

Let us now show the opposite direction. Since \( X \) and \( Y \) are directed and the concatenation is monotone we have \( \text{Kmp}(\bigcup X) \cdot \text{Kmp}(\bigcup Y) \subseteq \downarrow (X \cdot Y) \). Using Proposition 5.11 we deduce \( \downarrow X \cdot \downarrow Y = \downarrow \text{Kmp}(\bigcup X) \cdot \text{Kmp}(\bigcup Y) = \downarrow (\text{Kmp}(\bigcup X) \cdot \text{Kmp}(\bigcup Y)) \subseteq \downarrow (X \cdot Y) \).

The last theorem enables us to specify a compositional denotational semantics for recursively defined processes using resource traces as denotational domain. The denotational mechanism thereby used in order to define the semantics of recursive process terms slightly differs from the one sketched in the introduction. We illustrate below the main additional idea leading to the right denotational approach, for an extensive presentation of the denotational and the matching operational semantics the reader is referred to [GM99].

Consider for instance the process \( p \) defined by the recursive term \( \text{rec}.x.(a \cdot x) \). The denotational semantics of \( p \) should be some resource trace which is a fixed point of the continuous mapping \( \mathbb{F}_\rho \rightarrow \mathbb{F}_\rho, x \mapsto (a,\emptyset) \cdot x \).

A first idea would be to choose the classical least fixed point interpretation. Starting with the bottom element \( (1,R) \) and iterating the mapping we would obtain \( (a^\omega,R) \) as least fixed point semantics of \( p \). However, by claiming all resources the process \( p \) would prevent any other action from executing, even those which use no resource needed by \( a \). This clearly is not a desirable semantics for it unnecessarily sequentializes actions which otherwise could be allowed to execute concurrently.

A more appropriate semantics, and in fact the very one we aim at by introducing the model of resource traces, is to explicitly take into account the resources used by the process \( p \). A simple syntactical analysis of the defining term reveals that the only resources to be claimed by \( p \) should be those of \( a \). Therefore, \( (1,\text{res}(a)) \) should be an approximation of the semantics of \( p \). Starting from this new lower bound and iterating the mapping now leads to the fixed point \( (a^\omega,\text{res}(a)) \), which indeed captures our intended semantics since no resources other than those claimed by \( a \) will be blocked by the process \( p \).
6 Topology

In this section, we introduce a very natural ultra-metric $d$ on $\rho$-traces. We show that the topology induced by this ultra-metric is precisely the Lawson topology \cite{Law88} induced by the approximation order. Using this fact and the properties of the approximation ordering, we deduce that the metric space $(\mathbb{F}_\rho, d)$ is compact, hence complete. Moreover, the set of finite $\rho$-traces is countable, discrete and dense in $(\mathbb{F}_\rho, d)$. Finally we prove that the concatenation is uniformly continuous with respect to the defined ultra-metric.

We start by recalling some well-known topological definitions. Let $(X, \mathcal{O})$ be a topological space. A family $\mathcal{B} \subseteq \mathcal{O}$ is called a basis of the topology iff any open set is an arbitrary union of sets in $\mathcal{B}$. A family $\mathcal{S} \subseteq \mathcal{O}$ is called a subbasis of the topology iff the family of finite intersections of sets in $\mathcal{S}$ is a basis of the topology.

A function $d : X \times X \to \mathbb{R}$ is called an ultrametric on $X$ iff for any $x, y, z \in X$ it is non-degenerate, i.e. $d(x, y) = 0 \iff x = y$, symmetric, i.e. $d(x, y) = d(y, x)$, and satisfies the ultrametric inequality, i.e. $d(x, z) \leq \max(d(x, y), d(y, z))$.

Finally, let $\text{Kmp}_n = \{ k \in \text{Kmp} \mid |k| \leq n \}$ and $\text{Kmp}_n(x) = \text{Kmp}_n \cap \text{Kmp}(x)$ for any $n \in \mathbb{N}$, $x \in \mathbb{F}_\rho$.

**Definition 6.1** Let $x, y \in \mathbb{F}_\rho$. The distance between $x$ and $y$ is defined by $d(x, y) = 2^{-l(x, y)}$ where

$$l(x, y) = \inf \{ n \in \mathbb{N} \mid \text{Kmp}_n(x) \neq \text{Kmp}_n(y) \} \in \mathbb{N} \cup \{\infty\}$$

and by convention, $\inf(\emptyset) = \infty$ and $2^{-\infty} = 0$.

Let $x \in \mathbb{F}_\rho$ and $n \in \mathbb{N}$. The $n$th open ball centred in $x$ is

$$B(x, n) = \{ y \in \mathbb{F}_\rho \mid l(x, y) > n \}.$$ 

For all $x \in \mathbb{F}_\rho$ we have by convention $d(x, x) = 0$. Conversely, if $x, y \in \mathbb{F}_\rho$ are such that $d(x, y) = 0$ then $\text{Kmp}_n(x) = \text{Kmp}_n(y)$ for all $n \in \mathbb{N}$, hence, $\text{Kmp}(x) = \bigcup_{n \in \mathbb{N}} \text{Kmp}_n(x) = \bigcup_{n \in \mathbb{N}} \text{Kmp}_n(y) = \text{Kmp}(y)$, and using the algebraicity of $(\mathbb{F}_\rho, \subseteq)$ it follows that $x = \bigcup \text{Kmp}(x) = \bigcup \text{Kmp}(y) = y$. The distance function is symmetric by definition and, as can be easily checked, satisfies the ultrametric inequality. Therefore, $d$ is an ultra-metric on $\mathbb{F}_\rho$. Moreover, the family of open balls

$$\mathcal{B} = \{ B(x, n) \mid x \in \mathbb{F} \text{ and } n \in \mathbb{N} \}$$

is a basis of the topology induced by the metric $d$ on $\mathbb{F}_\rho$.

The Lawson topology can be defined in a quite general setting, namely for continuous complete semi-lattices (see \cite{GHK80} for a thorough treatment of the subject). Here we are mainly interested in the properties of the Lawson topology for Scott domains $(X, \leq)$. In this particular case the family of sets

$$\mathcal{S} = \{ \uparrow k \mid k \text{ is compact in } (X, \leq) \} \cup \{ \overline{\uparrow k} \mid k \text{ is compact in } (X, \leq) \}$$

can be shown to build a subbasis of the Lawson topology on $(X, \leq)$. We use this fact in order to prove the equivalence between the $d$-topology and the $\subseteq$-Lawson topology on $\mathbb{F}_\rho$.

**Proposition 6.2** The topology on $\mathbb{F}_\rho$ induced by the ultra-metric $d$ coincides with the Lawson topology on $\mathbb{F}_\rho$ induced by the approximation order $\subseteq$. 

17
**Proof:** Consider any compact element \( k \in \mathbf{F}_\rho \). For all \( x, y \in \mathbf{F}_\rho \) such that \( l(x, y) > |k| \) we have \( k \subseteq x \iff k \subseteq y \) or stated equivalently \( x \in \uparrow k \iff y \in \uparrow k \). Therefore, if \( x \in \uparrow k \) then \( B(x, |k|) \subseteq \uparrow k \) and if \( x \in \uparrow k \) then \( B(x, |k|) \subseteq \uparrow k \). This shows that \( \uparrow k \) and \( \uparrow k \) are both \( d \)-open (and \( d \)-closed). Thus, the \( d \)-topology is finer than the \( \sqsubset \)-Lawson topology.

We now show that, conversely, for any \( x \in \mathbf{F}_\rho \), \( n \in \mathbf{N} \) the open ball \( B(x, n) \) is Lawson-open. For \( k \in \text{Kmp} \), let \( O_x(k) = \uparrow k \) if \( k \subseteq x \) and \( O_x(k) = \uparrow k \) otherwise. Note that in either case \( O_x(k) \) belongs to the subbasis of the Lawson topology. Now, for any \( y \in \mathbf{F}_\rho \) we have \( y \in B(x, n) \) iff \( l(x, y) > n \) iff \( \text{Kmp}_n(x) = \text{Kmp}_n(y) \). It is easy to check that the later statement is equivalent to \( y \in O_x(k) \) for all \( k \in \text{Kmp}_n \). Therefore, \( B(x, n) = \bigcap_{k \in \text{Kmp}_n} O_x(k) \). Since \( \text{Kmp}_n \) is a finite set, \( B(x, n) \) is a finite intersection of sets belonging to the subbasis of the Lawson topology, hence Lawson-open itself. Therefore, the \( \sqsubset \)-Lawson topology is finer than the \( d \)-topology, thus concluding the equivalence proof between the two topologies.

From Proposition 6.2, we infer the following property which is a general fact for the Lawson topology (see [GHK+80]). Essentially it states that \( \sqsubset \)-convergence is stronger than \( d \)-convergence and that the two limit notions, when defined in both settings, coincide.

**Proposition 6.3** Let \((x_n)_{n \geq 0} \subseteq \mathbf{F}_\rho \) be an \( \sqsubset \)-increasing sequence. Then, the sequence \((x_n)_{n \geq 0} \) is \( d \)-convergent and furthermore

\[
\lim_{n \to \infty} x_n = \bigcup_{n \geq 0} x_n
\]

For any partial order \((X, \leq)\) which is both bounded complete and directed complete the Lawson topology is quasi-compact and \( T_1 \). If \((X, \leq)\) is in addition \( k \)-algebraic, then the Lawson topology becomes \( T_2 \) (Hausdorff), whence compact, and the set of compact elements of \((X, \leq)\) is dense in \( X \). For all these statements, the reader is again referred to [GHK+80] where they are proved for the more general case of continuous complete semi-lattices. Using these properties we now give the most important corollary of Proposition 6.2.

**Theorem 6.4** The metric space \((\mathbf{F}_\rho, d)\) is compact (hence complete). Furthermore, the set of finite \( \rho \)-traces is countable, dense, discrete and open in \((\mathbf{F}_\rho, d)\).

**Proof:** By Theorem 5.7, the partial order \((\mathbf{F}_\rho, \sqsubseteq)\) is a bounded complete domain. Since the \( d \)-topology is the \( \sqsubset \)-Lawson topology, it follows that \((\mathbf{F}_\rho, d)\) is a compact (hence complete) metric space. Moreover, since the compact \( \rho \)-traces are by Theorem 5.8 precisely the finite ones, it follows that the countable set of finite \( \rho \)-traces is dense in \((\mathbf{F}_\rho, d)\). There remains for us to show that the set of finite \( \rho \)-traces is discrete and open (which is no general fact for the Lawson topology of bounded complete domains).

To this end we prove that \( B(k, |k| + 1) = \{k\} \) for any finite \( \rho \)-trace \( k \). Consider, therefore, any \( x \in B(k, |k| + 1) \). Then \( l(k, x) > |k| + 1 \), hence, for any finite \( \rho \)-trace \( h \) with \( |h| \leq |k| + 1 \) we have \( h \subseteq k \iff h \subseteq x \). In particular, taking \( h = k \), it follows that \( k \subseteq x \), hence, \( \text{Re}(k) \leq \text{Re}(x) \) and \( |k| \leq |x| \).

Assume that \( \text{Re}(k) < \text{Re}(x) \). Then we can find a finite real trace \( l \) such that \( \text{Re}(k) < l \leq \text{Re}(x) \) with \( |l| = |k| + 1 \). Taking now \( h = l \cap x \) we have by Proposition 5.5 \( \text{Re}(h) = \).
l \land \text{Re}(x) = l$, whence $|h| = |l| = |k| + 1$. Since furthermore $h \subseteq x$, we infer $h \subseteq k$ which contradicts $\text{Re}(k) < l = \text{Re}(h)$.

Therefore, $\text{Re}(k) = \text{Re}(x)$ and $|x| = |k|$. Taking $h = x$, we then deduce that $x \subseteq k$, which together with $k \subseteq x$ allows us to conclude $x = k$.  

We have seen in the previous section (Proposition 5.12) that the concatenation of $\rho$-traces is continuous with respect to the approximation order. We now prove the following stronger result.

**Proposition 6.5** The concatenation on $\rho$-traces is uniformly $d$-continuous. Moreover, for any $x, x', y, y' \in \mathbb{F}_\rho$

$$d(xy, x'y') \leq \max(d(x, x'), d(y, y')).$$

**Proof:** Let $x, x', y, y' \in \mathbb{F}_\rho$. We have to show that for all $n \in \mathbb{N}$ we have $l(x, x') > n$ and $l(y, y') > n \Rightarrow l(xy, x'y') > n$ or, stated equivalently, $\text{Kmp}_n(x) = \text{Kmp}_n(x')$ and $\text{Kmp}_n(y) = \text{Kmp}_n(y') \Rightarrow \text{Kmp}_n(xy) = \text{Kmp}_n(x'y')$. The later statement is a direct consequence of the following claim: for any $x, y \in \mathbb{F}_\rho$, $n \in \mathbb{N}$

$$\text{Kmp}_n(xy) = \text{Kmp}_n \cap \downarrow (\text{Kmp}_n(x) \cdot \text{Kmp}_n(y)).$$

Note that this claim strengthens Proposition 5.11 and has exactly the same proof. For the non trivial inclusion, we only have to note that in the proof of Proposition 5.11 we have $|x'| \leq |z|$ and $|y'| \leq |z|$.  

Note that Proposition 5.12 is in fact a corollary of Proposition 6.5. Indeed, a Lawson-continuous mapping between two domains is necessarily $\sqsubseteq$-continuous. However, the converse is false in general. Hence, we cannot infer the uniform continuity of the concatenation from its $\sqsubseteq$-continuity and the compactness of $(\mathbb{F}_\rho, d)$.

**7. The Calculus of $\alpha$- and $\delta$-Traces**

The calculus of $\alpha$- and $\delta$-traces has been previously introduced in [Die93a, DG95] as a means to cope with the problem of non-continuity of trace concatenation. We discuss in short some of their strong and weak points, thus providing an a posteriori motivation for the model we have put forward in this paper.

Let the alphabet at infinity of a trace $x \in \mathfrak{G}$ be defined by

$$\text{alphinf}(x) = \bigcap \{ \text{alph}(k^{-1}x) \mid k \in \mathbb{M}, k \leq x \}$$

It determines a mapping $\mathfrak{G} \rightarrow \mathcal{P}(\Sigma)$, $x \mapsto \text{alphinf}(x)$. For real traces, the alphabet at infinity represents the set of actions occurring infinitely many times as event labels in the trace. For non-real traces, it additionally contains the alphabet of the non-real part, hence $\text{alphinf}(x) = \text{alphinf}(\text{Re}(x)) \cup \text{alph}(\text{Re}(x)^{-1}x)$.

**Definition 7.1** An $\alpha$-trace (or $\delta$-trace respectively) over the dependence alphabet $(\Sigma, D)$ is a pair $x = (r, A)$ $(x = (r, D(A))$ respectively), where $r \in \mathbb{R}(\Sigma, D)$ is a real trace, and $A \subseteq \Sigma$ is a subset of actions such that $\text{alphinf}(r) \subseteq A$. The trace $r$ is denoted by $\text{Re}(x)$ and called the real part of $x$. The set $A$ ($D(A)$ respectively) is denoted by $\text{Im}(x)$ and called the imaginary part of $x$. The set of $\alpha$-traces ($\delta$-traces respectively) over $(\Sigma, D)$ is denoted by $\mathbb{F}_\alpha$ ($\mathbb{F}_\delta$ respectively).
The informal semantics of $\alpha$-traces is much the same as for $\rho$-traces, except that the restriction imposed by the second component on the continuation of the process applies directly to the actions that can still be performed rather than to the resources that can still be used by the process. The interpretation is similar for $\delta$-traces.

Specification refinement leads as expected to the following natural approximation orderings for $\mathcal{F}_\alpha$ and $\mathcal{F}_\delta$

For $\alpha$-traces:
\[(r, A) \sqsubseteq (s, B) \text{ if and only if } r \leq s, \alpha(r^{-1}s) \cup B \subseteq A\]

For $\delta$-traces:
\[(r, D(A)) \sqsubseteq (s, D(B)) \text{ if and only if } r \leq s, D(\alpha(r^{-1}s) \cup B) \subseteq D(A)\]

Furthermore, $\mathcal{F}_\alpha$ and $\mathcal{F}_\delta$ are monoids with $(1, \emptyset)$ as neutral element and the following concatenations

For $\alpha$-traces:
\[(r, A) \cdot (s, B) = (r \mu_A(s), A \cup \sigma_A(s) \cup B)\]

For $\delta$-traces:
\[(r, D(A)) \cdot (s, D(B)) = (r \mu_A(s), D(A \cup \sigma_A(s) \cup B))\]

where
\[
\mu_A(s) = \bigvee\{ k \mid k \in \mathbb{M}, k \leq s, \alpha(k) \cap D(A) = \emptyset \},
\]
\[
\sigma_A(s) = \alpha(\mu_A(s)^{-1}s)
\]

are the maximal real prefix of $s$ independent of $A$ and the corresponding alphabetic suffix. Note that the concatenation is well-defined in the $\delta$-case since $\mu_A(s)$ and $\sigma_A(s)$ depend on $D(A)$ only, which is actually the main reason for introducing the $\delta$-model. The canonical mapping $\varphi_{\alpha\delta} : \mathcal{F}_\alpha \to \mathcal{F}_\delta$, $\varphi_{\alpha\delta}(r, A) = (r, D(A))$ is at the same time a monoid homomorphism and a projection between $\mathcal{F}_\alpha$ and $\mathcal{F}_\delta$. The associated embedding is given by $\psi_{\delta\alpha} : \mathcal{F}_\delta \to \mathcal{F}_\alpha$, $\psi_{\delta\alpha}(r, D(A)) = (r, 1(I(A)))$. Therefore, $\mathcal{F}_\delta$ can be regarded as being a more abstract view of $\mathcal{F}_\alpha$, in both algebraical and order-theoretical senses.

As shown in [DG95], the approximation orders $(\mathcal{F}_\alpha, \sqsubseteq)$ and $(\mathcal{F}_\delta, \sqsubseteq)$ are CCPOs. Moreover, the concatenation operations are monotone and even continuous with respect to the approximation orderings defined in each case. We thus have in both cases compatible algebraic and order structures. This is perhaps the most pleasing aspect of these models and actually the original motivation for their investigation.

Now, $(\mathcal{F}_\alpha, \sqsubseteq)$ is shown in [DG95] to be $\mathcal{p}$-algebraic for any dependence alphabet, hence makes a good candidate as a process algebra. The main drawback of $(\mathcal{F}_\alpha, \sqsubseteq)$ is that we do not recover in the sequential case the classical completion using one termination symbol and one non-termination symbol, since we allow for any subset of actions to be specified on the second component. This information about the future of the process is highly redundant in the sequential case and reveals, as a matter of fact, that the $\alpha$-model is not abstract enough.

The case of $(\mathcal{F}_\delta, \sqsubseteq)$ is in a certain sense exactly opposite. We do recover the classical completion in the sequential case, since the alphabetic information about the future actions restricts to dependency and, therefore, delivers either $\top = \emptyset$, or $\bot = \Sigma$, in all cases. However, $(\mathcal{F}_\delta, \sqsubseteq)$ is only $k$-algebraic, but fails to be $\mathcal{p}$-algebraic in general. One can easily check this even on small dependence alphabets such as the square or the line of four actions [DG95]. This indicates that the $\delta$-model is too abstract, in the sense that some of the (atomic) information necessary for $\mathcal{p}$-algebraicity has been abstracted away.

20
The next propositions show that the cause for this lack of p-algebraicity comes indeed from the fact that the family of sets appearing on the second component, \(D(P(\Sigma)) = \{ D(A) \subseteq \Sigma | A \subseteq \Sigma \} \), is not p-algebraic with respect to inclusion. The p-algebraicity of \((D(P(\Sigma)), \subseteq)\) is furthermore shown to be equivalent to a representation condition for the dependence alphabet \((\Sigma, D)\). To this purpose we need the following definitions.

Let us say that two actions share an action \(a\) iff they are both dependent on \(a\). An action \(a \in \Sigma\) is called a resource (register) iff any two actions sharing it are mutually dependent. This is tantamount to saying that \(D(a)\) forms a clique of the dependence alphabet \((\Sigma, D)\).

**Proposition 7.2**

- (1) Suppose \(\alpha \in \Sigma\) is a resource iff \(\alpha \in D(a)\) implies \(D(\alpha) \subseteq D(a)\) for all \(a \in \Sigma\).
- (2) The prime elements of the lattice \((D(P(\Sigma)), \subseteq)\) are the subsets \(D(\alpha)\), where \(\alpha\) is a resource.

**Proof:**

(1) Suppose \(\alpha\) is a resource and \(\alpha \in D(a)\). Then \(D(\alpha)\) is a clique and \(a \in D(\alpha)\), therefore, \(D(\alpha) \subseteq D(a)\). Conversely, let \(a, b \in D(\alpha)\). Then \(\alpha \in D(a)\), hence \(b \in D(\alpha) \subseteq D(a)\), which shows that \(D(\alpha)\) is a clique.

(2) The family of sets \((D(P(\Sigma)), \subseteq)\) is closed under unions, hence the supremum of any set of elements in \((D(P(\Sigma)), \subseteq)\) is given by their union. Furthermore, since \(D(A) = \bigcup_{a \in A} D(\alpha)\) for all \(A \subseteq \Sigma\), we can easily check that any prime in \((D(P(\Sigma)), \subseteq)\) must be of the form \(D(\alpha)\) for some \(\alpha \in \Sigma\). We show that \(D(\alpha)\) is prime iff \(\alpha\) is a resource.

Let first \(\alpha\) be a resource and assume that \(D(\alpha) \subseteq \bigcup_{a \in A} D(a)\). Since \(\alpha \in D(\alpha)\), there exists some \(a \in A\), such that \(\alpha \in D(a)\). By (1) this implies \(D(\alpha) \subseteq D(a)\), thus showing that \(D(\alpha)\) is prime.

Let now \(D(\alpha)\) be prime. We show that \(\alpha \in D(a)\) implies \(D(\alpha) \subseteq D(a)\) for all \(a \in \Sigma\), which by (1) proves that \(\alpha\) is a resource. Assume that \(\alpha \in D(a)\). We have \(D(\alpha) \subseteq D(a) \cup D(\alpha) = D(a) \cup (D(\alpha) \setminus D(a)) \subseteq D(\alpha) \cup D(D(\alpha) \setminus D(a))\). As \(D(\alpha)\) is prime, we must have either the case that \(D(\alpha) \subseteq D(\alpha)\), or else that \(D(\alpha) \subseteq D(D(\alpha) \setminus D(a))\). Now, \(D(\alpha) \cap (D(\alpha) \setminus D(a)) = \emptyset\) hence \(a \notin D(D(\alpha) \setminus D(a))\). Since \(a \in D(a)\) this implies that \(D(\alpha) \not\subseteq D(D(\alpha) \setminus D(a))\), which excludes the second case. We are therefore left with the first case, namely, \(D(\alpha) \subseteq D(a)\).

Let \(R \subseteq \Sigma\) denote the set of resources of \((\Sigma, D)\). The function \(\text{res} : P(\Sigma) \to P(R)\), \(\text{res}(A) = R \cap D(A)\) assigns to every subset \(A \subseteq \Sigma\) the set of resources on which it depends. Note that \(\text{res}(A \cup B) = \text{res}(A) \cup \text{res}(B)\) for all \(A, B \subseteq \Sigma\). In particular, \(\text{res}(A) = \bigcup_{a \in A} \text{res}(a)\), and using Proposition 7.2.(1), we see that \(\text{res}(A) = \{ \alpha \in R \mid D(\alpha) \subseteq D(A)\}\), hence \(D(\text{res}(A)) \subseteq D(A)\) for all \(A \subseteq \Sigma\). By Proposition 7.2.(2), \(\text{res}(A)\) denotes therefore all actions \(a\), such that \(D(\alpha)\) is a prime below \(D(A)\) in \((D(P(\Sigma)), \subseteq)\).

The very definition of resources guarantees that two actions are mutually dependent, whenever they share some common resource, i.e. \(\text{res}(a) \cap \text{res}(b) \neq \emptyset\) implies \(a \in D(b)\) for all \(a, b \in \Sigma\). A dependence alphabet is said to be resource representable iff the converse implication is also true, that is, iff two actions share some common resource, whenever they are mutually dependent, i.e. \(a \in D(b)\) implies \(\text{res}(a) \cap \text{res}(b) \neq \emptyset\) for all \(a, b \in \Sigma\).

**Proposition 7.3** The following statements are equivalent for any dependence alphabet \((\Sigma, D)\):

1. \(\Sigma\) is a resource alphabet.
2. \(D(\alpha)\) is prime for all \(\alpha \in \Sigma\).
3. \(\text{res}(a) \cap \text{res}(b) \neq \emptyset\) implies \(a \in D(b)\) for all \(a, b \in \Sigma\).
4. \(\text{res}(a) \cap \text{res}(b) \neq \emptyset\) implies \(a \in D(b)\) for all \(a, b \in \Sigma\), and \(D(\alpha)\) is prime for all \(\alpha \in \Sigma\).
5. \(\text{res}(a) \cap \text{res}(b) \neq \emptyset\) implies \(a \in D(b)\) for all \(a, b \in \Sigma\), and \(D(\alpha)\) is prime for all \(\alpha \in \Sigma\) and \(\text{res}(\alpha) \subseteq \text{res}(\beta)\) for all \(\alpha, \beta \in \Sigma\).

21
(1) \((\mathbb{F}_\delta, \sqsubseteq)\) is \(p\)-algebraic.

(2) \((D(\mathcal{P}(\Sigma)), \supseteq)\) is a \(p\)-algebraic (distributive) lattice.

(3) \(D(A) = D(\text{res}(A))\) for all \(A \subseteq \Sigma\).

(4) \((\Sigma, D)\) is resource representable.

**Proof:**

(1) \(\Rightarrow\) (2): Let \((\mathbb{F}_\delta, \sqsubseteq)\) be \(p\)-algebraic. The set of \(\delta\)-traces \(\{ (1, D(A)) \mid A \in \Sigma \}\) is a downward closed subset of \((\mathbb{F}_\delta, \sqsubseteq)\). Hence, as can be easily checked, the restriction of the approximation ordering to this subset is \(p\)-algebraic. But this suborder is evidently isomorphic to the lattice \((D(\mathcal{P}(\Sigma)), \supseteq)\), which is therefore \(p\)-algebraic (distributive) too.

(2) \(\Rightarrow\) (3): If \((D(\mathcal{P}(\Sigma)), \supseteq)\) is a \(p\)-algebraic lattice, it follows from general lattice-theoretical arguments that its dual \((D(\mathcal{P}(\Sigma)), \sqsubseteq)\) is a \(p\)-algebraic lattice, too. Therefore, every element of \((D(\mathcal{P}(\Sigma)), \sqsubseteq)\) is the supremum of the primes below itself. Using Proposition 7.2.(2), we see that \(D(\mathcal{P}(\Sigma))\) is a \(p\)-algebraic lattice, too. Therefore, every element of \((D(\mathcal{P}(\Sigma)), \sqsubseteq)\) is \(p\)-algebraic (distributive) too.

(3) \(\Rightarrow\) (4): For any \(a, b \in \Sigma\), such that \(a \mathcal{D} b\), we have \(b \in D(a) = D(\text{res}(a))\). Consequently, there exists \(\alpha \in \text{res}(a)\) with \(b \in D(\alpha)\). But then \(\alpha \in \mathcal{R} \cap D(b) = \text{res}(b)\), which implies that \(\alpha = \text{res}(\alpha) \cap \text{res}(b) \neq \emptyset\).

(4) \(\Rightarrow\) (1): We show that \((\mathbb{F}_\delta(\Sigma, D), \sqsubseteq_\delta)\) is isomorphic to \((\mathbb{F}_\rho(\Sigma, \mathcal{R}', \text{res}'), \sqsubseteq_\rho)\) for a suitable resource alphabet \((\Sigma, \mathcal{R}', \text{res}')\), which by Theorem 5.7 entails that it is \(p\)-algebraic.

Let \((\Sigma, D)\) be resource representable. This implies directly from the definition that the restriction of \(D\) to the set of resources \(\mathcal{R}\) is an equivalence relation. Let \(\mathcal{R}' \subseteq \mathcal{R}\) denote a set of representatives with respect to this equivalence relation and define the resource map \(\text{res}' : \Sigma \to \mathcal{P}(\mathcal{R}') \setminus \{\emptyset\}\), \(\text{res}'(a) = D(\alpha) \cap \mathcal{R}'\), which we extend in the canonical way to \(\mathcal{P}(\Sigma)\). Obviously, \(\mathcal{R}'\) is composed of pairwise independent actions, hence, for any \(R \subseteq \mathcal{R}'\)

\[
\text{res}'(R) = R.
\]

(1)

For all \(a, b \in \Sigma\) we have by definition that \(a \mathcal{D} b\) iff there exists \(\alpha \in \mathcal{R}\) such that \(a, b \in D(\alpha)\). Since \(\alpha\) is a resource and for the representative \(\alpha' \in \mathcal{R}'\) of \(\alpha\) we have \(\alpha' \in D(\alpha)\), it follows that \(a, b \in D(\alpha')\). Hence, \((a, b) \in D\) if and only if there exists \(\alpha' \in \mathcal{R}'\) such that \(a, b \in D(\alpha')\), which can be rewritten as \(\text{res}'(a) \cap \text{res}'(b) \neq \emptyset\). This shows that \(D\) is the dependence relation induced by the resource mapping \(\text{res}'\).

For all \(b \in \Sigma\) and \(A \subseteq \Sigma\), we then have that \(b \in D(A)\) iff \(\text{res}'(b) \cap \text{res}'(A) \neq \emptyset\) iff \(D(b) \cap D(A) \neq \emptyset\) iff \(b \in D(\text{res}'(A))\). Therefore, for all \(A \subseteq \Sigma\),

\[
D(A) = D(\text{res}'(A)).
\]

(2)

Together with the defining equation \(\text{res}'(A) = D(A) \cap \mathcal{R}'\) we obtain for any \(A, B \subseteq \Sigma\) the equivalence

\[
D(A) \subseteq D(B) \Leftrightarrow \text{res}'(A) \subseteq \text{res}'(B).
\]

(3)

Let now \(\mathbb{F}_\rho(\Sigma, \mathcal{R}', \text{res}')\) be the set of \(\rho\)-traces associated with the resource alphabet \((\Sigma, \mathcal{R}', \text{res}')\) and consider the mapping \(\psi : (\mathbb{F}_\rho(\Sigma, \mathcal{R}', \text{res}'), \sqsubseteq_\rho) \to (\mathbb{F}_\delta(\Sigma, D), \sqsubseteq_\delta)\) defined by \((r, R) \mapsto (r, D(R))\).

First, from equation (2) we see that the mapping \(\psi\) is surjective.

22
Secondly, using equation (1) and equation (3) we obtain the following chain of equivalences for any \( x, y \in F_\rho(\Sigma, R', res') \), \( x = (r, R) \), \( y = (s, S) \):

\[
x \sqsubseteq_\rho y \iff r \leq s \text{ and } res'(r^{-1}s) \cup S \subseteq R
\]

\[
\iff r \leq s \text{ and } D(r^{-1}s) \cup D(S) \subseteq D(R)
\]

\[
\iff (r, D(R)) \sqsubseteq_\delta (s, D(S))
\]

\[
\iff \psi(x) \sqsubseteq_\delta \psi(y)
\]

This proves that the mapping \( \psi \) is injective and, furthermore, that \( \psi \) and \( \psi^{-1} \) are monotone. Thus \( \psi \) is an order isomorphism between \((F_\delta(\Sigma, D), \sqsubseteq_\delta)\) and \((F_\rho(\Sigma, R', res'), \sqsubseteq_\rho)\), which allows by Theorem 5.7 to finally conclude that \((F_\delta(\Sigma, D), \sqsubseteq_\delta)\) is p-algebraic.

Let us furthermore note that the mapping \( \psi \) is also a monoid isomorphism between \((F_\rho(\Sigma, R', res'), \cdot_\rho)\) and \((F_\delta(\Sigma, D), \cdot_\delta)\).  

The proposition above provides an efficient algorithm to test the p-algebraicity of \( F_\delta \) by checking the resource representability of the dependence graph. One can easily check by hand that this condition is not fulfilled by the square or the line of four actions. Furthermore, all dependence alphabets not containing these two configurations as induced subgraphs can be shown to satisfy resource representability. Indeed, it is well-known that this class of graphs coincides with the so-called transitive forests, that is graphs obtained by taking the transitive closure of the descendent relation in a disjoint union of rooted trees. One can easily see that in a transitive forest, two vertices are connected by an edge if and only if they lie on a common branch. Hence, all leaves are resources and two actions are dependent if and only if they depend on a common leaf, which shows that the dependence alphabet is, in effect, resource representable. Moreover, the set of leaves of the transitive forest is a set of representatives of the equivalence classes of resources defined in the proof of Proposition 7.3.

On the other hand, it is clear that the class of resource representable graphs is not definable by forbidden induced subgraphs, since every graph can be completed to a resource representable graph, by extending the vertex set with the elements of any clique covering of the graph and by setting every clique dependent on the vertices it contains.

The proof of the theorem also shows the remarkable fact that in the case of representable dependence alphabets the model of \( \delta \)-traces is isomorphic to a particular model of resource traces where the set of resources is chosen as a subset of the action set. The idea of dropping this additional restriction meanwhile retaining the equivalence between dependence and resource sharing lead us in fact to the investigation of the more abstract model of resource traces which we have presented in this paper.

8 Conclusion and Outlook

Much of the theory presented above can be substantially simplified by using stable embedding-projection pairs. We intend to do this as the next step, in order to clarify the mechanism by which a minimal completion, having compatible algebraic and order structures, can be achieved for trace monoids.

There also remains some non-trivial work to be done, which consists in identifying for all dependence alphabets a canonical set of resources. Since the resource (register) representation we use is similar to the one employed in constructing asynchronous au-
tomata, we think that Zielonka’s time-stamping function could provide such a standard representation.

The domain-theoretic properties which we have proved for resource traces allow to define a denotational semantics for recursive process terms involving solely a concatenation operation. This language being still rather poor, the primary next concern will be to define on the set of resource traces further operations commonly used in CSP-like process algebras such as for instance a parallel composition, a hiding and a choice operator. A first step in this direction containing a detailed presentation of the denotational and operational semantics of a deterministic parallel operator is presented in [GM99].

Acknowledgment

We would like to thank V. Diekert for fruitful discussions and in particular for pointing out that transitive forests satisfy our resource representability condition.

References


