Modelling Concurrency with Quotient Monoids

Ryszard Janicki[⋆] Dai Tri Man Lê[⋆⋆]

Department of Computing and Software,
McMaster University,
Hamilton, ON, L8S 4K1 Canada
{janicki,ledt}@mcmaster.ca

Abstract. Four quotient monoids over step sequences and one with compound generators are introduced and discussed. They all can be regarded as extensions (of various degrees) of Mazurkiewicz traces [14] and comtraces of [10].

Keywords: generalised trace theory, trace monoids, step sequences, stratified partial orders, stratified order structures.

1 Introduction

Mazurkiewicz traces or partially commutative monoids [1, 5] are quotient monoids over sequences (or words). They have been used to model various aspects of concurrency theory since the late seventies and their theory is substantially developed [5]. As a language representation of partial orders, they can nicely model "true concurrency."

For Mazurkiewicz traces, the basic monoid (whose elements are used in the equations that define the trace congruence) is just a free monoid of sequences. It is assumed that generators, i.e. elements of trace alphabet, have no visible internal structure, so they could be interpreted as just names, symbols, letters, etc. This can be a limitation, as when the generators have some internal structure, for example if they are sets, this internal structure may be used when defining the set of equations that generate the quotient monoid. In this paper we will assume that the monoid generators have some internal structure. We refer to such generators as 'compound', and we will use the properties of that internal structure to define an appropriate quotient congruence.

One of the limitations of traces and the partial orders they generate is that neither traces nor partial orders can model the "not later than" relationship [9]. If an event a is performed "not later than" an event b, and let the step $\{a, b\}$ model the simultaneous performance of a and b, then this "not later than" relationship can be modelled by the following set of two step sequences $s = \{\{a\}\{b\}, \{a, b\}\}$. But the set s cannot be represented by any trace. The problem is that the trace independency relation is symmetric, while the "not later than" relationship is not, in general, symmetric.

⋆ Partially supported by NSERC grant of Canada.
⋆⋆ Partially supported by Ontario Graduate Scholarship.

To overcome those limitations the concept of a *comtrace* (*com*bined *trace*) was introduced in [10]. Comtraces are finite sets of equivalent step sequences and the congruence is determined by a relation *ser*, which is called *serialisability* and is, in general, not symmetric. Monoid generators are 'steps', i.e. finite sets, so they have internal structure. The set union is used to define comtrace congruence. Comtraces provide a formal language counterpart to *stratified order structures* and were used to provide a semantic of Petri nets with inhibitor arcs. However [10] contains very little theory of comtraces, and only their relationship to stratified order structures has been substantially developed.

Stratified order structures [6, 8, 10, 11] are triples $(X, \prec, \sqsubset)$, where $\prec$ and $\sqsubset$ are binary relations on X. They were invented to model both "earlier than" (the relation $\prec$) and "not later than" (the relation $\sqsubset$) relationships, under the assumption that all system runs are modelled by stratified partial orders, i.e. step sequences. They have been successfully applied to model inhibitor and priority systems, asynchronous races, synthesis problems, etc. (see for example [10, 12, 13] and others). It was shown in [10] that each comtrace defines a finite stratified order structure. However, thus far, comtraces have been used much less often than stratified order structures, even though in many cases they appear to be more natural than stratified order structures. Perhaps this is due to the lack of substantial theory development of quotient monoids different from that of Mazurkiewicz traces.

It appears that comtraces are a special case of a more general class of quotient monoids, which will be called *absorbing monoids*. For absorbing monoids, generators are still steps, i.e. sets. When sets are replaced by arbitrary compound generators (together with appropriate rules for the generating equations), a new model, called *absorbing monoids with compound generators*, is created. This model allows us to describe formally *asymmetric synchrony*.

Both comtraces and stratified order structures can adequately model concurrent histories only when the paradigm π_3 of [9, 11] is satisfied. For the general case, we need *generalised stratified order structures*, which were introduced and analysed in [7]. Generalised stratified order structures are triples $(X, <\!\!>, \sqsubset)$, where $<\!\!>$ and $\sqsubset$ are binary relations on X modelling "earlier than or later than" and "not later than" relationships respectively under the assumption that all system runs are modelled by stratified partial orders.

In this paper a sequence counterpart of generalised stratified order structures, called *generalised comtraces*, and their equational generalisation, called *partially commutative absorbing monoids*, are introduced and their properties are discussed.

In the next section we recall the basic concepts of partial orders and the theory of monoids. Section 3 introduces *equational monoids with compound generators* and other types of monoids that are discussed in this paper. In Section 4 the concept of canonical representations of traces is reviewed; while Section 5 proves the *uniqueness of canonical representations for comtraces*. In Section 6 the notion of *generalised comtraces* is introduced and the relationship between comtraces, generalised comtraces and their respective order structures is thoroughly discussed. Section 7 briefly describes the relationship between comtraces and different paradigms of concurrent histories, and Section 8 contains some final comments.

2 Orders, Monoids, Sequences and Step Sequences

Let X be a set. A relation $\prec \subseteq X \times X$ is a (*strict*) *partial order* if it is irreflexive and transitive, i.e. if $\neg(a \prec a)$ and $a \prec b \prec c \;\Rightarrow\; a \prec c$, for all $a, b, c \in X$.

We write $a \simeq_{\prec} b$ if $\neg(a \prec b) \wedge \neg(b \prec a)$, that is if a and b are either *distinct incomparable* (w.r.t. $\prec$) or *identical* elements of X; and $a \frown_{\prec} b$ if $a \simeq_{\prec} b \wedge a \neq b$.

We will also write $a \prec^{\frown} b$ if $a \prec b \vee a \frown_{\prec} b$.

The partial order $\prec$ is *total* (or *linear*) if $\frown_{\prec}$ is empty, and *stratified* (or *weak*) if $\simeq_{\prec}$ is an equivalence relation.

The partial order $\prec_2$ is an *extension* of $\prec_1$ iff $\prec_1 \subseteq \prec_2$. Every partial order is uniquely represented by the intersection of all its total extensions.

A triple $(X, \circ, 1)$, where X is a set, $\circ$ is a total binary operation on X, and $1 \in X$, is called a *monoid*, if $(a \circ b) \circ c = a \circ (b \circ c)$ and $a \circ 1 = 1 \circ a = a$, for all $a, b, c \in X$.

A nonempty equivalence relation $\sim \subseteq X \times X$ is a *congruence* in the monoid $(X, \circ, 1)$ if

$$a_1 \sim b_1 \wedge a_2 \sim b_2 \Rightarrow (a_1 \circ a_2) \sim (b_1 \circ b_2),$$

for all $a_1, a_2, b_1, b_2 \in X$. Standardly $X/\sim$ denotes the set of all equivalence classes of $\sim$ and $[a]_{\sim}$ (or simply $[a]$) denotes the equivalence class of $\sim$ containing the element $a \in X$. The triple $(X/\sim, \hat{\circ}, [1])$, where $[a]\hat{\circ}[b] = [a \circ b]$, is called the *quotient monoid* of $(X, \circ, 1)$ under the congruence $\sim$. The mapping $\phi : X \to X/\sim$ defined as $\phi(a) = [a]$ is called the *natural homomorphism* generated by the congruence $\sim$ (for more details see for example [2]). The symbols $\circ$ and $\hat{\circ}$ are often omitted if this does not lead to any discrepancy.

By an *alphabet* we shall understand any finite set. For an alphabet Σ, Σ^* denotes the set of all finite sequences of elements of Σ, λ denotes the empty sequence, and any subset of Σ^* is called a *language*. In this paper all sequences are finite. Each sequence can be interpreted as a total order and each finite total order can be represented by a sequence. The triple $(\Sigma^*, \cdot, \lambda)$, where $\cdot$ is sequence concatenation (usually omitted), is a *monoid* (of sequences).

For each set X, let $\mathscr{P}(X)$ denote the set of all subsets of X and $\mathscr{P}^{\emptyset}(X)$ denote the set of all *non-empty* subsets of X. Consider an alphabet $\Sigma_{step} \subseteq \mathscr{P}^{\emptyset}(X)$ for some finite X. The elements of Σ_{step} are called *steps* and the elements of Σ_{step}^* are called *step sequences*. For example if $\Sigma_{step} = \{\{a\}, \{a, b\}, \{c\}, \{a, b, c\}\}$ then $\{a, b\}\{c\}\{a, b, c\} \in \Sigma_{step}^*$ is a step sequence. The triple $(\Sigma_{step}^*, \bullet, \lambda)$, where $\bullet$ is step sequence concatenation (usually omitted), is a *monoid* (of step sequences) (see for example [10] for details).

3 Equational Monoids with Compound Generators

In this section we will define all types of monoids that are discussed in this paper.

3.1 Equational Monoids and Mazurkiewicz Traces

Let $M = (X, \circ, 1)$ be a *monoid* and let $EQ = \{\ x_1 = y_1\ ,\ \ldots\ ,\ x_n = y_n\ \}$, where $x_i, y_i \in X$, $i = 1, ..., n$, be a finite set of *equations*. Define $\equiv_{EQ}$ (or just $\equiv$) as the *least congruence* on M satisfying, $x_i = y_i \implies x_i \equiv_{EQ} y_i$, for each equation $x_i = y_i \in EQ$. We will call the relation $\equiv_{EQ}$ the *congruence defined by* EQ, or EQ*-congruence*.

The *quotient monoid* $M_{\equiv} = (X/\!\equiv, \hat{\circ}, [1]_{\equiv})$, where $[x]\hat{\circ}[y] = [x \circ y]$, will be called an *equational monoid* (see for example [15]).

The following folklore result shows that the relation $\equiv_{EQ}$ can also be defined explicitly.

Proposition 1. *For equational monoids the EQ-congruence $\equiv$ can be defined explicitly as the reflexive and transitive closure of the relation $\approx \cup \approx^{-1}$, i.e. $\equiv = (\approx \cup \approx^{-1})^*$, where $\approx \subseteq X \times X$, and*

$$x \approx y \iff \exists\, x_1, x_2 \in X.\ \exists\, (u = w) \in EQ.\ x = x_1 \circ u \circ x_2 \wedge y = x_1 \circ w \circ x_2.$$

Proof. Define $\dot{\approx} = \approx \cup \approx^{-1}$. Clearly $(\dot{\approx})^*$ is an equivalence relation. Let $x_1 \equiv y_1$ and $x_2 \equiv y_2$. This means $x_1 (\dot{\approx})^k y_1$ and $x_2 (\dot{\approx})^l y_2$ for some $k, l \geq 0$. Hence $x_1 \circ x_2\ (\dot{\approx})^k\ y_1 \circ x_2\ (\dot{\approx})^l\ y_1 \circ y_2$, i.e. $x_1 \circ x_2 \equiv y_1 \circ y_2$. Therefore $\equiv$ is a congruence. Let $\sim$ be a congruence that satisfies $(u = w) \in EQ \implies u \sim w$ for each $u = w$ from EQ. Clearly $x \dot{\approx} y \implies x \sim y$. Hence $x \equiv y \iff x (\dot{\approx})^m y \implies x \sim^m y \Rightarrow x \sim y$. Thus $\equiv$ is the least. □

If $M = (E^*, \circ, \lambda)$ is a *free monoid* generated by E, $ind \subseteq E \times E$ is an irreflexive and symmetric relation (called *independency* or *commutation*), and $EQ = \{ab = ba \mid (a, b) \in ind\}$, then the quotient monoid $M_{\equiv} = (E^*/\!\equiv, \hat{\circ}, [\lambda])$ is a *partially commutative free monoid* or *monoid of Mazurkiewicz traces* [5, 14]. The tuple (E, ind) is often called *concurrent alphabet*.

Example 1. Let $E = \{a, b, c\}$, $ind = \{(b, c), (c, b)\}$, i.e. $EQ = \{\ bc = cb\ \}$. For example $abcbca \equiv accbba$ (since $abcbca \approx acbbca \approx acbcba \approx accbba$), $t_1 = [abc] = \{abc, acb\}$, $t_2 = [bca] = \{bca, cba\}$ and $t_3 = [abcbca] = \{abcbca, abccba, acbbca, acbcba, abbcca, accbba\}$ are traces, and $t_3 = t_1 \hat{\circ} t_2$ (as $[abcbca] = [abc]\hat{\circ}[bca]$). For more details the reader is referred to [5, 14] (and [15] for equational representations). □

3.2 Absorbing Monoids and Comtraces

The standard definition of a free monoid $(E^*, \circ, \lambda)$ assumes the elements of E have no internal structure (or their internal structure does not affect any monoidal properties), and they are often called 'letters', 'symbols', 'names', etc. When we assume the elements of E have some internal structure, for instance they are sets, this internal structure may be used when defining the set of equations EQ.

Let E be a finite set and $\mathcal{S} \subseteq \mathscr{P}^{\emptyset}(E)$ be a non-empty set of non-empty subsets of E satisfying $\bigcup_{A\in\mathcal{S}} A = E$. The free monoid $(\mathcal{S}^*, \circ, \lambda)$ is called a *free monoid of step sequences* over E, with the elements of $\mathcal{S}$ called *steps* and the elements of $\mathcal{S}^*$ called *step sequences*. We assume additionally that the set $\mathcal{S}$ is *subset closed* i.e. for all $A \in \mathcal{S}$, $B \subseteq A$ and B is not empty, implies $B \in \mathcal{S}$.

Let EQ be the following set of equations:

$$EQ = \{\ C_1 = A_1B_1\ ,\ \dots\ ,\ C_n = A_nB_n\ \},$$

where $A_i, B_i, C_i \in \mathcal{S}$, $C_i = A_i \cup B_i$, $A_i \cap B_i = \emptyset$, for $i = 1, \dots, n$, and let $\equiv$ be EQ-congruence (i.e. the least congruence satisfying $C_i = A_iB_i$ implies $C_i \equiv A_iB_i$).

The quotient monoid $(\mathcal{S}^*/\equiv, \hat{\circ}, [\lambda])$ will be called an ***absorbing monoid* over step sequences**.

Example 2. Let $E = \{a, b, c\}$, $\mathcal{S} = \{\{a, b, c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a\}, \{b\}, \{c\}\}$, and EQ be the following set of equations:

$$\{a, b, c\} = \{a, b\}\{c\} \quad \text{and} \quad \{a, b, c\} = \{a\}\{b, c\}.$$

In this case, for example, $\{a, b\}\{c\}\{a\}\{b, c\} \equiv \{a\}\{b, c\}\{a, b\}\{c\}$ (as we have $\{a, b\}\{c\}\{a\}\{b, c\} \approx \{a, b, c\}\{a\}\{b, c\} \approx \{a, b, c\}\{a, b, c\} \approx \{a\}\{b, c\}\{a, b, c\} \approx \{a\}\{b, c\}\{a, b\}\{c\}$), $x = [\{a, b, c\}]$ and $y = [\{a, b\}\{c\}\{a\}\{b, c\}]$ belong to $\mathcal{S}^*/\equiv$, and

$x = \{\{a, b, c\}, \{a, b\}\{c\}, \{a\}\{b, c\}\}$,
$y = \{\{a, b, c\}\{a, b, c\}, \{a, b, c\}\{a, b\}\{c\}, \{a, b, c\}\{a\}\{b, c\}, \{a, b\}\{c\}\{a, b, c\},$
$\{a, b\}\{c\}\{a, b\}\{c\}, \{a, b\}\{c\}\{a\}\{b, c\}, \{a\}\{b, c\}\{a, b, c\},$
$\{a\}\{b, c\}\{a, b\}\{c\}, \{a\}\{b, c\}\{a\}\{b, c\}\}$.

Note that $y = x\hat{\circ}x$ as $\{a, b\}\{c\}\{a\}\{b, c\} \equiv \{a, b, c\}\{a, b, c\}$. □

Comtraces, introduced in [10] as an extension of Mazurkiewicz traces to distinguish between "earlier than" and "not later than" phenomena, are a special case of absorbing monoids of step sequences. The equations EQ are in this case defined implicitly via two relations *simultaneity* and *serialisability*.

Let E be a finite set (of events), $ser \subseteq sim \subset E \times E$ be two relations called *serialisability* and *simultaneity* respectively. The triple (E, sim, ser) is called *comtrace alphabet*. We assume that sim is irreflexive and symmetric. Intuitively, if $(a, b) \in sim$ then a and b can occur simultaneously (or be a part of a *synchronous* occurrence in the sense of [12]), while $(a, b) \in ser$ means that a and b may occur simultaneously and a may occur before b (and both happenings are equivalent). We define $\mathcal{S}$, the set of all (potential) *steps*, as the set of all cliques of the graph (E, sim), i.e.

$$\mathcal{S} = \{A \mid A \neq \emptyset \wedge (\forall a, b \in A.\ a = b \vee (a, b) \in sim)\}.$$

The set of equations EQ can now be defined as:

$$EQ = \{C = AB \mid C = A \cup B \in \mathcal{S} \wedge A \cap B = \emptyset \wedge A \times B \subseteq ser\}.$$

Let $\equiv$ be the EQ-congruence defined by the above set of equations. The absorbing monoid $(\mathcal{S}/\equiv, \hat{\circ}, [\lambda])$ is called a monoid of *comtraces*.

Example 3. Let $E = \{a, b, c\}$ where a, b and c are three atomic operations defined as follows (we assume simultaneous reading is allowed):

$$a: \ y \leftarrow x + y, \quad b: \ x \leftarrow y + 2, \quad c: \ y \leftarrow y + 1.$$

Only b and c can be performed simultaneously, and the simultaneous execution of b and c gives the same outcome as executing b followed by c. We can then define $sim = \{(b, c), (c, b)\}$ and $ser = \{(b, c)\}$, and we have $\mathcal{S} = \{\{a\}, \{b\}, \{c\}, \{b, c\}\}$, $EQ = \{\{b, c\} = \{b\}\{c\}\}$. For example $x = [\{a\}\{b, c\}] = \{\{a\}\{b, c\}, \{a\}\{b\}\{c\}\}$ is a comtrace. Note that $\{a\}\{c\}\{b\} \notin x$. □

Even though Mazurkiewicz traces are quotient monoids over sequences and comtraces are quotient monoids over step sequences, Mazurkiewicz traces can be regarded as a special case of comtraces. In principle, each trace commutativity equation $ab = ba$ corresponds to two comtrace absorbing equations $\{a, b\} = \{a\}\{b\}$ and $\{a, b\} = \{b\}\{a\}$. This relationship can formally be formulated as follows.

Proposition 2. *If $ser = sim$ then for each comtrace $t \in \mathcal{S}^*/\equiv_{ser}$ there is a step sequence $x = \{a_1\} \ldots \{a_k\} \in \mathcal{S}^*$, where $a_i \in E$, $i = 1, ..., k$ such that $t = [x]$.*

Proof. Let $t = [A_1 ... A_m]$, where $A_i \in \mathcal{S}$, $i = 1, ..., m$. Hence $t = [A_1]...[A_m]$. Let $A_i = \{a^i_1, ..., a^i_{k_i}\}$. Since $ser = sim$, we have $[A_i] = [\{a^i_1\}]...[\{a^i_{k_i}\}]$, for $i = 1, \ldots, m$, which ends the proof. □

This means that if $ser = sim$, then each comtrace $t \in \mathcal{S}^*/\equiv_{ser}$ can be represented by a Mazurkiewicz trace $[a_1 \ldots a_k] \in E^*/\equiv_{ind}$, where $ind = ser$ and $\{a_1\} \ldots \{a_k\}$ is a step sequence such that $t = [\{a_1\} \ldots \{a_k\}]$. Proposition 2 guarantees the existence of $a_1 \ldots a_k$.

While every comtrace monoid is an absorbing monoid, *not every* absorbing monoid can be defined as a comtrace. For example the absorbing monoid analysed in Example 2 *cannot* be represented by any comtrace monoid.

It appears the concept of the comtrace can be very useful to formally define the concept of *synchrony* (in the sense of [12]). In principle the events are *synchronous* if they *can* be executed in one step $\{a_1, ..., a_k\}$ but this execution *cannot* be modelled by any sequence of proper subsets of $\{a_1, ..., a_k\}$. In general 'synchrony' is not necessarily 'simultaneity' as it does not include the concept of time [4]. However, it appears that the mathematics used to deal with synchrony is very close to that to deal with simultaneity.

Let (E, sim, ser) be a given comtrace alphabet. We define the relations ind, syn and the set $\mathcal{S}_{syn}$ as follows:

- $ind \subseteq E \times E$, called *independency* and defined as $ind = ser \cap ser^{-1}$,
- $syn \subseteq E \times E$, called *synchrony* and defined as:
  $(a, b) \in syn \iff (a, b) \in sim \wedge (a, b) \notin ser \cup ser^{-1}$,
- $\mathcal{S}_{syn} \subseteq \mathcal{S}$, called *synchronous steps*, and defined as:
  $A \in \mathcal{S}_{syn} \iff A \neq \emptyset \wedge (\forall a, b \in A.\ (a, b) \in syn)$.

If $(a,b) \in ind$ then a and b are *independent*, i.e. they may be executed either simultaneously, or a followed by b, or b followed by a, with exactly the same result. If $(a,b) \in syn$ then a and b are *synchronous*, which means they might be executed in one step, either $\{a,b\}$ or as a part of bigger step, but such an execution is not equivalent to neither a followed by b, nor b followed by a. In principle, the relation syn is a counterpart of 'synchrony' as understood in [12]. If $A \in \mathcal{S}_{syn}$ then the set of events A can be executed as one step, but it *cannot* be simulated by any sequence of its subsets.

Example 4. Let $E = \{a,b,c,d,e\}$, $sim = \{(a,b),(b,a),(a,c),(c,a),(a,d),(d,a)\}$, and $ser = \{(a,b),(b,a),(a,c)\}$. Hence
$\mathcal{S} = \{\{a,b\},\{a,c\},\{a,d\},\{a\},\{b\},\{c\},\{e\}\}$, $ind = \{(a,b),(b,a)\}$,
$syn = \{(a,d),(d,a)\}$, $\mathcal{S}_{syn} = \{\{a,d\}\}$.
Since $\{a,d\} \in \mathcal{S}_{syn}$ the step $\{a,d\}$ *cannot* be split. For example the comtraces $x_1 = [\{a,b\}\{c\}\{a\}]$, $x_2 = [\{e\}\{a,d\}\{a,c\}]$, $x_3 = [\{a,b\}\{c\}\{a\}\{e\}\{a,d\}\{a,c\}]$, are the following sets of step sequences:

$$x_1 = \{\{a,b\}\{c\}\{a\}, \{a\}\{b\}\{c\}\{a\}, \{b\}\{a\}\{c\}\{a\}, \{b\}\{a,c\}\{a\}\},$$
$$x_2 = \{\{e\}\{a,d\}\{a,c\}, \{e\}\{a,d\}\{a\}\{c\}\},$$
$$\begin{aligned}x_3 = \{&\{a,b\}\{c\}\{a\}\{e\}\{a,d\}\{a,c\}, \{a\}\{b\}\{c\}\{a\}\{e\}\{a,d\}\{a,c\},\\ &\{b\}\{a\}\{c\}\{a\}\{e\}\{a,d\}\{a,c\}, \{b\}\{a,c\}\{a\}\{e\}\{a,d\}\{a,c\},\\ &\{a,b\}\{c\}\{a\}\{e\}\{a,d\}\{a\}\{c\}, \{a\}\{b\}\{c\}\{a\}\{e\}\{a,d\}\{a\}\{c\},\\ &\{b\}\{a\}\{c\}\{a\}\{e\}\{a,d\}\{a\}\{c\}, \{b\}\{a,c\}\{a\}\{e\}\{a,d\}\{a\}\{c\}\}.\end{aligned}$$

Notice that we have $\{a,c\} \equiv_{ser} \{a\}\{c\} \not\equiv_{ser} \{c\}\{a\}$, since $(c,a) \notin ser$. We also have $x_3 = x_1 \hat{\circ} x_2$. □

3.3 Partially Commutative Absorbing Monoids and Generalised Comtraces

There are reasonable concurrent histories that cannot be modelled by any absorbing monoid. In fact, absorbing monoids can only model concurrent histories conforming to the paradigm π_3 of [9] (see the Section 7 of this paper). Let us analyse the following example.

Example 5. Let $E = \{a,b,c\}$ where a, b and c are three atomic operations defined as follows (we assume simultaneous reading is allowed):

$$a: \; x \leftarrow x+1, \quad b: \; x \leftarrow x+2, \quad c: \; y \leftarrow y+1.$$

It is reasonable to consider them all as 'concurrent' as any order of their executions yields exactly the same results (see [9, 11] for more motivation and formal considerations). Note that while simultaneous execution of $\{a,c\}$ and $\{b,c\}$ are allowed, the step $\{a,b\}$ *is not!*

Let us consider set of all equivalent executions (or runs) involving one occurrence of a, b and c

$$\begin{aligned}x = \{&\{a\}\{b\}\{c\}, \{a\}\{c\}\{b\}, \{b\}\{a\}\{c\}, \{b\}\{c\}\{a\}, \{c\}\{a\}\{b\}, \{c\}\{b\}\{a\},\\ &\{a,c\}\{b\}, \{b,c\}\{a\}, \{b\}\{a,c\}, \{a\}\{b,c\}\}.\end{aligned}$$

Although x is a valid concurrent history or behaviour [9, 11], it *is not* a comtrace.

The problem is that we have here $\{a\}\{b\} = \{b\}\{a\}$ but $\{a, b\}$ *is not* a valid step, so no absorbing monoid can represent this situation. $\square$

The concurrent behaviour described by x from Example 5 can easily be modelled by a *generalised order structure* of [7]. In this subsection we will introduce the concept of *generalised comtraces*, quotient monoids representations of generalised order structures. But we start with a slightly more general concept of *partially commutative absorbing monoid* over step sequences.

Let E be a finite set and let $(\mathcal{S}^*, \circ, \lambda)$ be a free monoid of step sequences over E. Assume also that $\mathcal{S}$ is subset closed.

Let EQ, EQ', EQ'' be the following sets of equations:

$$EQ' = \{\ C'_1 = A'_1B'_1\ ,\ \ldots\ ,\ C'_n = A'_nB'_n\ \},$$

where $A'_i, B'_i, C'_i \in \mathcal{S}$, $C'_i = A'_i \cup B'_i$, $A'_i \cap B'_i = \emptyset$, for $i = 1, ..., n$,

$$EQ'' = \{\ B''_1A''_1 = A''_1B''_1\ ,\ \ldots\ ,\ B''_kA''_k = A''_kB''_k\ \},$$

where $A''_i, B''_i \in \mathcal{S}$, $A''_i \cap B''_i = \emptyset$, $A''_i \cup B''_i \notin \mathcal{S}$, for $i = 1, ..., k$, and

$$EQ = EQ' \cup EQ''.$$

Let $\equiv$ be the EQ-congruence defined by the above set of equations EQ (i.e. the least congruence such that $C'_i = A'_iB'_i \implies C'_i \equiv A'_iB'_i$, for $i = 1, ..., n$ and $B''_iA''_i = A''_iB''_i \implies B''_iA''_i \equiv A''_iB''_i$, for $i = 1, ..., k$). The quotient monoid $(\mathcal{S}/\equiv, \hat{\circ}, [\lambda])$ will be called aa ***partially commutative absorbing monoid* over step sequences**.

There is a *substantial difference* between $ab = ba$ for Mazurkiewicz traces, and $\{a\}\{b\} = \{b\}\{a\}$ for partially commutative absorbing monoids. For traces, the equation $ab = ba$ when translated into step sequences corresponds to $\{a, b\} = \{a\}\{b\}$, $\{a, b\} = \{b\}\{a\}$, and implies $\{a\}\{b\} \equiv \{b\}\{a\}$. For partially commutative absorbing monoids, the equation $\{a\}\{b\} = \{b\}\{a\}$ implies that $\{a, b\}$ *is not a step*, i.e. neither $\{a, b\} = \{a\}\{b\}$ nor $\{a, b\} = \{b\}\{a\}$ does exist. For Mazurkiewicz traces the equation $ab = ba$ means 'independency', i.e. any order or simultaneous execution are allowed and are equivalent. For partially commutative absorbing monoids, the equation $\{a\}\{b\} = \{b\}\{a\}$ means that both orders are equivalent but simultaneous execution does not exist.

We will now extend the concept of a comtrace by adding a relation that generates the set of equations EQ''.

Let E be a finite set (of events), $ser, sim, inl \subset E \times E$ be three relations called *serialisability*, *simultaneity* and *interleaving* respectively. The triple (E, sim, ser, inl) is called *generalised comtrace alphabet*. We assume that both sim and inl are irreflexive and symmetric, and

$$ser \subseteq sim, \quad sim \cap inl = \emptyset.$$

Intuitively, if $(a, b) \in sim$ then a and b can occur simultaneously (or be a part of a *synchronous* occurrence), $(a, b) \in ser$ means that a and b may occur simultaneously and a may occur before b (and both happenings are equivalent), and

$(a,b) \in inl$ means a and b cannot occur simultaneously, but their occurrence in any order is equivalent. As for comtraces, we define $\mathcal{S}$, the set of all (potential) *steps*, as the set of all cliques of the graph (E, sim).

The set of equations EQ can now be defined as $EQ = EQ' \cup EQ''$, where:
$EQ' = \{C = AB \mid C = A \cup B \in \mathcal{S} \wedge A \cap B = \emptyset \wedge A \times B \subseteq ser\}$, and
$EQ'' = \{BA = AB \mid A \cup B \notin \mathcal{S} \wedge A \cap B = \emptyset \wedge A \times B \subseteq inl\}$.

Let $\equiv$ be the EQ-congruence defined by the above set of equations. The quotient monoid $(\mathcal{S}^*/\!\equiv, \hat{\circ}, [\lambda])$ is called a monoid of ***generalised comtraces***. If inl is empty we have a monoid of comtraces.

Example 6. The set x from Example 5 is an element of the generalised comtrace with $E = \{a,b,c\}$, $ser = sim = \{(a,c),(c,a),(b,c),(c,a)\}$, $inl = \{(a,b),(b,a)\}$, and $\mathcal{S} = \{\{a,c\},\{b,c\},\{a\},\{b\},\{c\}\}$, and for example $x = [\{a,c\}\{b\}]$. □

3.4 Absorbing Monoids with Compound Generators

One of the concepts that *cannot* easily be modelled by quotient monoids over step sequences, is *asymmetric synchrony*. Consider the following example.

Example 7. Let a and b be atomic and potentially simultaneous events, and c_1, c_2 be two synchronous compound events built entirely from a and b. Assume that c_1 is equivalent to the sequence $a \circ b$, c_2 is equivalent to the sequence $b \circ a$, but c_1 in *not* equivalent to c_2. This situation cannot be modelled by steps as from a and b we can built only one step $\{a,b\} = \{b,a\}$. To provide more intuition consider the following simple problem.

Assume we have a buffer of 8 bits. Each event a and b consecutively fills 4 bits. The buffer is initially empty and whoever starts first fills the bits 1–4 and whoever starts second fills the bits 5–8. Suppose that the simultaneous start is impossible (begins and ends are instantaneous events after all), filling the buffer takes time, and simultaneous (i.e. time overlapping in this case) executions are allowed. We clearly have two synchronous events $c_1 =$ '*a starts first but overlaps with b*', and $c_2 =$ '*b starts first but overlaps with a*'. We now have $c_1 = a \circ b$, and $c_2 = b \circ a$, but obviously $c_1 \neq c_2$ and $c_1 \not\equiv c_2$. □

To adequately model situations like that in Example 7 we will introduce the concept of *absorbing monoid with compound generators*.

Let $(G^*, \circ, \lambda)$ be a free monoid generated by G, where $G = E \cup C$, $E \cap C = \emptyset$. The set E is the set of *elementary* generators, while the set C is the set of *compound* generators. We will call $(G^*, \circ, \lambda)$ a *free monoid with compound generators.*

Assume we have a function $comp : G \to \mathscr{P}^{\emptyset}(E)$, called *composition* that satisfies: for all $a \in E$, $comp(a) = \{a\}$ and for all $a \notin E$, $|comp(a)| \geq 2$.

For each $a \in G$, $comp(a)$ gives the set of all elementary elements from which a is composed. It *may happen* that $comp(a) = comp(b)$ and $a \neq b$.

The set of absorbing equations is defined as follows:

$$EQ = \{c_i = a_i \circ b_1 \mid i = 1, ..., n\}$$

where for each $i = 1, ..., n$, we have:

- $a_i, b_i, c_i \in G$,
- $comp(c_i) = comp(a_i) \cup comp(b_i)$,
- $comp(a_i) \cap comp(b_i) = \emptyset$.

Let $\equiv$ be the EQ-congruence defined by the above set of equations EQ. The quotient monoid $(G^*/\equiv, \hat{\circ}, [\lambda])$ is called an ***absorbing monoid with compound generators***.

Example 8. Consider the absorbing monoid with compound generators where: $E = \{a, b, c_1, c_2\}$, $comp(c_1) = comp(c_2) = \{a, b\}$, $comp(a) = \{a\}$, $comp(b) = \{b\}$, and $EQ = \{ \quad c_1 = a \circ b, \quad c_2 = b \circ a \quad \}$. Now we have $[c_1] = \{c_1, a \circ b\}$ and $[c_2] = \{c_2, b \circ a\}$, which models the case from Example 7. □

4 Canonical Representations

We will show that all of the kinds of monoids discussed in previous sections have some kind of *canonical* representation, which intuitively corresponds to a maximally concurrent execution of concurrent histories [3].

Let (E, ind) be a concurrent alphabet and $(E^*/\equiv, \hat{\circ}, [\lambda])$ be a monoid of Mazurkiewicz traces. A sequence $x = a_1...a_k \in E^*$ is called *fully commutative* if $(a_i, a_j) \in ind$ for all $i \neq j$ and $i, j = 1, ..., k$.

A sequence $x \in E^*$ is in the *canonical form* if $x = \lambda$ or $x = x_1...x_n$ such that

- each x_i if fully commutative, for $i = 1, ..., n$,
- for each $1 \leq i \leq n-1$ and for each element a of x_{i+1} there exists an element b of x_i such that $a \neq b$ and $(a, b) \notin ind$.

If x is in the canonical form, then x is *a canonical representation* of $[x]$.

Theorem 1 ([1, 3]). *For every trace $t \in E^*/\equiv$, there exists $x \in E^*$ such that $t = [x]$ and x is in the canonical form.* □

With the canonical form as defined above, a trace may have more than one canonical representations. For instance the trace $t_3 = [abcbca]$ from Example 1 has four canonical representations: $abcbca, acbbca, abccba, acbcba$. Intuitively, x in the canonical form represents the maximally concurrent execution of a concurrent history represented by $[x]$. In this representation fully commutative sequences built from the same elements can be considered equivalent (this is better seen when *vector firing sequences* of [16] are used to represent traces, see [3] for more details). To get the uniqueness it suffices to order fully commutative sequences. For example we may introduce an arbitrary total order on E, extend it lexicographically to E^* and add the condition that in the representation $x = x_1...x_n$, each x_i is minimal with the lexicographic ordering. The canonical form with this additional condition is called *Foata canonical form*.

Theorem 2 ([1]). *Every trace has a unique representation in the Foata canonical form.* □

A canonical form as defined at the beginning of this Section can easily be adapted to step sequences and various equational monoids over step sequences, as well as to monoids with compound generators. In fact, step sequences better represent the intuition that canonical representation corresponds to the maximally concurrent execution [3].

Let $(\mathcal{S}^*, \circ, \lambda)$ be a free monoid of step sequences over E, and let

$$EQ = \{ \ C_1 = A_1B_1 \ , \ ... \ , \ C_n = A_nB_n \ \}$$

be an appropriate set of absorbing equations. Let $M_{absorb} = (\mathcal{S}^*/\!\equiv, \hat{\circ}, [\lambda])$.

A step sequence $t = A_1...A_k \in \mathcal{S}^*$ is *canonical* (w.r.t. M_{absorb}) if for all $i \geq 2$ there is *no* $B \in \mathcal{S}$ satisfying:

$$(\ \ A_{i-1} \cup B = A_{i-1}B \ \ ) \in EQ$$
$$(\ \ A_i = B(A_i - B) \ \ ) \in EQ$$

When M_{absorb} is a monoid of comtraces, the above definition is equivalent to the definition of canonical step sequence proposed in [10].

Let $(\mathcal{S}^*, \circ, \lambda)$ be a free monoid of step sequences over E, and let

$$EQ' = \{ \ C'_1 = A'_1B'_1 \ , \ ... \ , \ C'_n = A'_nB'_n \ \},$$
$$EQ'' = \{ \ B''_1A''_1 = A''_1B''_1 \ , \ ... \ , \ B''_kA''_k = A''_kB''_k \ \}$$

be an appropriate set of partially commutative absorbing equations. Then let $M_{abs\&pc} = (\mathcal{S}^*/\!\equiv, \hat{\circ}, [\lambda])$.

A step sequence $t = A_1...A_k \in \mathcal{S}^*$ is *canonical* (w.r.t. $M_{abs\&pc}$) if for all $i \geq 2$ there is *no* $B \in \mathcal{S}$ satisfying:

$$(\ \ A_{i-1} \cup B = A_{i-1}B \ \ ) \in EQ'$$
$$(\ \ A_i = B(A_i - B) \ \ ) \in EQ'$$

Note that the set of equations EQ'' *does not* appear in the above definition. Clearly the above definition also applies to generalised comtraces.

Let $(G^*, \circ, \lambda)$ be a free monoid with compound generators, and let

$$EQ = \{ \ c_1 = a_1b_1 \ , \ ... \ , \ c_n = a_nb_n \ \}$$

be an appropriate set of absorbing equations. Let $M_{cg\&absorb} = (G^*/\!\equiv, \hat{\circ}, [\lambda])$.

Finally, a sequence $t = a_1...a_k \in G^*$ is *canonical* (w.r.t. $M_{cg\&absorb}$) if for all $i \geq 2$ there is *no* $b, d \in G$ satisfying:

$$(\ \ c = a_{i-1}b \ \ ) \in EQ$$
$$(\ \ a_i = bd \ \ ) \in EQ$$

For all above definitions, if x is in the canonical form, then x is *a canonical representation* of $[x]$.

Theorem 3. *Let M_{absorb} be an absorbing monoid over step sequences, $\mathcal{S}$ its set of steps, and EQ its set of absorbing equations. For every step sequence $t \in \mathcal{S}^*$ there is a canonical step sequence u such that $t \equiv u$.*

Proof. For every step sequence $x = B_1...B_r$, let $\mu(x) = 1 \cdot |B_1| + ... + r \cdot |B_r|$. There is (at least one) $u \in [t]$ such that $\mu(u) \leq \mu(x)$ for all $x \in [t]$. Suppose $u = A_1...A_k$ is not canonical. Then there is $i \geq 2$ and a step $B \in \mathcal{S}$ satisfying:

$$\begin{array}{l} (\ \ A_{i-1} \cup B = A_{i-1}B\ \ ) \in EQ \\ (\ \ A_i = B(A_i - B)\ \ ) \in EQ \end{array}$$

If $B = A_i$ then $w \approx u$ and $\mu(w) < \mu(u)$, where

$$w = A_1...A_{i-2}(A_{i-1} \cup A_i)A_{i+1}...A_k.$$

If $B \neq A_i$, then $w \approx z$ and $u \approx z$ and $\mu(w) < \mu(u)$, where

$$\begin{array}{l} z = A_1...A_{i-2}A_{i-1}B(A_i - B)A_{i+1}...A_k \\ w = A_1...A_{i-2}(A_{i-1} \cup B)(A_i - B)A_{i+1}...A_k. \end{array}$$

In both cases it contradicts the minimality of $\mu(u)$. Hence u is canonical. □

For partially commutative absorbing monoids over step sequences the proof is virtually identical, the only change is to replace EQ with EQ'. The proof can also be adapted (some 'notational' changes only) to absorbing monoids with compound generators.

Corollary 1. *Let $M = (X, \hat{\circ}, [\lambda])$ be an absorbing monoid over step sequences, or partially commutative absorbing monoid over step sequences, or absorbing monoid with compound generators. For every $x \in X$ there is a canonical sequence u such that $x = [u]$.* □

Unless additional properties are assumed, the canonical representation is not unique for all three kinds of monoids mentioned in Corollary 1. To prove this, it suffices to show that this is not true for the absorbing monoids over step sequences. Consider the following example.

Example 9. Let $E = \{a, b, c\}$, $\mathcal{S} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a\}, \{b\}, \{c\}\}$ and EQ be the the following set of equations:

$$\{a, b\} = \{a\}\{b\}, \quad \{a, c\} = \{a\}\{c\}, \quad \{b, c\} = \{b\}\{c\}, \quad \{b, c\} = \{c\}\{b\}.$$

Note that $\{a, b\}\{c\}$ and $\{a, c\}\{b\}$ are canonical step sequences, and $\{a, b\}\{c\} \approx \{a\}\{b\}\{c\} \approx \{a\}\{b, c\} \approx \{a\}\{c\}\{b\} \approx \{a, c\}\{b\}$, i.e. $\{a, b\}\{c\} \equiv \{a, c\}\{b\}$. Hence $[\{a, b\}\{c\}] = \{\{a, b\}\{c\}, \{a\}\{b\}\{c\}, \{a\}\{c\}\{b\}, \{a, c\}\{b\}\}$, has two canonical representations $\{a, b\}\{c\}$ and $\{a, c\}\{b\}$. One can easily check that this absorbing monoid is not a monoid of comtraces. □

The canonical representation is also not unique for generalised comtraces if $inl \neq \emptyset$. For any generalised comtrace, if $\{a, b\} \subseteq E$, $(a, b) \in inl$, then $x = [\{a\}\{b\}] = \{\{a\}\{b\}, \{b\}\{a\}\}$ and x has two canonical representations $\{a\}\{b\}$ and $\{b\}\{a\}$.

All the canonical representations discussed above can be extended to unique canonical representations by simply introducing some total order on step sequences, and adding a minimality requirement with respect to this total order to the definition of a canonical form. The technique used in the definition of Foata normal form is one possibility. However this has nothing to do with any property of concurrency and hence will not be discussed in this paper.

However the comtraces have a unique canonical representation as defined above. This was not proved in [10] and will be proved in the next section.

5 Canonical Representations of Comtraces

In principle the uniqueness of canonical representation for comtraces follows the fact that all equations can be derived from the properties of pairs of events. This results in very strong cancellation and projection properties, and very regular structure of the set of all steps $\mathcal{S}$.

Let $a \in E$ and $w \in \mathcal{S}^*$. We can define a *right cancellation* operator $\div_R$ as

$$\lambda \div_R a = \lambda, \qquad wA \div_R a = \begin{cases} (w \div_R a)A & \text{if } a \notin A \\ w & \text{if } A = \{a\} \\ w(A \setminus \{a\}) & \text{otherwise.} \end{cases}$$

Symmetrically, a *left cancellation* operator $\div_L$ is defined as

$$\lambda \div_L a = \lambda, \qquad Aw \div_L a = \begin{cases} A(w \div_L a) & \text{if } a \notin A \\ w & \text{if } A = \{a\} \\ (A \setminus \{a\})w & \text{otherwise.} \end{cases}$$

Finally, for each $D \subseteq E$, we define the function $\pi_D : \mathcal{S}^* \to \mathcal{S}^*$, *step sequence projection* onto D, as follows:

$$\pi_D(\lambda) = \lambda, \qquad \pi_D(wA) = \begin{cases} \pi_D(w) & \text{if } A \cap D = \emptyset \\ \pi_D(w)(A \cap D) & \text{otherwise.} \end{cases}$$

Proposition 3.

1. $u \equiv v \implies u \div_R a \equiv v \div_R a$. (right cancellation)
2. $u \equiv v \implies u \div_L a \equiv v \div_L a$. (left cancellation)
3. $u \equiv v \implies \pi_D(u) \equiv \pi_D(v)$. (projection rule)

Proof. For each step sequence $t = A_1 \ldots A_k \in \mathcal{S}^*$ let $\Sigma(t) = \bigcup_{i=1}^{k} A_i$. Note that for comtraces $u \approx v$ means $u = xAy$, $v = xBCy$, where $A = B \cup C$, $B \cap C = \emptyset$, $B \times C \subseteq ser$.

1. It suffices to show that $u \approx v \implies u \div_R a \approx v \div_R a$. There are four cases:
   (a) $a \in \Sigma(y)$. Let $z = y \div_R a$. Then $u \div_R a = xAz \approx xBCz = v \div_R a$.
   (b) $a \notin \Sigma(y)$, $a \in A \cap C$. Then $u \div_R a = x(A \setminus \{a\})y \approx xB(C \setminus \{a\})y = v \div_R a$.
   (c) $a \notin \Sigma(y)$, $a \in A \cap B$. Then $u \div_R a = x(A \setminus \{a\})y \approx x(B \setminus \{a\})Cy = v \div_R a$.
   (d) $a \notin \Sigma(Ay)$. Let $z = x \div_R a$. Then $u \div_R a = zAy \approx zBCy = v \div_R a$.
2. Dually to (1).
3. It suffices to show that $u \approx v \implies \pi_D(u) \approx \pi_D(v)$. Note that $\pi_D(A) = \pi_D(B) \cup \pi_D(C)$, $\pi_D(B) \cap \pi_D(C) = \emptyset$ and $\pi_D(B) \times \pi_D(C) \subseteq ser$, so $\pi_D(u) = \pi_D(x)\pi_D(A)\pi_D(y) \approx \pi_D(x)\pi_D(B)\pi_D(C)\pi_D(y) = \pi_D(v)$. □

Proposition 3 does not hold for an arbitrary absorbing monoid. For the absorbing monoid from Example 2 we have $u = \{a,b,c\} \equiv v = \{a\}\{b,c\}$, $u \div_R b = u \div_L b = \pi_{\{a,c\}}(u) = \{a,c\} \not\equiv \{a\}\{c\} = v \div_R b = v \div_L b = \pi_{\{a,c\}}(v)$.

Note that $(w \div_R a) \div_R b = (w \div_R b) \div_R a$, so we can define

$$w \div_R \{a_1, ..., a_k\} \stackrel{df}{=} (...((w \div_R a_1) \div_R a_2)...) \div_R a_k, \text{ and}$$
$$w \div_R A_1...A_k \stackrel{df}{=} (...((w \div_R A_1) \div_R A_2)...) \div_R A_k.$$

We define dually for $\div_L$.

Corollary 2. *For all $u, v, x \in \mathcal{S}$, we have*

1. $u \equiv v \implies u \div_R x \equiv v \div_R x$.
2. $u \equiv v \implies u \div_L x \equiv v \div_L x$. □

The uniqueness of canonical representation for comtraces follows directly from the following result.

Lemma 1. *For each canonical step sequence $u = A_1 \dots A_k$, we have*

$$A_1 = \{a \mid \exists w \in [u].\ w = C_1 \dots C_m \wedge a \in C_1\}.$$

Proof. Let $A = \{a \mid \exists w \in [u].\ w = C_1 \dots C_m \wedge a \in C_1\}$. Since $u \in [u]$, $A_1 \subseteq A$. We need to prove that $A \subseteq A_1$. Definitely $A = A_1$ if $k = 1$, so assume $k > 1$. Suppose that $a \in A \setminus A_1$, $a \in A_j$, $1 < j \le k$ and $a \notin A_i$ for $i < j$. Since $a \in A$, there is $v = Bx \in [u]$ such that $a \in B$. Note that $A_{j-1}A_j$ is also *canonical* and $u' = A_{j-1}A_j = (u \div_R (A_{j+1}...A_k)) \div_L (A_1...A_{j-2})$. Let $v' = (v \div_R (A_{j+1}...A_k)) \div_L (A_1...A_{j-2})$. We have $v' = B'x'$ where $a \in B'$. By Corollary 2, $u' \equiv v'$. Since $u' = A_{j-1}A_j$ is canonical then $\exists c \in A_{j-1}.\ (c,a) \notin ser$ or $\exists b \in A_j.\ (a,b) \notin ser$. For the former case: $\pi_{\{a,c\}}(u') = \{c\}\{a\}$ (if $c \notin A_j$) or $\pi_{\{a,c\}}(u') = \{c\}\{a,c\}$ (if $c \in A_j$). If $\pi_{\{a,c\}}(u') = \{c\}\{a\}$ then $\pi_{\{a,c\}}(v')$ equals either $\{a,c\}$ (if $c \in B'$) or $\{a\}\{c\}$ (if $c \notin B'$), i.e. in both cases $\pi_{\{a,c\}}(u') \not\equiv \pi_{\{a,c\}}(v')$, contradicting Proposition 3(3). If $\pi_{\{a,c\}}(u') = \{c\}\{a,c\}$ then $\pi_{\{a,c\}}(v')$ equals either $\{a,c\}\{c\}$ (if $c \in B'$) or $\{a\}\{c\}\{c\}$ (if $c \notin B'$). However in both cases $\pi_{\{a,c\}}(u') \not\equiv \pi_{\{a,c\}}(v')$, contradicting Proposition 3(3). For the latter case, let $d \in A_{j-1}$. Then $\pi_{\{a,b,d\}}(u') = \{d\}\{a,b\}$ (if

$d \notin A_j$), or $\pi_{\{a,b,d\}}(u') = \{d\}\{a,b,d\}$ (if $d \in A_j$). If $\pi_{\{a,b,d\}}(u') = \{d\}\{a,b\}$ then $\pi_{\{a,b,d\}}(v')$ is one of the following $\{a,b,d\}$, $\{a,b\}\{d\}$, $\{a,d\}\{b\}$, $\{a\}\{b\}\{d\}$ or $\{a\}\{d\}\{b\}$, and in either case $\pi_{\{a,b,d\}}(u') \not\equiv \pi_{\{a,b,d\}}(v')$, again contradicting Proposition 3(3). If $\pi_{\{a,b,d\}}(u') = \{d\}\{a,b,d\}$ then $\pi_{\{a,b,d\}}(v')$ is one of the following $\{a,b,d\}\{d\}$, $\{a,b\}\{d\}\{d\}$, $\{a,d\}\{b,d\}$, $\{a,d\}\{b\}\{d\}$, $\{a,d\}\{d\}\{b\}$, $\{a\}\{b\}\{d\}\{d\}$, $\{a\}\{d\}\{b\}\{d\}$, or $\{a\}\{d\}\{d\}\{b\}$. However in either case we have $\pi_{\{a,b,d\}}(u') \not\equiv \pi_{\{a,b,d\}}(v')$, contradicting Proposition 3(3) as well. □

The above lemma does not hold for an arbitrary absorbing monoid. For both canonical representations of $[\{a,b\}\{c\}]$ from Example 9, namely $\{a,b\}\{c\}$ and $\{a,c\}\{b\}$, we have $A = \{a \mid \exists w \in [u].\ w = C_1 \dots C_m \wedge a \in C_1\} = \{a,b,c\} \notin \mathcal{S}$. Adding A to $\mathcal{S}$ does not help as we still have $A \neq \{a,b\}$ and $A \neq \{a,c\}$.

Theorem 4. *For every comtrace $t \in \mathcal{S}^*/\equiv$ there exists exactly one canonical step sequence u such that $t = [u]$.*

Proof. The existence follows from Theorem 3. Suppose that $u = A_1 \dots A_k$ and $v = B_1 \dots B_m$ are both canonical step sequences and $u \equiv v$. By Lemma 1, we have $B_1 = A_1$. If $k = 1$, this ends the proof. Otherwise, let $u' = A_2 \dots A_k$ and $v' = B_2 \dots B_m$. By Corollary 2(2) we have $u' \equiv v'$. Since u' and v' are also canonical, by Lemma 1, we have $A_2 = B_2$, etc. Hence $u = v$. □

6 Relational Representation of Traces, Comtraces and Generalised Comtraces

It is widely known that Mazurkiewicz traces can represent partial orders. We show the similar relational relational equivalence for both comtraces and generalised comtraces.

6.1 Partial Orders and Mazurkiewicz Traces

Each trace can be interpreted as a partial order and each finite partial order can be represented by a trace. Let $t = \{x_1, ..., x_k\}$ be a trace, and let $\prec_{x_i}$ be a total order defined by a sequence x_i, $i = 1, ..., k$. The partial order generated by the trace t can then be defined as: $\prec_t = \bigcap_{i=1}^{k} \prec_{x_i}$. Moreover, the set $\{\prec_{x_1}, ..., \prec_{x_n}\}$ is the set of all total extensions of $\prec_t$. Let X be a finite set, $\prec \subset X \times X$ be a partial order, $\{\prec_1, ..., \prec_k\}$ be the set of all total extensions of $\prec$, and let $x_{\prec_i} \in X^*$ be a sequence that represents $\prec_i$, for $i = 1, ..., k$. The set $\{x_{\prec_1}, ..., x_{\prec_k}\}$ is a trace over the concurrent alphabet $(X, \frown_{\prec})$.

6.2 Stratified Order Structures and Comtraces

Mazurkiewicz traces can be interpreted as a formal language representation of finite partial orders. In the same sense comtraces can be interpreted as a formal language representation of finite *stratified order structures*. Partial orders can adequately model "earlier than" relationship but cannot model "not later than" relationship [9]. Stratified order structures are pairs of relations and can model "earlier than" and "not later than" relationships.

A *stratified order structure* is a triple $Sos = (X, \prec, \sqsubset)$, where X is a set, and $\prec$, $\sqsubset$ are binary relations on X that satisfy the following conditions:

C1:	$a \not\sqsubset a$	C3:	$a \sqsubset b \sqsubset c \wedge a \neq c \implies a \sqsubset c$
C2:	$a \prec b \implies a \sqsubset b$	C4:	$a \sqsubset b \prec c \vee a \prec b \sqsubset c \implies a \prec c$

C1–C4 imply that $\prec$ is a partial order and $a \prec b \Rightarrow b \not\sqsubset a$. The relation $\prec$ is called "causality" and represents the "earlier than" relationship while $\sqsubset$ is called "weak causality" and represents the "not later than" relationship. The axioms C1–C4 model the mutual relationship between "earlier than" and "not later than" provided the system runs are defined as stratified orders.

Stratified order structures were independently introduced in [6] and [8] (the defining axioms are slightly different from C1–C4, although equivalent). Their comprehensive theory has been presented in [11]. It was shown in [10] that each comtrace defines a finite stratified order structure. The construction from [10] did not use the results of [11]. In this paper we present a construction based on the results of [11], which will be intuitively closer to the one used to show the relationship between traces and partial orders in Section 6.1.

Let $Sos = (X, \prec, \sqsubset)$ be a stratified order structure. A *stratified* order $\lhd$ on X is an *extension* of Sos if for all $a, b \in X$, $a \prec b \implies a \lhd b$ and $a \sqsubset b \implies a \lhd^{\frown} b$. Let $ext(Sos)$ denote the set of all extensions of Sos.

Let $u = A_1 \ldots A_k$ be a step sequence. By $\overline{u} = \overline{A}_1 \ldots \overline{A}_k$ be the *event enumerated* representation of t. We will skip a lengthy but intuitively obvious formal definition (see for example [10]), but for instance if $u = \{a,b\}\{b,c\}\{c,a\}\{a\}$, then $\overline{u} = \{a^{(1)}, b^{(1)}\}\{b^{(2)}, c^{(1)}\}\{a^{(2)}, c^{(2)}\}\{a^{(3)}\}$. Let $\Sigma_u = \bigcup_{i=1}^{k} \overline{A}_i$ denote the set of all enumerated events occurring in u, for $u = \{a,b\}\{b,c\}\{c,a\}\{a\}$, $\Sigma_u = \{a^{(1)}, a^{(2)}, a^{(3)}, b^{(1)}, b^{(2)}, c^{(1)}, c^{(2)}\}$. For each $\alpha \in \Sigma_u$, let $pos_u(\alpha)$ denote the consecutive number of a step where α belongs, i.e. if $\alpha \in \overline{A}_j$ then $pos_u(\alpha) = j$. For our example $pos_u(a^{(2)}) = 3$, $pos_u(b^{(2)}) = 2$, etc. For each enumerated even $\alpha = e^{(i)}$, let $l(\alpha)$ denote the *label* of α, i.e. $l(\alpha) = l(e^{(i)}) = e$. One can easily show ([10]) that $u \equiv v \implies \Sigma_u = \Sigma_v$, so we can define $\Sigma_{[u]} = \Sigma_u$.

Given a step sequence u, we define a stratified order $\lhd_u$ on Σ_u by: $\alpha \lhd_u \beta \iff pos_u(\alpha) < pos_u(\beta)$. Conversely, let $\lhd$ be a stratified order on a set X. The set X can be partitioned into a unique sequence of non-empty sets $\Omega_\lhd = B_1 \ldots B_k$ ($k \geq 0$) such that

$$\lhd = \bigcup_{i<j} (B_i \times B_j) \quad \text{and} \quad \simeq_\lhd = \bigcup_i (B_i \times B_i).$$

Unfortunately the proofs of two theorems below require introducing additional concepts and results, so we only provide sketches.

Theorem 5. *Let t be a comtrace over* (E, sim, ser) *and let* $\prec_t$, $\sqsubset_t$ *be two binary relations on* Σ_t *defined as:*

$$\alpha \prec_t \beta \iff \forall u \in t.\ \alpha \lhd_u \beta,$$
$$\alpha \sqsubset_t \beta \iff \forall u \in t.\ \alpha \lhd_u^{\frown} \beta.$$

We have:

1. $Sos_t = (\Sigma_t, \prec_t, \sqsubset_t)$ *is a stratified order structure,*
2. $ext(Sos_t) = \{\lhd_u \mid u \in t\}$.

Proof (Sketch). The main part is to show that each stratified order $\lhd$ on Σ_t that satisfies: $\alpha \prec_t \beta \implies \alpha \lhd \beta$ and $\alpha \sqsubset_t \beta \implies \alpha \lhd^{\frown} \beta$ belongs to $\{\lhd_u \mid u \in t\}$. This can be done by induction on the number of steps of w, where w is the canonical step sequence such that $[w] = t$. The rest is a consequence of the results of [9, 11]. □

Theorem 6. *Let* $Sos = (X, \prec, \sqsubset)$ *be a stratified order structure, and let* $\Delta = \{\Omega_{\lhd} \mid \lhd \in ext(Sos)\}$. *Let relations* $sim, ser \subseteq X \times X$ *be defined as follows:*

- $(\alpha, \beta) \in sim \iff \alpha \frown_{\prec} \beta$,
- $(\alpha, \beta) \in ser \iff (\alpha, \beta) \in sim \ \wedge\ (\beta \not\sqsubset \alpha)$.

Then we have:

1. $\theta = (E, sim, ser)$ *is a comtrace concurrent alphabet,*
2. *for each* $u, v \in \Delta$ *we have* $u \equiv v$, *i.e.* Δ *is a comtrace over the alphabet* θ.

Proof (Sketch). (1) is straightforward. To prove (2) we first take the canonical stratified extension of $\prec$ (see [10]), show that it belongs to $ext(Sos)$, and then show that it represents a canonical step sequence. Next we prove (2) by induction on the number of steps of this canonical step sequence. □

6.3 Generalised Stratified Order Structures and Generalised Comtraces

A *generalised stratified order structure* is a triple $GSos = (X, <>, \sqsubset)$, where X is a non-empty set, and $<>$, $\sqsubset$ are two irreflexive relations on X, $<>$ is symmetric, and the triple $(X, \prec_G, \sqsubset)$, where $\prec_G = <> \cap \sqsubset$, is a stratified order structure (i.e. it satisfies C1–C4 from the previous subsection).

The relation $<>$ is called "commutativity" and represents the "earlier than or later than" relationship, while $\sqsubset$, called "weak causality" represents "not later than" relationship.

Generalised stratified order structures were introduced and their comprehensive theory has been presented in [7]. They can model any concurrent history when runs or observations are modelled by stratified orders (see [7]). We will show that each generalised comtrace defines a finite generalised stratified order structure and that each finite generalised stratified order structure can be represented by a generalised comtrace.

Let $GSos = (X, \Leftrightarrow, \sqsubset)$ be a generalised stratified order structure. A *stratified* order $\lhd$ on X is an *extension* of $GSos$ if for all $a, b \in X$, $a \Leftrightarrow b \implies a \lhd b$ or $b \lhd a$, and $a \sqsubset b \implies a \lhd^{\frown} b$. Let $ext(GSos)$ denote the set of all extensions of Sos.

Again the proofs of two theorems below require introducing additional concepts and results, so we only provide sketches.

Theorem 7. *Let t be a generalised comtrace over $(\mathcal{S}, sim, ser, inl)$ and let $\Leftrightarrow_t$, $\sqsubset_t$ be two binary relations on Σ_t defined as:*

$$\alpha \Leftrightarrow_t \beta \iff \forall u \in t.\ (\alpha \lhd_u \beta \vee \beta \lhd_u \alpha)\ ,$$
$$\alpha \sqsubset_t \beta \iff \forall u \in t.\ \alpha \lhd_u^{\frown} \beta.$$

We have:

1. *$GSos_t = (\Sigma_t, \Leftrightarrow_t, \sqsubset_t)$ is a generalised stratified order structure,*
2. *$ext(GSos_t) = \{\lhd_u \mid u \in t\}$.*

Proof (Sketch). The main part is to show that each stratified order $\lhd$ on Σ_t that satisfies: $\alpha \Leftrightarrow_t \beta \implies \alpha \lhd \beta \vee \beta \lhd \alpha$ and $\alpha \sqsubset_t \beta \implies \alpha \lhd^{\frown} \beta$ belongs to $\{\lhd_u \mid u \in t\}$. This can be done by induction on the number of steps of w, where w is the canonical step sequence such that $[w] = t$ (we do not need a canonical representation to be unique here). The rest follows from the results of [7, 11]. □

Theorem 8. *Let $Sos = (X, \Leftrightarrow, \sqsubset)$ be a generalised stratified order structure, and let $\Delta = \{\Omega_{\lhd} \mid \lhd \in ext(GSos)\}$. Let relations $sim, ser, inl \subseteq X \times X$ be defined as follows:*

- *$(\alpha, \beta) \in sim \iff \neg(\alpha \Leftrightarrow \beta)$,*
- *$(\alpha, \beta) \in ser \iff \neg(\alpha \Leftrightarrow \beta) \ \wedge\ \beta \not\sqsubset \alpha$,*
- *$(\alpha, \beta) \in inl \iff \alpha \Leftrightarrow \beta \ \wedge\ \neg(\alpha \sqsubset \beta \ \vee\ \beta \sqsubset \alpha)$.*

Then we have:

1. *$\theta = (E, sim, ser, inl)$ is a generalised comtrace concurrent alphabet,*
2. *for each $u, v \in \Delta$ we have $u \equiv v$, i.e. Δ is a generalised comtrace over the generalised comtrace alphabet θ.*

Proof (Sketch). (1) is straightforward. The proof of (2) is more complex than the proof of (2) of Theorem 6, as we need to show that there is a stratified order in $ext(GSos)$ which can be represented as as an appropriate canonical step sequence (no uniqueness needed). Next we prove (2) by induction on the number of steps of this canonical step sequence. □

7 Paradigms of Concurrency

The general theory of concurrency developed in [9] provides a hierarchy of models of concurrency, where each model corresponds to a so called "paradigm", or a general rule about the structure of concurrent histories, where concurrent histories are defined as sets of equivalent partial orders representing particular system runs. In principle, a paradigm describes how simultaneity is handled in concurrent histories. The paradigms are denoted by π_1 through π_8. It appears that only paradigms π_1, π_3, π_6 and π_8 are interesting from the point of view of concurrency theory. The paradigms were formulated in terms of partial orders. Comtraces are sets of step sequences, each step sequence uniquely defines a stratified order, so the comtraces can be interpreted as sets of equivalent partial orders, i.e. concurrent histories (see [10] for details). The most general paradigm, π_1, assumes no additional restrictions for concurrent histories, so each comtrace conforms trivially to π_1. The paradigms π_3, π_6 and π_8, when translated into the comtrace formalism, impose the following restrictions.

Let (E, sim, ser, inl) be a generalised comtrace alphabet. The monoid of generalised comtraces (comtraces when $inl = \emptyset$) $(\mathcal{S}^*/\!\equiv, \hat{\circ}, [\lambda])$ conforms to:

$$\text{paradigm } \pi_3 \iff \forall a, b \in E.\ (\{a\}\{b\} \equiv \{b\}\{a\} \Rightarrow \{a, b\} \in \mathcal{S}).$$
$$\text{paradigm } \pi_6 \iff \forall a, b \in E.\ (\{a, b\} \in \mathcal{S} \Rightarrow \{a\}\{b\} \equiv \{b\}\{a\}).$$
$$\text{paradigm } \pi_8 \iff \forall a, b \in E.\ (\{a\}\{b\} \equiv \{b\}\{a\} \Leftrightarrow \{a, b\} \in \mathcal{S}).$$

Proposition 4. *1. Every monoid of comtraces conforms to π_3.*
2. If π_8 is satisfied then $ind = ser = sim$.

Proof. 1. Let $\{a\}\{b\} \equiv \{b\}\{a\}$ for some $a, b \in E$. This means $\{a\}\{b\} \approx^{-1} \{a, b\} \approx \{b\}\{a\}$, i.e. $\{a, b\} \in \mathcal{S}$.
2. Clearly $ind \subseteq ser \subseteq sim$. Let $(a, b) \in sim$. This means $\{a, b\} \in \mathcal{S}$, which, by π_8, implies $\{a\}\{b\} \equiv \{b\}\{a\}$, i.e. $(a, b) \in ind$. □

From Proposition 4 it follows that comtraces cannot model any concurrent behaviour (history) that does not conform to the paradigm π_3. Generalised comtraces conform only to π_1, so they can model any concurrent history that is represented by a set of equivalent step sequences.

If a monoid of comtraces conforms to π_6 it also conforms to π_8. Proposition 4 says all comtraces conforming to π_8 can be reduced to equivalent Mazurkiewicz traces.

8 Conclusion

The concepts of absorbing monoids over step sequences, partially commutative absorbing monoids over step sequences, absorbing monoids with compound generators, and monoids of generalised comtraces have been introduced and analysed. They all are generalisations of Mazurkiewicz traces [5] and comtraces [10].

Some new properties of comtraces and their relationship to stratified order structures [11] have been discussed. The relationship between generalised comtraces and generalised stratified order structures [7] was also analysed.

Despite some obvious advantages, for instance, very handy composition and no need to use labels, quotient monoids (perhaps with the exception of Mazurkiewicz traces) are much less popular in dealing with issues of concurrency than their relational counterparts partial orders, stratified order structures, occurrence graphs, etc. We believe that in many cases, quotient monoids could provide simpler and more adequate models of concurrent histories than their relational equivalences.

Acknowledgement. The authors thanks all four referees for a number of very detailed and helpful comments.

References

1. P. Cartier and D. Foata, *Problèmes combinatoires de commutation et réarrangements*, *Lecture Notes in Mathematics* 85, Springer 1969.
2. P. M. Cohn, *Universal Algebra*, D. Reidel 1981.
3. R. Davillers, R. Janicki, M. Koutny, P. E. Lauer, Concurrent and Maximally Concurrent Evolution of Non-sequential Systems, *Theoretical Computer Science* 43:213-238, 1986.
4. J. Desel, Private Information, Communicated to the authors by G. Juhás, 2007.
5. V. Diekert and G. Rozenberg (eds.), *The Book of Traces.* World Scientific 1995.
6. H. Gaifman and V. Pratt, Partial Order Models of Concurrency and the Computation of Function, *Proc. of LICS'87*, pp. 72-85.
7. G. Guo, R. Janicki, Modelling Concurrent Behaviours by Commutativity and Weak Causality Relations, Proc. of AMAST'02, *Lecture Notes in Computer Science* 2422 (2002), 178-191.
8. R. Janicki and M. Koutny, Invariants and Paradigms of Concurrency Theory, *Lecture Notes in Computer Science* 506, Springer 1991, pp. 59-74.
9. R. Janicki and M. Koutny, Structure of Concurrency, *Theoretical Compututer Science*, 112(1):5–52, 1993.
10. R. Janicki and M. Koutny, Semantics of Inhibitor Nets, *Information and Computation*, 123(1):1–16, 1995.
11. R. Janicki and M. Koutny, Fundamentals of Modelling Concurrency Using Discrete Relational Structures, *Acta Informatica*, 34:367–388, 1997.
12. G. Juhás, R. Lorenz, S. Mauser, Synchronous + Concurrent + Sequential = Earlier Than + Not Later Than, *Proc. of ACSD'06* (Application of Concurrency to System Design), Turku, Finland 2006, pp. 261-272, IEEE Press.
13. H. C. M. Kleijn, M. Koutny, Process Semantics of General Inhibitor Nets, *Information and Computation*, 190:18–69, 2004.
14. A. Mazurkiewicz, Introduction to Trace Theory, in [5], pp. 3–42.
15. E. Ochmański, Recognizable Trace Languages, in [5], pp. 167-204.
16. M. W. Shields, Adequate Path Expressions, *Lecture Notes in Computer Science* 70, Springer 1979, pp. 249-265.