Studies in Comtrace Monoids

Studies in Comtrace Monoids

By

Dai Tri Man Lê, B. Sc.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree of

Master of Computer Science
Department of Computing and Software
McMaster University

MASTER OF COMPUTER SCIENCE (2008)
(Department of Computing and Software)

McMaster University
Hamilton, Ontario

TITLE: Studies in Comtrace Monoids

AUTHOR: Dai Tri Man Lê, B. Sc. (McMaster University)

SUPERVISOR: Dr. Ryszard Janicki

NUMBER OF PAGES: vii, 121

Abstract

Mazurkiewicz traces were introduced by A. Mazurkiewicz in 1977 as a language representation of partial orders to model "true concurrency". The theory of Mazurkiewicz traces has been utilised to tackle not only various aspects of concurrency theory but also problems from other areas, including combinatorics, graph theory, algebra, and logic.

However, neither Mazurkiewicz traces nor partial orders can model the "not later than" relationship. In 1995, *comtraces* (*com*bined *traces*) were introduced by Janicki and Koutny as a formal language counterpart to *finite stratified order structures.* They show that each comtrace uniquely determines a finite stratified order structure, yet their work contains very little theory of comtraces.

This thesis aims at enriching the tools and techniques for studying the theory of comtraces.

Our first contribution is to introduce the notions of *absorbing monoids*, *generalised comtrace monoids*, *partially commutative absorbing monoids*, and *absorbing monoids with compound generators*, all of which are the generalisations of Mazurkiewicz trace and comtrace monoids. We also define and study the canonical representations of these monoids.

Our second contribution is to define the notions of *non-serialisable steps* and utilise them to study the construction which Janicki and Koutny use to build stratified order structures from comtraces. Moreover, we show that any finite stratified order structure can be represented by a comtrace.

Our third contribution is to study the relationship between *generalised comtraces* and *generalised stratified order structures.* We prove that each generalised comtrace uniquely determines a finite generalised stratified order structure.

Acknowledgements

My profound gratitude to my supervisor, Dr. Ryszard Janicki, for his thoughtful guidance, constructive criticism, and mentorship from the publishing my very first paper until the completion of this Masters' thesis.

I am deeply indebted to my mentor, Dr. Gregory H. Moore, for his wonderful courses which inspired me to do research relating to set theory and mathematical logic. Without him, I would never think of going to graduate school.

My appreciation to Dr. Jeffery Zucker, an excellent teacher, for his significant encouragement and the books he generously lent me. His lucid and humorous teaching style affects my conducting tutorials a great deal.

My thanks to Dr. Michael Soltys, for the knowledge elicited from in his extremely tough course on Complexity Theory.

I am grateful to Dr. Gregory H. Moore and the Master's Thesis committee members, Drs. Jeffery Zucker and Michael Soltys, for carefully reading the manuscript and for all the useful suggestions which led to further improvements of this thesis.

Many thanks to Drs. Christopher Anand and Jacques Carette, who introduced me into research activities during my summer work in 2003 and 2004, and Dr. Emil Sekerinski, who introduced me to Formal Methods and "the elements of style" when writing papers.

I am grateful to my wonderful professors, Antoine Deza, Franya Franek, Tom Maibaum, Sanzheng Qiao, William Smyth, Patrick Speissegger, Tamas Terlaky, and Alan Wassyng, who gave me precious lessons and paved the way for the completion of my thesis.

I truly appreciate the excellent service provided by the departmental administrator, Laurie Leblanc, and departmental secretaries, Daphne Kilgour, Tina Macala, Catherine Roberts and *Jessica Stewart.*

My thanks to Nadya Zubkova, who encouraged me in my writing of this thesis, and Marek Janicki, who proofread several of my papers.

I highly appreciate the support of the Ashbaugh Graduate Scholarship (2006–2007) and an Ontario Graduate Scholarship (2007–2008), which have played a significant role in my study as well as my life.

Finally, I would like to dedicate this thesis and all of my academic achievements to my parents and my sisters to express my eternal gratitude for their sacrifices, love, encouragement, and financial support for the last years.

> In the beginner's mind there are many possibilities, in the expert's mind there are few. – Shunryu Suzuki

> I have made this letter longer than usual, because I lack the time to make it short. – Blaise Pascal

Contents

Chapter 1

Introduction

Mazurkiewicz traces or partially commutative monoids [1, 24, 8] are quotient monoids over sequences (or words). The theory of traces has been utilised to tackle problems from quite diverse areas including combinatorics, graph theory, algebra, logic and especially concurrency theory [8].

As a language representation of partial orders, they can sufficiently model "true concurrency" in various aspects of concurrency theory. However, the basic monoid for Mazurkiewicz traces, 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 is a limitation when the generators have some internal structure; for instance, when they are sets, their internal structure may be used to define the set of equations that generate the quotient monoid. In this paper, we assume that the monoid generators have some internal structure. We call such generators *compound*, and then use the properties of that internal structure to define an appropriate quotient congruence.

Another limitation of Mazurkiewicz traces and their generated partial orders is that neither Mazurkiewicz traces nor partial orders can model the "not later than" relationship [13]. If an event a is performed "not later than" an event b, where 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 $x = \{\{a\}\{b\}, \{a, b\}\}$. But the set x 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.

To overcome those limitations the concept of a *comtrace* (*com*bined *trace*) was introduced in [14]. Comtraces are finite sets of equivalent step sequences and the congruence is determined by a relation *ser*, which is called *serialisability* and in general is 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, [14] contains very little theory of comtraces, only their relationship to stratified order structures has been considerably developed.

Stratified order structures [9, 12, 14, 15] 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 [14, 18, 20] and others). It was shown in [14] that each comtrace defines a finite stratified order structure. However, the comtraces are so far much less often used 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 sufficient theory development of quotient monoids different from that of Mazurkiewicz traces.

Both comtraces and stratified order structures can adequately model concurrent histories only when the paradigm π_3 of [13, 15] is satisfied. For the general case, we need *generalised stratified order structures*, introduced and analysed in [10]. 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 thesis, a sequence counterpart of generalised stratified order structures, called *generalised comtraces*, are introduced and their properties are discussed.

It appears comtraces and generalised comtraces are special cases of two more general classes of quotient monoids, which we call *absorbing monoids* and *partially* *commutative absorbing monoids* respectively. For these classes of 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.*

This thesis is the expansion and revision of our previous work in [17], where [17, Theorem 9.1], [17, Theorem 9.2], [17, Theorem 10.1] and some new major properties are fully proved and analysed. The content of the thesis is organised as following. In the next chapter, we review the basic concepts of order theory, which includes the important Szpilrajn Theorem [31], and monoids theory. Chapter 3 introduces *equational monoids with compound generators* and other types of monoids that are discussed in this thesis. In Chapter 4 the canonical representations of absorbing monoids, partially commutative absorbing monoids and absorbing monoids with compound generators are defined and briefly analysed. In Chapter 5, we introduce some basic algebraic operations on step sequences and utilise them to prove some properties of comtrace congruence and to give a new version of the proof that canonical representation for comtraces is *unique.* Chapter 6 studies some basic properties of comtrace languages. Chapter 7 reviews different paradigms of concurrent histories and discuss how comtraces and generalised comtraces are classified with respect to these paradigms. Chapter 8 surveys some basic background on relational structures model of concurrency [9, 12, 14, 15, 10, 11] to prepare the readers for the chapters followed. In Chapter 9, we introduce the notions of *non-serialisable steps* to study the construction from comtraces to finite stratified order structures by Janicki and Koutny in [14]; we then prove that any finite stratified order structure can be represented by a comtrace. In Chapter 10, analogous to the notion of $\diamond$-closure which Janicki and Koutny used to construct stratified order structures from comtraces, we define the notion of *commutative closure* and utilise it to construct generalised stratified order structures from comtraces; we prove that each generalised comtrace can be represented by a finite generalised stratified order structure. Chapter 11 contains some final discussion and comments on our future works.

Chapter 2

Background

2.1 Orders

In this section, we survey some standard order-theoretic definitions and results which are used intensively in this thesis.

2.1.1 Equivalence Relation

Let X be a set and I is an index set. The family of sets $\{A_i \mid i \in I\}$ is called a *partition* of X if and only if

1. $A_i \neq \emptyset$ for all i,

2. $A_i \cap A_j = \emptyset$ for all $i \neq j$, and

3. $X = \bigcup_{i \in I} A_i$.

We can observe that $\{\{x\} \mid x \in X\}$ (the set of all possible singletons of X) is the finest partition possible of the set X.

An *equivalence relation* R on a set X is reflexive, symmetric and transitive binary relation on X. In other words, the following must hold for all $a, b, c \in X$:

1. $a \; R \; a$, *(reflexive)*

2. $a \; R \; b \;\Rightarrow\; b \; R \; a$, *(symmetric)*

3. $a\,R\,b\,Rc \Rightarrow a\,R\,c.$ *(transitive)*

For every $x \in X$, the set $[x]_R = \{y \mid y\ R\ x \land y \in X\}$ is the equivalence class of x with respect to R. We drop the subscript and write $[x]$ to denote the equivalence class of x when R is clear from the context. The set X equipped with an equivalence relation R is called a *setoid*.

Proposition 2.1. *Let $R \subseteq X \times X$ be an equivalence relation on X. If $a, b \in X$, the following are equivalent:*

1. $a\,R\,b$

2. $[a] = [b]$

3. $[a] \cap [b] \neq \emptyset$

Proof.

- (1)⇒(2): Assume that $a\,R\,b$, since it also implies $b\,R\,a$ (by symmetry), it suffices to show $[a] \subseteq [b]$. For any $c \in [a] = \{x \mid x\,R\,a \land x \in X\}$, it follows that $c\,R\,a$. Since $a\,R\,b$, we have $c\,R\,b$ (by transitivity). Hence,

$$c \in [b] = \{x \mid x\,R\,b \land x \in X\}.$$

- (2)⇒(3): Since $[a] = [b]$, it follows that $a \in [a] \cap [b]$. Hence, $[a] \cap [b] \neq \emptyset$.

- (3)⇒(1): Since $[a] \cap [b] \neq \emptyset$, there exist some $c \in [a] \cap [b]$. Since $c \in [a]$ and $c \in [b]$, we have $c\,R\,a$ and $c\,R\,b$. By reflexivity we have $a\,R\,c$ and by transitivity we have $a\,R\,b$ as desired.

□

Corollary 2.1. *If R is an equivalence relation on X and $a, b \in X$, then*

$$(a, b) \notin R \iff [a] \cap [b] = \emptyset$$

Proof. From Proposition 2.1, we already have

$$(a, b) \in R \iff [a] \cap [b] \neq \emptyset.$$

This is logically equivalent to

$$(a, b) \notin R \iff [a] \cap [b] = \emptyset.$$

□

For every equivalence relation $R \subseteq X \times X$, we define $X/R \overset{df}{=} \{[a]_R \mid a \in X\}$. Clearly X/R is the set of all equivalence classes of R on X.

Proposition 2.2. *For every equivalence relation $R \subseteq X \times X$, X/R is a partition of the set X.*

Proof. From Corollary 2.1 we already know any two distinct equivalence classes are disjoint. It suffices to show $X = \bigcup_{A \in X/R} A$. But $\bigcup_{A \in X/R} A \subseteq X$ since $A \subseteq X$ for any $A \in X/R$. It remains to show $X \subseteq \bigcup_{A \in X/R} A$. But for any $x \in X$, $[x] \in X/R$ and hence $x \in \bigcup_{A \in X/R} A$. □

2.1.2 Partial Order

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

1. $\neg(a \prec a)$, *(irreflexive)*
2. $a \prec b \prec c \Rightarrow a \prec c$. *(transitive)*

The pair $(X, \prec)$ in this case is called a *partially ordered set* (also called a *poset*), i.e. the set X is partially ordered by the relation $\prec$. The pair $(X, \prec)$ is called a *finite partially ordered set* (also called a *finite poset*) if X is finite.

Given a poset $(X, \prec)$, we define the binary relation $\simeq_{\prec} \subseteq X \times X$ in a pointfree manner as follows:

$$\simeq_{\prec} \overset{df}{=} (X \times X) \setminus (\prec \cup \prec^{-1})$$

In other words, for all $a, b \in X$, $a \simeq_{\prec} b$ if and only if $\neg(a \prec b) \wedge \neg(b \prec a)$, that is if and only if a and b are either *distinct incomparable* with respect to (w.r.t.) $\prec$ or *identical* elements of X.

Let id_X denote the identity relation on X, i.e. $id_X = \{(x,x) | x \in X\}$. We then define the *distinct incomparability relation* as following

$$\frown_{\prec} \overset{df}{=} \simeq_{\prec} \setminus id_X.$$

Proposition 2.3. *For any poset* $(X, \prec)$, $\simeq_{\prec} = \frown_{\prec} \cup id_X$.

Proof. Since $\simeq_{\prec} \overset{df}{=} (X \times X) \setminus (\prec \cup \prec^{-1})$ and $id_X \nsubseteq \prec$, we have $id_X \subseteq \simeq_{\prec}$. Hence,

$$\frown_{\prec} \cup\, id_X = (\simeq_{\prec} \setminus id_X) \cup id_X = \simeq_{\prec} .$$

□

For our convenience, from a poset $(X, \prec)$ we also define the following binary relations $\prec^{\frown} \subseteq X \times X$ and $\preceq \subseteq X \times X$ as

$$\prec^{\frown} \overset{df}{=} \prec \cup \frown_{\prec}$$
$$\preceq \overset{df}{=} \prec \cup\, id_X$$

Intuitively, $a \prec^{\frown} b$ means a is "less than" or incomparable to b and $a \preceq b$ means a is "less than" or equal to b.

If the relation $\frown_{\prec}$ of a poset $(X, \prec)$ is empty, then the partial order $\prec$ is called a *total* (or *linear*) order, and the pair $(X, \prec)$ is called a *totally ordered set.*

A binary relation $\prec \subseteq X \times X$ is a *stratified* (or *weak*) order if and only if $(X, \prec)$ is a poset and $\simeq_{\prec}$ is an equivalence relation.

Proposition 2.4. *For any poset* $(X, \prec)$ *the following are equivalent:*

1. $\simeq_{\prec}$ *is an equivalence relation*
2. *for all* $x, y, z \in X$, *if* $(x \frown_{\prec} y \wedge y \frown_{\prec} z)$ *then* $(x \frown_{\prec} z \vee x = z)$

Proof.
- (1)⇒(2): Assume that $\simeq_{\prec}$ is an equivalence relation and $x \frown_{\prec} y$ and $y \frown_{\prec} z$, we want to show that $x \frown z$ or $x = z$. Since $\frown_{\prec} \subset \simeq_{\prec}$, it follows that $x \simeq_{\prec} y$ and $y \simeq_{\prec} z$. By the transitivity of the equivalence relation $\simeq_{\prec}$, we have $x \simeq_{\prec} z$. By Proposition 2.3 we have $\simeq_{\prec} = \frown_{\prec} \cup id_X$, so it follows that $x \frown z$ or $x = z$ as desired.

- (2)⇒(1): Assume that for all $x, y, z \in X$, if $x \frown_{\prec} y$ and $y \frown_{\prec} z$ then $x \frown_{\prec} z$ or $x = z$. We want to show $\simeq_{\prec}$ is indeed an equivalence relation.
  - Reflexivity: Since $id_X \subseteq \simeq_{\prec}$, the relation $\simeq_{\prec}$ is reflexive
  - Symmetry: If $a \simeq_{\prec} b$, then $\neg(a \prec b) \wedge \neg(b \prec a)$. But this implies $b \simeq_{\prec} a$. Hence, the relation $\simeq_{\prec}$ is symmetric.
  - Transitivity: Assume $a \simeq_{\prec} b$ and $b \simeq_{\prec} c$, we want to show $a \simeq_{\prec} c$. Since $\simeq_{\prec} = \frown_{\prec} \cup id_X$, there are three possible cases.
    * If $a \frown_{\prec} b$ and $b = c$, then $a \frown_{\prec} c$. Hence, $a \simeq_{\prec} c$.
    * If $a = b$ and $b \frown_{\prec} c$, again we have $a \simeq_{\prec} c$.
    * If $a \frown_{\prec} b$ and $b \frown_{\prec} c$, it follows that $a \frown_{\prec} c$ or $a = c$. Hence, $a \simeq_{\prec} c$.

□

As a result of Proposition 2.4, we can alternatively define that a binary relation $\prec \subseteq X \times X$ is a stratified order if and only if for all $x, y, z \in X$,

$$(x \frown_{\prec} y \wedge y \frown_{\prec} z) \Rightarrow (x \frown_{\prec} z \vee x = z).$$

If $(X, \prec)$ is a poset and A is a nonempty subset of X, and $a \in X$, then:

- a is a *maximal element* of A if $a \in X$ and $\forall x \in A.\ \neg a \prec x$.
- a is a *minimal element* of A if $a \in X$ and $\forall x \in A.\ \neg x \prec a$.
- a is the *greatest element* of A if $a \in A$ and $\forall x \in A.\ x \preceq a$.
- a is the *least element* of A if $a \in A$ and $\forall x \in A.\ a \preceq x$.
- a is an *upper bound* of A if $\forall x \in A.\ x \preceq a$.
- a is a *lower bound* of A if and only if $\forall x \in A.\ a \preceq x$.
- a is the *least upper bound* (also called *supremum*) of A, denoted $\sup(A)$, if
  - $x \preceq a$ for all $x \in A$,
  - for all $b \in X$ if b is an upper bound then $a \preceq b$.

- a is the *greatest lower bound* (also called *infimum*) of A, denoted $\inf(A)$, if
  - $a \preceq x$ for all $x \in A$,
  - for all $b \in X$ if b is a lower bound then $b \preceq a$.
- a set A is called a *chain* if and only if $(A, \prec|_{A\times A})$ is a totally ordered set where

$$R\,|_{B\times C} \overset{df}{=} R \cap (B \times C).$$

The greatest element, the least element, upper bound, lower bound, supremum and infimum might fail to exist. Note that if X is totally ordered by $\prec$, then a maximal element of X is its greatest element (similarly for a minimal element).

2.1.3 Szpilrajn Theorem

Let $\prec_1$ and $\prec_2$ be partial orders on a set X. The partial order $\prec_2$ is is defined to be an *extension* of $\prec_1$ if and only if $\prec_1 \subseteq \prec_2$. The goal of this subsection is to review the Szpilrajn Theorem [31], which is fundamental in the foundation of concurrency theory. Since the original paper is in French, we provide a version of the proof to make the theorem more accessible and the thesis self-contained. Furthermore, the results in Chapter 9 and Chapter 10 are motivated by the Szpilrajn Theorem and its proof. But before doing so, we need some preliminary results.

Lemma 2.1. *Let $(X, \prec)$ be a poset, $a, b \in X$ such that $a \frown_{\prec} b$. The relation $\prec_{a,b}$ defined as*

$$x \prec_{a,b} y \iff (x \prec y \vee (x \preceq a \wedge b \preceq y))$$

is a partial order on X satisfying

1. *$a \prec_{a,b} b$*
2. *$\prec_{a,b}$ is an extension of $\prec$, i.e. $\prec \subset \prec_{a,b}$*

Proof. Firstly, we have to show $\prec_{a,b}$ is indeed a partial order.

- Irreflexivity: for any element $x \in X$, we want to show $\neg(x \prec_{a,b} x)$. Since $\prec$ is irreflexive, we have $\neg(x \prec x)$. It remains to show that $\neg(x \preceq a \wedge b \preceq x)$. Suppose for a contradiction that $(x \preceq a \wedge b \preceq x)$. Since $\prec$ is transitive (and so is $\preceq$), it follows that $a = b$, but this contradicts that $a \frown_{\prec} b$.

- Transitivity: for any three elements $x, y, z \in X$ such that $x \prec_{a,b} y \prec_{a,b} z$, we want to show $x \prec_{a,b} z$. By the definition of $\prec_{a,b}$, there are three possible cases to consider:
  - If $x \prec y$ and $(y \preceq a \wedge b \preceq z)$: Since $x \prec y$ and $y \preceq a$, it follows that $x \preceq a$. So $(x \preceq a \wedge b \preceq z)$.
  - If $(x \preceq a \wedge b \preceq y)$ and $y \prec z$: Since $b \prec y$ and $y \preceq z$, it follows that $b \preceq z$. So $(x \preceq a \wedge b \preceq z)$.
  - If $(x \preceq a \wedge b \preceq y)$ and $(y \preceq a \wedge b \preceq z)$: Since $b \preceq y$ and $y \preceq a$, by transitivity of $\preceq$ we have $b \preceq a$. But this contradicts that $a \frown_{\prec} b$.

Secondly, we have to verify that $a \prec_{a,b} b$, which follows from that $(a \preceq a \wedge b \preceq b)$. Finally, we want to show $\prec \subset \prec_{a,b}$ but this follows from the definition of $\prec_{a,b}$. □

Lemma 2.1 says that for any partial order $(X, \prec)$ if there exists a pair of distinct incomparable elements a, b then we can add suitable pairs of elements into the relation $\prec$ (extends the relation $\prec$) to build a relation $\prec_{a,b}$ such that $a \prec_{a,b} b$, i.e. a is comparable to b.

Although we are only interested in the case of finite sets, Szpilrajn Theorem is proved for the general case of arbitrary posets $(X, \prec)$, where X can be *infinite.* As a result, the proof of Szpilrajn Theorem requires the *Axiom of Choice* (cf. [30, 21, 3]). For the sake of completion we include an equivalent form of the Axiom of Choice called the Kuratowski-Zorn Lemma. Since the proof of the Kuratowski-Zorn Lemma requires introducing prerequisite background on *axiomatic set theory* up to the concepts of *ordinal number* and *transfinite recursion* (cf. [30, 21]), we state the result with only an informal proof sketch. This proof sketch follows the idea of a very short and elegant proof given in [32].

Kuratowski-Zorn Lemma. *Every partially ordered set $(X, \prec)$ in which every chain $C \subseteq X$ has an upper bound contains at least one maximal element.*

Proof. Suppose for a contradiction that the lemma were false. Then there exists a poset $(X, \prec)$ such that every totally ordered subset has an upper bound, and every element $x \in X$ has an element $y \in X$ such that $y > x$. For every chain $C \subseteq X$

we pick an upper bound $g(C) \notin C$, because C has at least one upper bound, and that upper bound has a greater element. However, to actually define the function $g : \mathscr{P}X \to X$, we need the Axiom of Choice to magically "pick the right elements" from the arbitrary large set X.

Using the function g, starting from an arbitrary element $a_0 \in X$, we are going to define a sequence of elements $a_0 < a_1 < a_2 < a_3 < \ldots$ in X using transfinite recursion by defining $a_i = g(\{a_j \mid j < i\})$. We know that every pair of element a_i and a_j are *distinct*, otherwise we have a cycle which contradicts that $(X, \prec)$ is a partial order.

This sequence is really long: the indices are not just the natural numbers, but *all* ordinals. In other words, we can define an injective map from all the ordinals into X. Since there is no set with the "size" of *all* ordinals, we have the desired contradiction. □

Note that we do not need the Axiom of Choice for this proof of the Kuratowski-Zorn Lemma when X is finite. The proof of the Kuratowski-Zorn Lemma for the finite case follows.

Proposition 2.5. *Every finite partially ordered set $(X, \prec)$ in which every chain $C \subseteq X$ has an upper bound contains at least one maximal element.*

Proof. We proceed similarly to the previous proof by assuming the proposition were false. Then there exists a finite poset $(X, \prec)$ such that every chain $C \subseteq X$ has an upper bound, and every element has a greater one. For every chain $C \subseteq X$ we find an upper bound $g(C) \notin C$, and this process is exhaustive because we only search through the finite search space X.

Using the function g, starting from an arbitrary element $a_0 \in X$, we build a sequence of distinct elements $a_0 < a_1 < a_2 < a_3 < \ldots$ in X recursively by defining $a_i = g(\{a_j \mid j < i\})$. Since X is finite, there is some natural number m such that $|X| = m$. Suppose for some a_k where $k < m - 1$, we cannot find any element in X greater than a_k, then we have the desired contradiction. Otherwise, considering the element a_{m-1}, by the assumption, there exist some $y \in X$ such that $a_{m-1} < y$. But y can only be one of the $a_0, \ldots, a_{m-2}$, which implies $y = a_i < \ldots < a_{m-1} < a_i = y$. This contradicts that $(X, \prec)$ is a poset. □

We now provide a proof of Szpilrajn Theorem using Lemma 2.1 and Kuratowski-Zorn Lemma.

Szpilrajn Theorem ([31]). *For every poset $(X, \prec)$ there exists a totally ordered set $(X, \mathcal{T})$ such that $\prec \subseteq \mathcal{T}$.*

Proof. Let us define

$$\tau = \{\mathcal{T} \mid \mathcal{T} \text{ is a partial order on } X \text{ and } \prec \subseteq \mathcal{T}\}.$$

Since $\prec \subseteq \prec$, we know $\tau \neq \emptyset$. Consider $(\tau, \subset)$. Clearly $(\tau, \subset)$ is a poset. Let $C \subseteq \tau$ be a chain, i.e. for each $\mathcal{T}_1, \mathcal{T}_2 \in C$, $\mathcal{T}_1 \subset \mathcal{T}_2$ or $\mathcal{T}_2 \subset \mathcal{T}_1$ or $\mathcal{T}_1 = \mathcal{T}_2$. Define the binary relation $\mathcal{T}_C$ on X as

$$\mathcal{T}_C \overset{df}{=} \bigcup_{\mathcal{T} \in C} \mathcal{T}.$$

We want to show $\mathcal{T}_C$ is a partial order. Clearly $\mathcal{T}_C$ is irreflexive since each $\mathcal{T}$ in C is irreflexive. We need to show transitivity. Assume $x \; \mathcal{T}_C \; y \; \mathcal{T}_C \; z$, we want to show $x \; \mathcal{T}_C \; z$. But it follows that there exist $\mathcal{T}_1, \mathcal{T}_2 \in C$ such that $x \; \mathcal{T}_1 \; y$ and $y \; \mathcal{T}_2 \; z$. There are three cases to consider:

- $\mathcal{T}_1 = \mathcal{T}_2$: This means $x \; \mathcal{T}_1 \; y$ and $y \; \mathcal{T}_1 \; z$. Hence, $x \; \mathcal{T}_C \; z$ by transitivity of $\mathcal{T}_1$.
- $\mathcal{T}_1 \subset \mathcal{T}_2$: This means This means $x \; \mathcal{T}_2 \; y$ and $y \; \mathcal{T}_2 \; z$. Hence, $x \; \mathcal{T}_C \; z$ by transitivity of $\mathcal{T}_1$.
- $\mathcal{T}_1 \subset \mathcal{T}_2$: We have $x \; \mathcal{T}_C \; z$ by transitivity of $\mathcal{T}_1$.

Hence, the relation $\mathcal{T}_C$ is a partial order. By the definition, $\forall \mathcal{T} \in C.\ \mathcal{T} \subseteq \mathcal{T}_C$, so $\mathcal{T}_C$ is an upper bound of the chain C.

We want to show that there exist some element $\mathcal{T}_\prec \in \tau$ such that $\mathcal{T}_\prec$ is the maximal element of τ. From Kuratowski-Zorn Lemma, we can now deduce that there exists $\mathcal{T}_\prec$ such that $\mathcal{T}_\prec$ is a maximal element of τ and $\prec \subseteq \mathcal{T}_\prec$.

We want to show that $\mathcal{T}_\prec$ is total. Suppose for a contradiction that $\mathcal{T}_\prec$ is not total, i.e. there are some pair of element a, b such that $a \frown_{\mathcal{T}_\prec} b$. We can then using Lemma 2.1 to construct $\mathcal{T}_{\prec_{a,b}}$. Clearly since $\prec \subseteq \mathcal{T}_\prec \subseteq \mathcal{T}_{\prec_{a,b}}$, $\prec \subseteq \mathcal{T}_{\prec_{a,b}}$. Hence, $\mathcal{T}_{\prec_{a,b}} \in \tau$ and $\mathcal{T}_\prec \subseteq \mathcal{T}_{\prec_{a,b}}$, which is a contradiction since $\mathcal{T}_\prec$ is maximal. Hence, $(X, \mathcal{T}_\prec)$ is a totally ordered set extending the partial order $\prec$ as desired. □

A total order $\mathcal{T}$ which extends the partial order $\prec$ on X is called a *total (linear) order extension* of $\prec$. A corollary of Szpilrajn Theorem is that every partial order is uniquely determined by the intersection of all of its total order extensions. In other words, a partial order is completely defined by the set of all of its total order extensions.

Lemma 2.2. *Let I be an index set and each $(X, \prec_i)$ be a poset. Then $(X, \prec)$ where*

$$\prec \overset{df}{=} \bigcap_{i \in I} \prec_i$$

is also a poset.

Proof. We want to check:

- Irreflexivity: Assume for a contradiction that there exists $x \in X$ such that $x \prec x$. Since $\prec = \bigcap_{i \in I} \prec_i$, we have $x \prec_i x$. But this contradicts that each $\prec_i$ is a partial order.

- Transitivity: Suppose $x \prec y \prec z$ for some $x, y, z \in Z$, we want to show $x \prec z$. Since it follows that $(x, y), (y, z) \in \bigcap_{i \in I} \prec_i$, we have

  $$\forall i \in I.\ ((x, y) \in \prec_i\ \wedge (y, z) \in \prec_i).$$

  Hence, by transitivity of $\prec_i$,

  $$\forall i \in I.\ (x, z) \in \prec_i .$$

  Thus, $(x, z) \in \bigcap_{i \in I} \prec_i$, which means $x \prec z$.

Hence, the relation $\prec$ is a partial order on X. □

Let $(X, \prec)$ be a poset, we define

$$\mathit{Total}_X(\prec) \overset{df}{=} \{\mathcal{T} \mid (X, \mathcal{T}) \text{ is a totally ordered set and } \prec \subseteq \mathcal{T}\}.$$

Corollary 2.2. *For every poset $(X, \prec)$,*

$$\prec = \bigcap_{\mathcal{T} \in \mathrm{Total}_X(\prec)} \mathcal{T}.$$

Proof. The corollary is correctly formulated, i.e. $\bigcap_{\mathcal{T} \in \mathit{Total}_X(\prec)} \mathcal{T}$ is well-defined, because it follows from Szpilrajn Theorem that $\mathit{Total}_X(\prec) \neq \emptyset$.

($\subseteq$) Since every $\mathcal{T} \in \mathit{Total}_X(\prec)$ satisfies $\prec \subseteq \mathcal{T}$, it follows that

$$\prec \subseteq \bigcap_{\mathcal{T} \in \mathit{Total}_X(\prec)} \mathcal{T}.$$

($\supseteq$) Suppose for a contradiction that $\bigcap_{\mathcal{T} \in \mathit{Total}_X(\prec)} \mathcal{T} \not\subseteq \prec$. Then there is some pair (x, y) satisfying $(x, y) \in \mathcal{T}$ for all $\mathcal{T} \in \mathit{Total}_X(\prec)$ but $(x, y) \notin \prec$. Hence, either $y \prec x$ or $x \frown_{\prec} y$.

- If $y \prec x$: For any $\mathcal{T} \in \mathit{Total}_X(\prec)$, since $\prec \subseteq \mathcal{T}$, it follows that $(x, y) \in \mathcal{T}$ and $(y, x) \in \mathcal{T}$. This contradicts that $\mathcal{T}$ is a total order.

- If $x \frown_{\prec} y$: We observe that by Lemma 2.1, we can build the extension $\prec_{y,x}$ of the partial order $\prec$ where $(y, x) \in \prec_{y,x}$. We then apply the Szpilrajn Theorem for $(X, \prec_{y,x})$ to get a total extension $\mathcal{T}_{y,x}$ of $\prec_{y,x}$, where $(y, x) \in \mathcal{T}_{y,x}$.

  But since $\prec \subseteq \prec_{y,x}$, it follows that $\mathcal{T}_{y,x}$ is also a total extension of $\prec$. Hence, $\mathcal{T}_{y,x} \in \mathit{Total}_X(\prec)$. Since we assume that $(x, y) \in \mathcal{T}$ for all $\mathcal{T} \in \mathit{Total}_X(\prec)$, it follows that $(x, y) \in \mathcal{T}_{y,x}$ and $(y, x) \in \mathcal{T}_{y,x}$, which contradicts that $\mathcal{T}_{y,x}$ is a total order.

Thus, we conclude $\prec = \bigcap_{\mathcal{T} \in \mathit{Total}_X(\prec)} \mathcal{T}$ as desired. □

2.2 Monoids

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

An equivalence relation $\sim \subseteq X \times X$ is a *congruence* in the monoid $(X, \circ, \mathbb{1})$ 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 $a_1, a_2, b_1, b_2 \in X$.

The triple $(X/\sim, \hat{\circ}, [\mathbb{1}])$, where $[a]\hat{\circ}[b] = [a \circ b]$, is called the *quotient monoid* of $(X, \circ, 1)$ under the congruence $\sim$. The mapping $\phi : X \rightarrow X/\sim$ 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.

2.3 Sequences and Step Sequences

By an *alphabet* we shall understand any finite set. For an alphabet Σ, Σ^* denotes the set of all finite sequences of elements (words) of Σ, λ denotes the empty sequence, and any subset of Σ^* is called a *language.* In the scope of this thesis, we only deal with *finite* sequences. Let $\cdot$ denote the sequence concatenation operator (usually omitted). Since the sequence concatenation operator is associative, the triple $(\Sigma^*, \cdot, \lambda)$ is a *monoid* (of sequences).

For each set X, let $\mathscr{P}(X)$ denote the set of all subsets of X, i.e.

$$\mathscr{P}(X) \overset{df}{=} \{Y \mid Y \subseteq X\}.$$

We also let $\widehat{\mathscr{P}}(X)$ denote the set of all *non-empty* subsets of X, i.e.

$$\widehat{\mathscr{P}}(X) \overset{df}{=} \mathscr{P}(X) \setminus \{\emptyset\}.$$

Let $f : A \rightarrow B$ be a function and C is a set, then we let $f[C]$ denote the *range of the restriction* of the function f to the set C, i.e.

$$f[C] \overset{df}{=} \{f(a) \mid a \in C\}.$$

Consider an alphabet $\mathbb{S} \subseteq \widehat{\mathscr{P}}(X)$ for some finite X. The elements of $\mathbb{S}$ are called *steps* and the elements of $\mathbb{S}^*$ are called *step sequences.* For example if $\mathbb{S} = \{\{a\}, \{a,b\}, \{c\}, \{a,b,c\}\}$ then $\{a,b\}\{c\}\{a,b,c\} \in \mathbb{S}^*$ is a step sequence. The triple $(\mathbb{S}^*, \bullet, \lambda)$, where $\bullet$ is the step sequence concatenation operator (usually omitted), is a *monoid* (of step sequences), since the step sequence operator is also associative.

Let $t = A_1 \dots A_k$ be a step sequence. We can uniquely construct its *event-enumerated step sequence* $\overline{t}$ as

$$\overline{t} \overset{df}{=} \overline{A_1} \dots \overline{A_k}$$

where

$$\#event_e(A_1 \dots A_m) \overset{df}{=} |\{i : e \in A_i \wedge 1 \leq i \leq m\}|$$

and

$$\overline{A_i} \overset{df}{=} \{e^{(\#event_e(A_1 \dots A_{i-1})+1)} : e \in A_i\}.$$

We will call such $\alpha = e^{(i)} \in \overline{A_i}$ an *event occurrence* of e. For each event occurrence $\alpha = e^{(i)}$, let $l(\alpha)$ denote the *label* of α, i.e. $l(\alpha) = l(e^{(i)}) = e$. Then from an event-enumerated step sequence $\overline{t} = \overline{A_1} \dots \overline{A_k}$, we can also uniquely construct its corresponding step sequence

$$t = l[\overline{A_1}] \dots l[\overline{A_k}].$$

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 event occurrences in all steps of u. For example, when $u = \{a, b\}\{b, c\}\{c, a\}\{a\}$,

$$\Sigma_u = \{a^{(1)}, a^{(2)}, a^{(3)}, b^{(1)}, b^{(2)}, c^{(1)}, c^{(2)}\}.$$

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 example $pos_u(a^{(2)}) = 3$, $pos_u(b^{(2)}) = 2$, etc.

Given a step sequence u, we define a stratified order $\lhd_u$ on Σ_u by:

$$\alpha \lhd_u \beta \iff pos_u(\alpha) < pos_u(\beta).$$

And we define a relation $\simeq_u$ on Σ_u by:

$$\alpha \simeq_u \beta \iff pos_u(\alpha) = pos_u(\beta).$$

Obviously, we have $\lhd_u^{\simeq} = \lhd_u \cup \simeq_u$ and $\lhd_u^{\frown} = \lhd_u \cup \frown_u$. We can also define $\lhd_u^{\simeq}$ and $\lhd_u^{\frown}$ explicitly as follows:

$$\alpha \lhd_u^{\simeq} \beta \iff pos_u(\alpha) \leq pos_u(\beta)$$
$$\alpha \lhd_u^{\frown} \beta \iff \alpha \neq \beta \wedge pos_u(\alpha) \leq pos_u(\beta)$$

Proposition 2.6. *Given a step sequence $u = B_1 \dots B_n$, the relation $\simeq_u$ is an equivalence relation on Σ_u.*

Proof. Since $\alpha \simeq_u \beta \iff pos_u(\alpha) = pos_u(\beta)$, it follows that $\alpha, \beta \in \overline{B_i}$ for some $1 \leq i \leq n$. Hence, $\simeq_u$ is an equivalence relation defined based on the partitions $\overline{B_1}, \dots, \overline{B_n}$ of Σ_u □

Conversely, let $\lhd$ be a stratified order on a set Σ. The set Σ can be represented as a sequence of equivalence classes $\Omega_{\lhd} = B_1 \dots B_k$ ($k \geq 0$) such that

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

The sequence $\Omega_{\lhd}$ is exactly the *event-enumerated step sequence* which represents $\lhd$. The correctness of the existence of $\Omega_{\lhd}$ is shown the in following proposition.

Proposition 2.7. *If $\lhd$ is a stratified order on a set Σ and A, B are two distinct equivalence classes of $\simeq_{\lhd}$, then either $A \times B \subseteq \lhd$ or $B \times A \subseteq \lhd$.*

Proof. Since both A and B are *non-empty* equivalence classes of $\simeq_{\lhd}$, we pick $a \in A$ and $b \in B$. Clearly, $a \lhd b$ or $b \lhd a$, otherwise $a \frown_{\lhd} b$ which contradicts that a,b are elements from two distinct equivalence classes. There are two cases:

1. If $a \lhd b$: we want to show $A \times B \subseteq \lhd$. Let $c \in A$ and $d \in B$, it suffices to show $c \lhd d$. Assume for contradiction that $\neg(c \lhd d)$. Since $c \not\simeq_{\lhd} d$, it follows that $d \lhd c$. There are three different subcases:

   (a) If $a = c$, then $d \lhd a$ and $a \lhd b$. Hence, $d \lhd b$. This contradicts that $d, b \in B$.

   (b) If $b = d$, then $b \lhd c$ and $a \lhd b$. Hence, $a \lhd c$. This contradicts that $a, c \in A$.

   (c) If $a \neq c$ and $b \neq d$, then $a \frown_{\lhd} c$ and $b \frown_{\lhd} d$ and $\neg(a \frown_{\lhd} d)$ and $\neg(c \frown_{\lhd} b)$. Since $\neg(a \frown_{\lhd} d)$, either $a \lhd d$ or $d \lhd a$.

- If $a \lhd d$: since $d \lhd c$, it follows $a \lhd c$. This contradicts $a \frown_{\lhd} c$.
- If $d \lhd a$: since $a \lhd b$, it follows $d \lhd b$. This contradicts $d \frown_{\lhd} b$.

Therefore, we conclude $A \times B \subseteq \lhd$.

2. If $b \lhd a$: using a symmetric argument, it follows that $B \times A \subseteq \lhd$.

□

The idea of Proposition 2.7 is that if we define a relation $\hat{\lhd}$ on the set of equivalence classes $\{B_1, \ldots, B_n\}$ of $\simeq_{\lhd}$ such that

$$B_i \hat{\lhd} B_j \iff B_i \times B_j \subseteq \lhd,$$

then $\hat{\lhd}$ is a total order on $\{B_1, \ldots, B_n\}$. Hence, Proposition 2.7 is fundamental for understanding the *equivalence* of stratified partial orders and step sequences.

Since total order is a special case of stratified order (equivalence classes of $\simeq_{\lhd}$ are singletons), each sequence can be interpreted as a total order, and each finite total order can be represented by a sequence. Observe that each $s = x_1 \ldots x_n$ can be seen as the step sequence $s' = \{x_1\} \ldots \{x_n\}$. Hence, if $\overline{s'} = \{\alpha_1\} \ldots \{\alpha_n\}$ is the event-enumerated step sequence of s', then we can define the enumerated sequence of s to be the sequence $\overline{s} = \alpha_1 \ldots \alpha_n$. We let $\Sigma_s = \Sigma_{s'}$, $\lhd_s = \lhd_{s'}$ and $\frown_s = \frown_{s'}$. Since $\frown_s = \emptyset$, it follows that $(\Sigma_s, \lhd_s)$ is a totally ordered set representing the sequence s. Conversely, given a finite totally ordered set $(\Sigma, \lhd)$ (assume Σ is a set of event occurrences), we let $\Omega_{\lhd} = \{\alpha_1\} \ldots \{\alpha_n\}$. Then we apply the label function l to get a sequence $s_{\lhd} = l(\alpha_1) \ldots l(\alpha_n)$, which represents the totally ordered set $(\Sigma, \lhd)$.

Chapter 3

Equational Monoids with Compound Generators

3.1 Equational Monoids and Mazurkiewicz Traces

Let $M = (X, \circ, \mathbb{1})$ be a *monoid* and let

$$EQ = \{\ x_i = y_i \mid i = 1, \ldots, n\ \}$$

be a finite set of *equations*. Define $\equiv_{EQ}$ (or just $\equiv$) to be 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 call the relation $\equiv_{EQ}$ as the *congruence defined by EQ*, or *EQ-congruence.*

The *quotient monoid* $M_{\equiv_{EQ}} = (X/{\equiv_{EQ}}, \hat{\circ}, [\mathbb{1}])$, where $[x]\hat{\circ}[y] = [x \circ y]$, is called an *equational monoid* (see for example [26]).

The following folklore result shows that the relation $\equiv_{EQ}$ can also be *uniquely* defined in an explicit way.

Proposition 3.1. *For equational monoids, the EQ-congruence $\equiv$ is the reflexive symmetric transitive closure of the relation $\approx$, 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., $x_1 \circ x_2 \equiv y_1 \circ y_2$. Thus, $\equiv$ 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. □

Definition 3.1 ([8, 25]). Let $M = (E^*, \circ, \lambda)$ be a *free monoid* generated by E, the relation $ind \subseteq E \times E$ be an irreflexive and symmetric relation (called *independency* or *commutation*), and

$$EQ \overset{df}{=} \{ab = ba \mid (a, b) \in ind\}.$$

Let $\equiv_{ind}$, called *trace congruence*, be the congruence defined by EQ. Then the equational monoid $M_{\equiv_{ind}} = (E^*/\equiv_{ind}, \hat{\circ}, [\lambda])$ is a *free partially commutative monoid* or *monoid of Mazurkiewicz traces*. The pair (E, ind) is called a *concurrent alphabet* (or *trace alphabet*).

We will omit the subscript *ind* from trace congruence and write $\equiv$ if it causes no ambiguity.

Example 3.1. 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 Mazurkiewicz traces. Also $t_3 = t_1 \hat{\circ} t_2$ (as $[abcbca] = [abc] \hat{\circ} [bca]$).

For more details on Mazurkiewicz traces, the reader is referred to [8, 25]. For the equational representations of Mazurkiewicz traces, the reader is referred to [26].

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 that they are sets, this internal structure may be used when defining the set of equations EQ.

Let E be a finite set and $\mathbb{S} \subseteq \widehat{\mathscr{P}}(E)$ be a non-empty set of non-empty subsets of E satisfying $\bigcup_{A\in\mathbb{S}} A = E$. The free monoid $(\mathbb{S}^*, \circ, \lambda)$ is called a *free monoid of step sequences* over E, with the elements of $\mathbb{S}$ called *steps* and the elements of $\mathbb{S}^*$ called *step sequences.* We assume additionally that the set $\mathbb{S}$ is *subset closed*, i.e. for all $A \in \mathbb{S}$, $\widehat{\mathscr{P}}(A) \subseteq \mathbb{S}$.

Definition 3.2. Let EQ be the following set of equations:

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

where $A_i, B_i, C_i \in \mathbb{S}$, $C_i = A_i \cup B_i$, $A_i \cap B_i = \emptyset$, for $i = 1, \ldots, n$, and let $\equiv_{abs}$ be the congruence defined by EQ. The equational monoid $(\mathbb{S}^*/\!\equiv_{abs}, \hat{\circ}, [\lambda])$ will be called an *absorbing monoid over step sequences.*

We will omit the subscript *abs* from the absorbing monoid congruence and write $\equiv$ if it causes no ambiguity.

Example 3.2. Let $E = \{a, b, c\}$, $\mathbb{S} = \{\{a, b, c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a\}, \{b\}, \{c\}\}$, 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 $\mathbb{S}^*/\!\equiv$, and

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

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

Comtraces (*com*bined *traces*), introduced in [14] 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*.

Definition 3.3. ([14]) Let E be a finite set (of events) and let $ser \subseteq sim \subset E \times E$ be two relations called *serialisability* and *simultaneity* respectively and the relation sim is irreflexive and symmetric. Then the triple (E, sim, ser) is called the *comtrace alphabet.*

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 [18]), 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 $\mathbb{S}$, the set of all (potential) *steps*, as the set of all cliques of the graph (E, sim), i.e.

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

Definition 3.4. Let (E, sim, ser) be a comtrace alphabet and $\equiv_{ser}$, called *comtrace congruence,* be the EQ-congruence defined by the set of equations

$$EQ \overset{df}{=} \{A = BC \mid A = B \cup C \in \mathbb{S} \wedge B \times C \subseteq ser\}.$$

Then the absorbing monoid $(\mathbb{S}^*/\equiv_{ser}, \hat{\circ}, [\lambda])$ is called a monoid of *comtraces* over (E, sim, ser).

In Definition 3.4, since ser is irreflexive, it follows that for each $(A = BC) \in EQ$ we have $B \cap C = \emptyset$. Hence, each comtrace monoid is an absorbing monoid.

By Proposition 3.1, the comtrace congruence relation can also be defined explicitly in non-equational form as follows.

Definition 3.5 ([14]). Let $\theta = (E, sim, ser)$ be a comtrace alphabet and let $\mathbb{S}^*$ the set of all step sequences defined on θ. Let $\approx_{ser} \subseteq \mathbb{S}^* \times \mathbb{S}^*$ be the relation comprising all pairs (t, u) of step sequences such that $t = wAz$ and $u = wBCz$ where $w, z \in \mathbb{S}^*$ and A, B, C are steps satisfying $B \cup C = A$ and $B \times C \subseteq ser$. Then we define $\equiv_{ser} \overset{df}{=} (\approx_{ser} \cup \approx_{ser}^{-1})^*$, i.e. $\equiv_{ser}$ is the reflexive symmetric transitive closure of $\approx_{ser}$.

We will omit the subscript ser from comtrace congruence and $\approx_{ser}$, and only write $\equiv$ and $\approx$ if it causes no ambiguity.

Example 3.3. Let $E = \{a, b, c\}$ where a, b and c are three atomic operations defined as follows (we assume simultaneous reading is allowed):

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

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 $\mathbb{S} = \{\{a\}, \{b\}, \{c\}, \{b, c\}\}$, $EQ = \{\{b, c\} = \{b\}\{c\}\}$. 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 (and the fact that steps are sets is used in the definition of quotient congruence), 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 3.2. *If $ser = sim$ then for each comtrace $t \in \mathbb{S}^*/\equiv_{ser}$ there is a step sequence $x = \{a_1\} \ldots \{a_k\} \in \mathbb{S}^*$, where $a_i \in E$, $i = 1, \ldots, k$ such that $t = [x]$.*

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

This means that if $ser = sim$, then each comtrace $t \in \mathbb{S}^*/\equiv_{ser}$ 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 3.2 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 3.2 *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 [18]). In principle the events are *synchronous* if they can be executed in one step $\{a_1, \ldots, a_k\}$ but this execution cannot be modelled by any sequence of proper subsets of $\{a_1, \ldots, a_k\}$. In general 'synchrony' is not necessarily 'simultaneity' as it does not include the concept of time [5]. It appears however the mathematics used to deal with synchrony is very close to that to deal with simultaneity.

Definition 3.6. Let (E, sim, ser) be a given comtrace alphabet. We define the relations *ind*, *syn* and the set $\mathbb{S}_{syn}$ 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:

$$(a, b) \in syn \iff (a, b) \in sim \wedge (a, b) \notin ser \cup ser^{-1},$$

- $\mathbb{S}_{syn} \subseteq \mathbb{S}$, called *synchronous steps*, and defined as:

$$A \in \mathbb{S}_{syn} \iff A \neq \emptyset \wedge (\forall a, b \in A.\ (a, b) \in syn).$$

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 either a followed by b, or b followed by a. In principle, the relation *syn* is a counterpart of 'synchrony' as understood in [18]. If $A \in \mathbb{S}_{syn}$ then the set of events A can be executed as one step, but it *cannot* be simulated by any sequence of its subsets.

Example 3.4. 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,

$$\begin{aligned}
\mathbb{S} &= \{\{a, b\}, \{a, c\}, \{a, d\}, \{a\}, \{b\}, \{c\}, \{e\}\} \\
ind &= \{(a, b), (b, a)\} \\
syn &= \{(a, d), (d, a)\} \\
\mathbb{S}_{syn} &= \{\{a, d\}\}
\end{aligned}$$

Since $\{a,d\} \in \mathbb{S}_{syn}$ the step $\{a,d\}$ *cannot* be split. For example the comtraces $x_1 = [\{a,b\}\{c\}\{a\}]$, $x_2 = [\{e\}\{a,d\}\{a,c\}]$, and $x_3 = [\{a,b\}\{c\}\{a\}\{e\}\{a,d\}\{a,c\}]$ are the following sets of step sequences:

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

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 [13] (see Chapter 7 of this thesis). Let us analyse the following example.

Example 3.5. 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 [13, 15] 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*, since simultaneous writing on the same variable x is not allowed!

The set of all equivalent executions (or runs) involving one occurrence of the operations a, b and c,

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

is a valid concurrent history or behaviour [13, 15].

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

The concurrent behaviour described by x from Example 3.5 can easily be modelled by a *generalised order structure* of [10] (see Chapter 8 of this thesis). In this subsection we will introduce the concept of *generalised comtraces*, quotient monoid representations of generalised stratified order structures. But we start with a slightly more general concept of *partially commutative absorbing monoid over step sequences.*

Definition 3.7. Let E be a finite set and let $(\mathbb{S}^*, \circ, \lambda)$ be a free monoid of step sequences over E where $\mathbb{S}$ is subset closed. Let EQ_1, EQ_2, EQ be the following sets of equations

$$EQ_1 = \{\ C_1 = A_1B_1\ , \ldots ,\ C_n = A_nB_n\ \},$$

where $A_i, B_i, C_i \in \mathbb{S}$, $C_i = A_i \cup B_i$, $A_i \cap B_i = \emptyset$, for $i = 1, \ldots, n$,

$$EQ_2 = \{\ E_1F_1 = F_1E_1\ , \ldots ,\ E_kF_k = F_kE_k\ \},$$

where $E_i, F_i \in \mathbb{S}$, $E_i \cap F_i = \emptyset$, $E_i \cup F_i \notin \mathbb{S}$, for $i = 1, \ldots, k$, and

$$EQ = EQ_1 \cup EQ_1.$$

Let $\equiv_{pcabs}$ be the EQ-congruence defined by the set of equations EQ. Then the equational monoid $(\mathbb{S}^*/\equiv_{pcabs}, \hat{\circ}, [\lambda])$ will be called an *partially commutative absorbing monoid over step sequences.*

We will omit the subscript *pcabs* from partially commutative absorbing monoid congruence and write $\equiv$ if it causes no ambiguity.

Remark 3.1. There is an *important difference* between the equation $ab = ba$ for Mazurkiewicz traces, and the equation $\{a\}\{b\} = \{b\}\{a\}$ for partially commutative absorbing monoids. For Mazurkiewicz 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 $\{a,b\} = \{b\}\{a\}$ belongs to the set of equations. In other words, 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 execution orders are equivalent but simultaneous execution is *not* allowed. □

We will now extend the concept of a comtrace by adding a relation that generates the set of equations EQ_2.

Definition 3.8. Let E be a finite set (of events). Let $ser, sim, inl \subset E \times E$ be three relations called *serialisability*, *simultaneity* and *interleaving* respectively satisfying:

- sim and inl are irreflexive and symmetric,
- $ser \subseteq sim$, and
- $sim \cap inl = \emptyset$.

Then the triple (E, sim, ser, inl) is called a *generalised comtrace alphabet.*

The interpretation of the relations sim and ser is as in Definition 3.3, 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 $\mathbb{S}$, the set of all (potential) *steps*, as the set of all cliques of the graph (E, sim).

Definition 3.9. Let (E, sim, ser, inl) be a generalised comtrace alphabet and $\equiv_{gcom}$, called *generalised comtrace congruence*, be the EQ-congruence defined by the set of equations $EQ = EQ_1 \cup EQ_2$, where

$$EQ_1 \overset{df}{=} \{A = BC \mid A = B \cup C \in \mathbb{S} \wedge B \times C \subseteq ser\},$$

and

$$EQ_2 \overset{df}{=} \{BA = AB \mid A \in \mathbb{S} \wedge B \in \mathbb{S} \wedge A \times B \subseteq inl\}.$$

The equational monoid $(\mathbb{S}^*/\equiv_{gcom}, \hat{\circ}, [\lambda])$ is called a monoid of *generalised comtraces* over (E, sim, ser, inl).

In Definition 3.9, since *ser* and *inl* are irreflexive, we have

- if $(A = BC) \in EQ_1$, then $B \cap C = \emptyset$, and
- if $(AB = BA) \in EQ_2$, then $A \cap B = \emptyset$.

Also since $inl \cap sim = \emptyset$, we know that $(AB = BA) \in EQ_2$ implies that $A \cup B \notin \mathbb{S}$. Hence, each monoid of generalised comtraces is a commutative absorbing monoid.

By Proposition 3.1, the generalised comtrace congruence relation can also be defined explicitly in non-equational form as following.

Definition 3.10. Let $\theta = (E, sim, ser, inl)$ be a generalised comtrace alphabet and let $\mathbb{S}^*$ the set of all step sequences defined on θ.

Let $\approx_1 \subseteq \mathbb{S}^* \times \mathbb{S}^*$ be the relation comprising all pairs (t, u) of step sequences such that $t = wAz$ and $u = wBCz$ where $w, z \in \mathbb{S}^*$ and A, B, C are steps satisfying $B \cup C = A$ and $B \times C \subseteq ser$.

Let $\approx_2 \subseteq \mathbb{S}^* \times \mathbb{S}^*$ be the relation comprising all pairs (t, u) of step sequences such that $t = wABz$ and $u = wBAz$ where $w, z \in \mathbb{S}^*$ and A, B are steps satisfying $A \times B \subseteq inl$.

Let $\approx_{gcom} \overset{df}{=} \approx_1 \cup \approx_2$. Then we define $\equiv_{gcom} \overset{df}{=} (\approx_{gcom} \cup \approx_{gcom}^{-1})^*$, i.e. $\equiv_{gcom}$ is the reflexive symmetric transitive closure of $\approx_{gcom}$.

The name "generalised comtraces" comes from that fact that when $inl = \emptyset$, Definition 3.9 is the same as Definition 3.4 of comtrace monoids. We will omit the subscript *gcom* from the generalised comtrace congruence and $\approx_{gcom}$, and only write $\equiv$ and $\approx$ if it causes no ambiguity.

Example 3.6. The set x from Example 3.5 is a generalised comtrace with $E = \{a, b, c\}$, $ser = sim = \{(a, c), (c, a), (b, c), (c, b)\}$, $inl = \{(a, b), (b, a)\}$, and $\mathbb{S} = \{\{a, c\}, \{b, c\}, \{a\}, \{b\}, \{c\}\}$. So we write $x = [\{a, c\}\{b\}]$.

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 3.7. 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 build only one step $\{a, b\} = \{b, a\}$.

To provide more intuition, consider the following interpretation of a, b, c_1 and c_2. Assume we have a buffer of 8 bits. Each event a or b fills consecutively 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 a simultaneous start is impossible (beginnings and endings are instantaneous events after all), filling the buffer takes time, and simultaneous executions (i.e. time overlaps in this case) 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 model adequately the situations as that in Example 3.7 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 *decomp* : $G \to \widehat{\mathscr{P}}(E)$, called *decomposition*, that satisfies for all $a \in E$, $decomp(a) = \{a\}$ and for all $a \notin E$, $|decomp(a)| \geq 2$.

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

Definition 3.11. The set of absorbing equations is defined as follows:

$$EQ \overset{df}{=} \{c_i = a_i \circ b_1 \mid i = 1, \ldots, n\}$$

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

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

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

We will omit the subscript *abs&cg* from the congruence of absorbing monoid with compound generators and write $\equiv$ if it causes no ambiguity.

Example 3.8. Consider the absorbing monoid with compound generators where: $E = \{a, b\}$, $G = \{a, b, c_1, c_2\}$, $decomp(c_1) = decomp(c_2) = \{a, b\}$, $decomp(a) = \{a\}$, $decomp(b) = \{b\}$, and $EQ = \{\ c_1 = a \circ b,\ c_2 = b \circ a\ \}$. Now we have $[c_1] = \{c_1, a \circ b\}$ and $[c_2] = \{c_2, b \circ a\}$, which models the case from Example 3.7.

Chapter 4

Canonical Representations

We will show that all kinds of monoids discussed in previous chapter have some kind of *canonical* representation, which intuitively corresponds to *maximally concurrent execution of concurrent histories*, i.e. "executing as much as possible in parallel". This kind of semantics is formally defined and analysed in [4].

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

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

- each x_i is fully commutative, for $i = 1, \ldots, n$,
- 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 4.1 ([1, 4]). *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 representation. For instance the trace $t_3 = [abcbca]$ from Example 3.1 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 [28] are used to represent Mazurkiewicz traces, see [4] for more details). To get 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 $x = x_1 \dots x_n$, each x_i is minimal with the lexicographic ordering. The canonical form with this additional condition is called *Foata canonical form.*

Theorem 4.2 ([1]). *Every trace has a unique representation in the Foata canonical form.* □

A canonical form as defined at the beginning of this chapter 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 represent intuition better than canonical representation corresponds to the maximally concurrent execution [4]. An alternative characterisation of Foata normal form introduced in [7] involved the concept of *elementary step*, which is very similar to the notion of step sequence, and will be discussed later in Proposition 5.3.

Definition 4.1. Let $(\mathbb{S}^*, \circ, \lambda)$ be a free monoid of step sequences over E, and let

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

be an appropriate set of absorbing equations. Let $M_{abs} = (\mathbb{S}^*/\equiv, \hat{\circ}, [\lambda])$ be the absorbing monoid determined by EQ. A step sequence $t = A_1 \dots A_k \in \mathbb{S}^*$ is *canonical* (w.r.t. M_{abs}) if for all $i \geq 2$ there is *no* step $B \subseteq A_i$ satisfying:

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

It is very important to notice that in the above definition $B = A_i$ is allowed but $B = \emptyset$ is not, since B is a *step*.

For every step sequence $x = B_1 \ldots B_r$, we define

$$\mu(x) \overset{df}{=} 1 \cdot |B_1| + \ldots + r \cdot |B_r| \tag{4.1}$$

Theorem 4.3. *Let M_{abs} be an absorbing monoid over step sequences, $\mathbb{S}$ be its set of steps, and EQ be its set of absorbing equations. For every step sequence $t \in \mathbb{S}^*$ there is a canonical step sequence u representing $[t]$.*

Proof. We know that there is at least one $u \in [t]$ such that $\mu(u) \leq \mu(x)$ for all $x \in [t]$. Suppose $u = A_1 \ldots A_k$ is not canonical. Then there is $i \geq 2$ and a step $B \in \mathbb{S}$ satisfying:

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

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

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

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

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

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

Corollary 4.1. *Let M_{abs} be an absorbing monoid over step sequences, $\mathbb{S}$ be its set of steps, and EQ be its set of absorbing equations. If a step sequence $u \in \mathbb{S}^*$ satisfying $\mu(u) \leq \mu(x)$ for all $x \in [u]$, then u is canonical w.r.t M_{abs}.* □

When M_{abs} is a monoid of comtraces, Definition 4.1 is equivalent to the definition of canonical step sequence proposed in [14] as shown in the following proposition.

Proposition 4.1. *If a step sequence $u = A_1 \ldots A_k \in \mathbb{S}^*$ is canonical w.r.t. a comtrace monoid $(\mathbb{S}^*/\equiv, \hat{\circ}, [\lambda])$ over a comtrace alphabet (E, sim, ser) if and only if for all $i \geq 2$ there is no step $B \subseteq A_i$ satisfying $A_{i-1} \times B \subseteq ser$ and $B \times (A_i \setminus B) \subseteq ser$.*

Proof. Recall the set of equations for comtrace in Definition 3.4 is defined as:

$$EQ \overset{df}{=} \{C = AB \mid C = A \cup B \in \mathbb{S} \wedge A \times B \subseteq ser\}.$$

Hence, u is canonical if and only if for all $i \geq 2$ there is no step $B \subseteq A_i$ such that $A_{i-1} \times B \subseteq ser$ and $B \times (A_i \setminus B) \subseteq ser$ as desired. □

Definition 4.2. Let $(\mathbb{S}^*, \circ, \lambda)$ be a free monoid of step sequences over E, and $M_{pcabs} = (\mathbb{S}^*/\equiv, \hat{\circ}, [\lambda])$ be a partially commutative absorbing monoid. Then a step sequence $t = A_1 \ldots A_k \in \mathbb{S}^*$ is *canonical* (w.r.t. M_{pcabs}) if $\mu(t) \leq \mu(u)$ for all $u \in [t]$.

Since each generalised comtrace monoid is a special case of partially commutative absorbing monoid, the above definition also applies to generalised comtrace monoids.

Definition 4.3. Let $(G^*, \circ, \lambda)$ be a free monoid with compound generators, and let

$$EQ = \{\ c_1 = a_1 b_1\ , \ldots,\ c_n = a_n b_n\ \}$$

be an appropriate set of absorbing equations. Let $M_{abs\&cg} = (G^*/\equiv, \hat{\circ}, [\lambda])$. A sequence $t = a_1 \ldots a_k \in G^*$ is *canonical* (w.r.t. $M_{abs\&cg}$) 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 $[x]$.

Since the proof of Theorem 4.3 can also be applied to the case of a free monoid with compound generators, we have the following proposition.

Proposition 4.2. *Let $(X, \hat{\circ}, [\lambda])$ be an absorbing monoid over step sequences, or a partially commutative absorbing monoid over step sequences, or an absorbing monoid with compound generators. Then 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 Proposition 4.2. To prove this lack of uniqueness, it suffices to show it for the absorbing monoids over step sequences. Consider the following example.

Example 4.1. Let $E = \{a, b, c\}$, $\mathbb{S} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a\}, \{b\}, \{c\}\}$ and EQ be 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 $\{a,b\}\{c\} \approx \{a\}\{b\}\{c\} \approx \{a\}\{c\}\{b\} \approx \{a,c\}\{b\}$, 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 construction which we will give in Definition 10.4 is an example of how to do so with the assumption that there is a total order on a set of events E.

However, each comtrace has a *unique* canonical representation as defined in Definition 4.1. Although not mentioned in [14], the *uniqueness* of canonical representation follows directly from [14, Proposition 3.1] and [14, Proposition 3.1]. However, we will provide an alternative proof using only the algebraic properties of comtrace congruence in the next chapter.

Chapter 5

Algebraic Properties of Comtrace Congruence

Analogous to how operations on sequences (words) provide more tools to study their generated partial orders in the theory of Mazurkiewicz traces, the goal of this chapter is to provide similar algebraic operations for step sequences which we hope will eventually help to analyse stratified order structure [15]. When dealing with Mazurkiewicz traces, the tools to deal with sequences (words) are simple but powerful operations like *left/right cancellation* and *projection* on sequences, which are well-known and intuitive (see [25]). However, it is not obvious what operations are needed when working with step sequences. In the next section, we try to tackle this problem by introducing similar tools for step sequences and utilise them to analyse some basic properties of comtrace congruence.

5.1 Operations on Step Sequences and Properties of Comtrace Congruence

Let us consider a comtrace alphabet $\theta = (E, sim, ser)$ where we reserve $\mathbb{S}$ to denote the set of all possible steps of θ throughout this chapter.

For each step sequence or enumerated step sequence $x = X_1 \dots X_k$, let

$$weight(x) \stackrel{df}{=} \Sigma_{i=1}^{k} |X_i|$$

denote the *step sequence weight* of w, where $|X_i|$ denotes the cardinality of the set X_i. We also define

$$\biguplus(x) \stackrel{df}{=} \bigcup_{i=1}^{k} X_i.$$

For any $a \in E$ and a step sequence $w = A_1 \dots A_k \in \mathbb{S}^*$, we define $|w|_a$, the number of occurrences of a in w, as $|w|_a \stackrel{df}{=} |\{A_i | 1 \le i \le k \wedge a \in A_i\}|$.

Due to the commutativity of the independency relation for Mazurkiewicz traces, the *mirror rule*, which says if two sequences are congruent then their *reverses* are also congruent, holds for *Mazurkiewicz trace congruence* [8]. Hence, in trace theory, we only need a *right cancellation* operation to get new congruent sequences from the old ones, since the *left cancellation* comes from the right cancellation of the reverses.

However, the *mirror rule* does not hold for comtrace congruence since the relation *ser* is usually not commutative. Example 3.3 works as a counter example since $\{a\}\{b, c\} \equiv \{a\}\{b\}\{c\}$ but $\{b, c\}\{a\} \not\equiv \{c\}\{b\}\{a\}$. Thus, we define separate left and right cancellation operators for comtraces.

Let $a \in E$, $A \in \mathbb{S}$ and $w \in \mathbb{S}^*$. The operator $\div_R$, *step sequence right cancellation*, is defined as

$$\lambda \div_R a \stackrel{df}{=} \lambda,$$

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

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

$$\lambda \div_L a \stackrel{df}{=} \lambda,$$

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

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

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

Proposition 5.1.

1. $u \equiv v \implies weight(u) = weight(v)$. (step sequence weight equality)

2. $u \equiv v \implies |u|_a = |v|_a$. (event-preserving)

3. $u \equiv v \implies u \div_R a \equiv v \div_R a$. (right cancellation)

4. $u \equiv v \implies u \div_L a \equiv v \div_L a$. (left cancellation)

5. $u \equiv v \iff \forall s, t \in \mathbb{S}^*.\ sut \equiv svt$. (step subsequence cancellation)

6. $u \equiv v \implies \pi_D(u) \equiv \pi_D(v)$. (projection rule)

Proof. 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 $u \approx v \implies weight(u) = weight(v)$. Because $A = B \cup C$ and $B \cap C = \emptyset$, we have $weight(A) = |A| = |B| + |C| = weight(BC)$. Hence, $weight(u) = weight(x) + weight(A) + weight(z) = weight(x) + weight(BC) + weight(z) = weight(v)$.

2. It suffices to show that $u \approx v \implies |u|_a = |v|_a$. There are two cases:

- $a \in A$: Then it can't be the case that $a \in B \wedge a \in C$ because $B \cap C = \emptyset$. Since $A = B \cup C$, either $a \in B$ or $a \in C$. Then $|A|_a = |BC|_a$. Therefore, $|u|_a = |x|_a + |A|_a + |z|_a = |x|_a + |BC|_a + |z|_a = |v|_a$.

- $a \notin A$: Since $A = B \cup C$, $a \notin B \wedge a \notin C$. So $|A|_a = |BC|_a = 0$. Therefore, $|u|_a = |x|_a + |z|_a = |v|_a$.

3. It suffices to show that $u \approx v \implies u \div_R a \approx v \div_R a$. There are four cases:

- $a \in \biguplus(y)$: Let $z = y \div_R a$. Then $u \div_R a = xAz \approx xBCz = v \div_R a$.
- $a \notin \biguplus(y)$, $a \in A \cap C$: Then $u \div_R a = x(A \setminus \{a\})y \approx xB(C \setminus \{a\})y = v \div_R a$.
- $a \notin \biguplus(y)$, $a \in A \cap B$: Then $u \div_R a = x(A \setminus \{a\})y \approx x(B \setminus \{a\})Cy = v \div_R a$.
- $a \notin \biguplus(Ay)$: Let $z = x \div_R a$. Then $u \div_R a = zAy \approx zBCy = v \div_R a$.

4. Dually to (3).

5. ($\Rightarrow$) It suffices to show that $u \approx v \implies \forall s, t \in \mathbb{S}^*.\ sut \approx svt$. For any two step sequences $s, t \in \mathbb{S}^*$, we have $sut = sxAyt$ and $svt = sxBCyt$. But this clearly implies $sut \approx svt$ by the definition of $\approx$.

($\Leftarrow$) For any two step sequences $s, t \in \mathbb{S}^*$, since $sut \equiv svt$, it follows that

$$(sut \div_R t) \div_L s = u \equiv v = (svt \div_R t) \div_L s.$$

Therefore, $u \equiv v$.

6. It suffices to show that $u \approx v \implies \pi_D(u) \approx \pi_D(v)$. Note that $\pi_D(A) = \pi_D(B) \cup \pi_D(C)$ 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 5.1 (3), (4) and (6) do not hold for an arbitrary absorbing monoid. For the absorbing monoid from Example 3.2 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:

$$w \div_R \{a_1, \ldots, a_k\} \overset{df}{=} (\ldots((w \div_R a_1) \div_R a_2) \ldots) \div_R a_k,$$

and

$$w \div_R A_1 \ldots A_k \overset{df}{=} (\ldots((w \div_R A_1) \div_R A_2) \ldots) \div_R A_k.$$

We define dually for $\div_L$.

Corollary 5.1. *For all $u, v, x \in \mathbb{S}^*$, we have*

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

Proof. 1. We prove it by induction on k, the number of steps of x. When $k = 0$, have $x = \lambda$. Hence, from $u \equiv v$, it follows that

$$u \div_R x = u \equiv v = v \div_R x.$$

When $k > 0$, we assume $x = A_1 \ldots A_k$. By the induction hypothesis, we have

$$u \div_R A_1 \ldots A_{k-1} \equiv v \div_R A_1 \ldots A_{k-1}.$$

Let $t = u \div_R A_1 \ldots A_{k-1}$ and $s = v \div_R A_1 \ldots A_{k-1}$. It suffices to show $t \div_R A_k \equiv s \div_R A_k$. Let $A_k = \{a_1 \ldots a_n\}$. We will prove it by induction on n. When $n = 1$, by Proposition 5.1(3), we have

$$t \div_R A_k = t \div_R a_1 \equiv s \div_R a_1 = s \div_R A_k.$$

When $n > 1$, by the induction hypothesis, we have

$$t \div_R \{a_1 \ldots a_{n-1}\} \equiv s \div_R \{a_1 \ldots a_{n-1}\}.$$

It follows that

$$\begin{aligned} t \div_R A_k &= (t \div_R \{a_1 \ldots a_{n-1}\}) \div_R a_n \\ &\equiv (s \div_R \{a_1 \ldots a_{n-1}\}) \div_R a_n = s \div_R A_k. \end{aligned}$$

2. Dually to (1). □

To prepare for the proof of uniqueness property of canonical representation for comtraces, we prove the following technical lemma.

Lemma 5.1. *For all step sequences $u, w, s \in \mathbb{S}^*$, steps $A, B, C_1, \dots, C_n \in \mathbb{S}$ and a symbol $a \in E$, the following hold*

1. $A \equiv C_1 \dots C_{k-1}C_k \dots C_n \implies \biguplus(C_1 \dots C_{k-1}) \times \biguplus(C_k \dots C_n) \subseteq ser$
2. $(u(A \cup \{a\}) \equiv wB \wedge a \notin A \wedge a \notin B) \implies \{a\} \times (B \setminus A) \subseteq ind$
3. $((A \cup \{a\})u \equiv Bw \wedge a \notin A \wedge a \notin B) \implies \{a\} \times (B \setminus A) \subseteq ind$
4. $s(B \cup \{a\}) \equiv uv \wedge a \notin B \wedge a \notin \biguplus(v) \implies \{a\} \times (\biguplus(v) \setminus B) \subseteq ind.$
5. $(B \cup \{a\})s \equiv vu \wedge a \notin B \wedge a \notin \biguplus(v) \implies \{a\} \times (\biguplus(v) \setminus B) \subseteq ind.$

Proof. 1. From the definition of $\equiv$, we have $\biguplus(C_1 \dots C_{k-1}) \cap \biguplus(C_k \dots C_n) = \emptyset$. Hence, for all $i = 1, \dots, k-1$ and all $j = k, \dots, n$, we have

$$\pi_{C_i \cup C_j}(A) \equiv \pi_{C_i \cup C_j}(C_1 \dots C_n) = C_iC_j \Rightarrow C_i \times C_j \subseteq ser.$$

Therefore, $\biguplus(C_1 \dots C_{i-1}) \times \biguplus(C_i \dots C_k) \subseteq ser.$

2. For any symbol $a \in A$, from our assumption $u(A \cup \{a\}) \equiv wB$, we first have

$$u(A \cup \{a\}) \div_R A \equiv wB \div_R A$$

Since $wB \div_R A = (w \div_R (A \setminus B))(B \div_R A)$, we have

$$u\{a\} \equiv (w \div_R (A \setminus B))(B \div_R A)$$

where $(B \div_R A) = \lambda$ if $(B \setminus A) = \emptyset$ and $(B \div_R A) = (B \setminus A)$ otherwise. Let $x = (w \div_R (A \setminus B)) \div_R a$. So we have

$$u\{a\} \div_L x \equiv ((w \div_R (A \setminus B))(B \div_R A)) \div_L x = \{a\}(B \div_R A).$$

Notice that we right-cancel an instance of a out of $(w \div_R (A \setminus B))$ to have x, so $u\{a\} \div_L x$ has a form of $v\{a\}$ where $v = u \div_L x$. Hence, we have $v\{a\} \equiv \{a\}(B \div_R A)$.

We consider two possible cases:

Case (i): $(B \div_R A) = \lambda$. We have the trivial case $B \setminus A = \emptyset$. Hence,

$$\{a\} \times (B \setminus A) = \emptyset \subseteq \mathit{ind}$$

Case (ii): $(B \div_R A) \neq \lambda$. Then $(B \setminus A) \neq \emptyset$, let $C = B \setminus A$. We will use induction on $|C|$.

For $|C| = 1$, we have $C = \{b\}$ where $b \neq a$ and $v = \{b\}$. Hence, $\{b\}\{a\} \equiv \{a\}\{b\}$, i.e. $\{b\}\{a\}(\approx \cup \approx^{-1})^*\{a\}\{b\}$. This means there exists a step $\{a, b\} \in \mathbb{S}$ such that $\{b\}\{a\} \approx^{-1} \{a, b\} \approx \{a\}\{b\}$. Thus, $(a, b) \in \mathit{ser} \wedge (b, a) \in \mathit{ser}$. But this implies $(a, b) \in \mathit{ind}$.

Now we need to prove the inductive step, i.e. assuming $v\{a\} \equiv \{a\}(C \cup \{c\})$ where $c \notin C$ and $c \neq a$, we want to show $\{a\} \times (C \cup \{c\}) \subseteq \mathit{ind}$. Using cancellation properties again, we have

$$(v \div_R c)\{a\} \equiv \{a\}C = \{a\}(C \cup \{c\}) \div_R c.$$

This together with the induction hypothesis implies $\{a\} \times C \subseteq \mathit{ind}$. But then $\{a\}(C \cup \{c\}) \div_R C = \{a\}\{c\}$. This forces $(v\{a\}) \div_R C = \{c\}\{a\}$. Hence, $\{c\}\{a\} \equiv \{a\}\{c\}$. Similar to case (i), we obtain $(a, c) \in \mathit{ind}$. Hence, $\{a\} \times (C \cup \{c\}) \subseteq \mathit{ind}$.

3. Dually to (2).

4. We prove this by induction on v. The case of $v = \lambda$ is obvious. When $v = A_k \dots A_1$ $(k > 0)$, by induction hypothesis, we have $\{a\} \times (\biguplus(A_{k-1} \dots A_1) \setminus B) \subseteq \mathit{ind}$. We want to show that $\{a\} \times (A_k \setminus B) \subseteq \mathit{ind}$.

Let $s'(B' \cup \{a\}) = s(B \cup \{a\}) \div_R A_{k-1} \dots A_1$, we get

$$s'(B' \cup \{a\}) \equiv uA_k = uv \div_R A_{k-1} \dots A_1$$

Applying (2) of this lemma, we get $\{a\} \times (A_k \setminus B') \subseteq \mathit{ind}$. But since $B' \subseteq B$, it follows that

$$\{a\} \times (A_k \setminus B) \subseteq \{a\} \times (A_k \setminus B') \subseteq \mathit{ind}.$$

Therefore,

$$\{a\} \times \left(\biguplus(v) \setminus B\right) = \{a\} \times \left(\left(\biguplus(A_1 \ldots A_{k-1}) \cup A_k\right) \setminus B\right) \subseteq ind.$$

5. Dually to (4). □

It is worth noticing that Lemma 5.1 (4),(5) also implies that comtraces belong to paradigm π_3 as classified by Janicki and Koutny in [13] which we will discuss more carefully in Chapter 7. The paradigm basically says that

$$\{a\}\{b\} \equiv \{b\}\{a\} \Rightarrow \{a, b\} \in \mathbb{S}.$$

The intuition comes from the following more general result which explains what it means for steps to be independent.

Proposition 5.2. *For steps $A, B \in \mathbb{S}$, let $C = A \cap B$. If $AB \equiv BA$, then $(A \setminus C) \times (B \setminus C) \subseteq ind$.*

Notice that it immediately follows from this proposition that $A \otimes B \in \mathbb{S}$ where the $\otimes$ operator denotes the symmetric difference operator on sets.

Proof. When $C = \emptyset$, the proposition follows directly from Lemma 5.1 (4) and (5). When $C \neq \emptyset$, it follows that

$$AB \equiv BA \Leftrightarrow (C \cup (A \setminus C))((B \setminus C) \cup C) \equiv (C \cup (B \setminus C))((A \setminus C) \cup C).$$

By cancelling C from the left and then from the right, we get:

$$\begin{aligned}&((C \cup (A \setminus C))((B \setminus C) \cup C) \div_L C) \div_R C\\ \equiv&((C \cup (B \setminus C))((A \setminus C) \cup C) \div_L C) \div_R C.\end{aligned}$$

Hence,

$$(A \setminus C)(B \setminus C) \equiv (B \setminus C)(A \setminus C).$$

Since $(A \setminus C) \cap (B \setminus C) = \emptyset$, by Lemma 5.1 (4) and (5), it follows that

$$(A \setminus C) \times (B \setminus C) \subseteq ind$$

as desired. □

Intuitively, the proposition says that although A and B are not independent steps when $C \neq \emptyset$, $(A \setminus C)$ and $(B \setminus C)$ are.

5.2 Uniqueness of Canonical Representation for Comtraces

As mentioned previously, the *uniqueness* of canonical representation is a consequence of [14, Proposition 3.1] and [14, Proposition 3.1], where the proofs use the properties of stratified order structure. However, the uniqueness of canonical representation can also be proved using only the algebraic properties of comtrace congruence from the last section. The uniqueness follows directly from the following result.

Lemma 5.2. *For each canonical step sequence $u = A_1 \ldots A_k$, we have*

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

The following proof of Lemma 5.2 uses the technical Lemma 5.1.

Proof. Let $A = \{a \mid \exists w \in [u].\ w = C_1 \ldots C_m \wedge a \in C_1\}$. Since $u \in [u]$, $A_1 \subseteq A$. Suppose that $A_1 \neq A$, i.e. we have $a \in A \setminus A_1$ for some a. Hence, there exists $v \in [u]$ such that $v = D_1 \ldots D_n$ and $a \in D_1$. Let j be the least index such that $a \in A_j$, which means $a \notin \biguplus(A_1 \ldots A_{j-1})$. Since $D_1 \ldots D_n \equiv A_1 \ldots A_{j-1} A_j A_{j+1} \ldots A_k$, we can right-cancel $A_{j+1} \ldots A_k$ from both sides of $\equiv$ to get

$$D'_1 \ldots D'_{n'} \equiv A_1 \ldots A_{j-1} A_j \tag{5.1}$$

where $D'_1 \ldots D'_{n'} = D_1 \ldots D_n \div_R A_{j+1} \ldots A_k$ and $a \in D'_1$ because we haven't cancelled the first left $a \in A_j$. We then left-cancel $A_1 \ldots A_{j-1}$ from the equivalence (5.1) to produce

$$D'_1 \ldots D'_{n'} \div_L A_1 \ldots A_{j-1} = D''_1 \ldots D''_{n''} \equiv A_j \tag{5.2}$$

where $a \in D''_1$. There are two cases:

Case (i):

If $n'' = 1$, the equivalence (5.2) becomes $D''_1 \equiv A_j$. So $D''_1 = A_j$. Thus $D''_1 \cap \biguplus(A_1 \ldots A_{j-1}) = \emptyset$, otherwise $D''_1 = A_j$ was not left out after left-cancelling $A_1 \ldots A_{j-1}$ from $D'_1 \ldots D'_{n'}$. Let $B = D'_1 \setminus A_j$, then by Lemma 5.1(5),

$$D''_1 \times (\biguplus(A_1 \ldots A_{j-1}) \setminus B) = A_j \times (\biguplus(A_1 \ldots A_{j-1}) \setminus B) \subseteq ind.$$

Hence,

$$(A_{j-1} \setminus B) \times A_j \subseteq ser \tag{5.3}$$

We next want to show $B \times A_j \subseteq ser$ to conclude that $A_{j-1} \times A_j \subseteq ser$. Observe that

$$D''_1 = D'_1 \ldots D'_{n'} \div_L A_1 \ldots A_{j-1} = (D'_1 \ldots D'_{n'} \div_L D'_2 \ldots D'_{n'}) \div_L B.$$

Hence, $\biguplus(D'_2 \ldots D'_{n'}) \cap D''_1 = \biguplus(D'_2 \ldots D'_{n'}) \cap A_j = \emptyset$. Right-cancelling $D'_2 \ldots D'_{n'}$ from both sides of $\equiv$ of the equivalence (5.1) produces

$$D'_1 \equiv uA_j = A_1 \ldots A_{j-1}A_j \div_R D'_2 \ldots D'_{n'}$$

where $u = A_1 \ldots A_{j-1} \div_R D'_2 \ldots D'_{n'}$. Since $\biguplus(u) = D'_1 \setminus A_j = B$, by Lemma 5.1(1) we conclude

$$B \times A_j = \biguplus(u) \times A_j \subseteq ser \tag{5.4}$$

From the results (5.3) and (5.4), we conclude that $A_{j-1} \times A_j \subseteq ser$. However, since $A_1 \ldots A_k$ is canonical, $A_1 \ldots A_j$ is also canonical. By Proposition 4.1, it follows $A_{j-1} \times A_j \not\subseteq ser$, a contradiction.

Case (ii):

If $n'' > 1$, the equivalence (5.2) becomes $D''_1 \ldots D''_{n''} \equiv A_j$. By Lemma 5.1(1), we obtain $D''_1 \times (A_j \setminus D''_1) = D''_1 \times \biguplus(D''_2 \ldots D''_{n''}) \subseteq ser$. We also have $D''_1 \cap \biguplus(A_1 \ldots A_{j-1}) = \emptyset$, otherwise D''_1 was not left out after left-cancelling $A_1 \ldots A_{j-1}$ from $D'_1 \ldots D'_{n'}$. Let $F = D'_1 \setminus D''_1$, then by Lemma 5.1(5) $D''_1 \times (\biguplus(A_1 \ldots A_{j-1}) \setminus F) \subseteq ind$. So we conclude

$$(A_{j-1} \setminus F) \times D''_1 \subseteq ser \tag{5.5}$$

To show $A_{j-1} \times D''_1 \subseteq ser$, it suffices to show that $F \times D''_1 \subseteq ser$. We first show $D''_1 \cap \biguplus(D'_2 \ldots D'_{n'}) = \emptyset$. For each element $e \in D''_1$, since $D''_1 \cap \biguplus(A_1 \ldots A_{j-1}) = \emptyset$, we have $|D'_1 \ldots D'_{n'}|_e = |A_1 \ldots A_j|_e = |A_j|_e = 1$. This shows $D''_1 \cap \biguplus(D'_2 \ldots D'_{n'}) = \emptyset$.

Hence, right-cancelling $D'_2 \ldots D'_{n'}$ from both sides of $\equiv$ of the equivalence (5.1) produces

$$D'_1 = F \cup D''_1 \equiv vD''_1 = A_1 \ldots A_{j-1}A_j \div_R D'_2 \ldots D'_{n'}.$$

From $F \cup D''_1 \equiv vD''_1$, it follows that $\biguplus(v) = F$. By Lemma 5.1(1), we then conclude

$$E \times D''_1 = \biguplus(v) \times D''_1 \subseteq ser \tag{5.6}$$

From the results (5.5) and (5.6), we have $A_{j-1} \times D''_1 \subseteq ser$. However, by Proposition 4.1, this contradicts that $A_1 \dots A_j$ is canonical, since $D''_1 \subseteq A_j$ and $D''_1 \times (A_j \setminus D''_1) \subseteq ser$.

Since both cases lead to contradiction, we conclude $A_1 = A$. □

The above lemma does not hold for an arbitrary absorbing monoid. For both two canonical representations of $[\{a,b\}\{c\}]$ from Example 4.1, namely $\{a,b\}\{c\}$ and $\{a,c\}\{b\}$, we have $A = \{a \mid \exists w \in [u].\ w = C_1 \dots C_m \wedge a \in C_1\} = \{a,b,c\} \notin \mathbb{S}$. Adding A to the set of possible steps $\mathbb{S}$ does not help as we still have $A \neq \{a,b\}$ and $A \neq \{a,c\}$.

Theorem 5.1. *For every comtrace $t \in \mathbb{S}^*/\equiv$ there exists exactly one canonical step sequence u representing t.*

Proof. The existence follows from Theorem 4.3. We only need to show uniqueness. 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 induction on $k = |u|$ we will show that $u = v$. By Lemma 5.2, we have $B_1 = A_1$. If $k = 1$, this ends the proof. Otherwise, let $u' = A_2 \dots A_k$ and $w' = B_2 \dots B_m$ and u', v' are both canonical step sequences of $[u']$. Since $|u'| < |u|$, by induction hypothesis, we obtain $A_i = B_i$ for $i = 2, \dots, k$ and $k = m$. □

When $ind = ser = sim$, Theorem 5.1 corresponds to the Foata normal form theorem, which we survey in Theorems 4.1 and 4.2 of this thesis. To clarify this point, we analyse the following form of Foata normal theorem, characterised by Volker Diekert in [7], where Diekert provides a proof based on complete semi-Thue systems. A step $F \in \mathbb{S}$ is defined to be *elementary* if $(a,b) \in ind$ for all $a, b \in F, a \neq b$. Notice that each elementary step A_i can be seen as a partial ordered set $(A_i, \emptyset)$. Thus, by the Szpilrajn Theorem, we can construct the Mazurkiewicz trace $[A_i]$ to be the set of all sequences which represent all total order extension of $(A_i, \emptyset)$ (see Section 9.1 for more discussion on relationship between partial orders and Mazurkiewicz traces). The Foata normal form theorem can then be stated as follows.

Proposition 5.3 ([7]). *Let $[s]$ be a Mazurkiewicz trace over a concurrent alphabet (X, ind). There exists exactly one sequence of elementary steps $(A_1, \dots, A_k)$ such that*

$[s] = [A_1]\hat{\circ}\ldots\hat{\circ}[A_k]$ and for all $i \geq 2$, for all $b \in A_i$, there is some $a \in A_{i-1}$ with $(a,b) \notin ser$.

Proof. Assume that $s = x_1 \ldots x_n$. By Theorem 5.1, there exists a step sequence $u = A_1 \ldots A_k$ defined as the canonical step sequence of the comtrace $[\{x_1\}\ldots\{x_n\}]_{ser}$ over the concurrent alphabet (X, sim, ser) as in Theorem 5.1, where $sim = ser = ind$. We observe that all steps A_i are elementary since $ind = sim$. So for each $b \in A_i$,

$$\{b\} \times (A_i \setminus \{b\}) \subseteq sim = ser.$$

Hence, by Proposition 4.1, $A_{i-1} \times \{b\} \nsubseteq ser$. So there is some $(a,b) \in A_{i-1} \times \{b\}$ such that $(a,b) \notin ser$.

By Proposition 3.2, the comtrace $[\{x_1\}\ldots\{x_n\}]_{ser}$ can be represented by the Mazurkiewicz trace $[s] = [x_1 \ldots x_n] = [A_1]\hat{\circ}\ldots\hat{\circ}[A_k]$ as required. □

Notice that Theorems 4.1 and 4.2 are direct consequences of Proposition 5.3. Although a sequence of elementary steps $A_1 \ldots A_k$ is *not* an element of the trace $[s]$, it is the canonical step sequence of the comtrace representing the trace $[s]$. This is another reason suggesting that the notion of comtraces is a convenient and intuitive generalisation of Mazurkiewicz traces.

Chapter 6

Comtrace Languages

Let $\theta = (E, sim, ser)$ be a comtrace alphabet and $\mathbb{S}$ be the set of all possible steps over θ. Any subset L of $\mathbb{S}^*$ is a *step sequence language* over θ, while any subset $\mathcal{L}$ of $\mathbb{S}^*/\equiv_{ser}$ is a *comtrace language* over θ.

For any step sequence language L, we define a comtrace language $[L]_\theta$ (or just $[L]$) as:

$$[L] \overset{df}{=} \{[u] \mid u \in L\} \tag{6.1}$$

The comtrace language $[L]$ is called *generated* by L.

For any comtrace language $\mathcal{L}$, we define

$$\bigcup \mathcal{L} \overset{df}{=} \{u \mid [u] \in \mathcal{L}\} \tag{6.2}$$

Given step sequence languages L_1, L_2 and comtrace languages $\mathcal{L}_1, \mathcal{L}_2$ over the alphabet θ, the *composition of languages* are defined as following:

$$L_1 L_2 \overset{df}{=} \{s_1 \circ s_2 \mid s_1 \in L_1 \wedge s_2 \in L_2\} \tag{6.3}$$

$$\mathcal{L}_1 \mathcal{L}_2 \overset{df}{=} \{t_1 \hat{\circ}\, t_2 \mid t_1 \in \mathcal{L}_1 \wedge t_2 \in \mathcal{L}_2\} \tag{6.4}$$

(Recall $\circ$ and $\hat{\circ}$ denote the operators for step sequence monoids and trace monoids respectively.)

We let L^* and $\mathcal{L}^*$ denote the *iteration* of the step sequence language L and the trace language $\mathcal{L}$ where

$$L^* \overset{df}{=} \bigcup_0^\infty L^n \text{ where } L^0 \overset{df}{=} \{\lambda\} \text{ and } L^{n+1} \overset{df}{=} L^n L \tag{6.5}$$

$$\mathcal{L}^* \overset{df}{=} \bigcup_0^\infty \mathcal{L}^n \text{ where } \mathcal{L}^0 \overset{df}{=} \{[\lambda]\} \text{ and } \mathcal{L}^{n+1} \overset{df}{=} \mathcal{L}^n \mathcal{L} \tag{6.6}$$

Since comtrace languages are sets, the standard set operations as union, intersection, difference, etc. can be used. The following result is a direct consequence of the comtrace language definition and the properties of comtrace composition "$\odot$".

Proposition 6.1. *Let L, L_1, L_2 and L_i for $i \in I$ be step sequence languages, and let $\mathcal{L}$ be a comtrace language. Then :*

1. $[\emptyset] = \emptyset$

2. $[L_1][L_2] = [L_1 L_2]$

3. $L_1 \subseteq L_2 \Rightarrow [L_1] \subseteq [L_2]$

4. $L \subseteq \bigcup[L]$

5. $\mathcal{L} = [\bigcup \mathcal{L}]$

6. $[L_1] \cup [L_2] = [L_1 \cup L_2]$

7. $\bigcup_{i\in I}[L_i] = [\bigcup_{i\in I} L_i]$

8. $[L]^* = [L^*]$.

Proof. 1. From (6.1), it follows that $[\emptyset] = \{[u] \mid u \in \emptyset\} = \emptyset$.

2.

$$\begin{aligned}
& [L_1][L_2] \\
= & \qquad \langle \text{ From (6.4) } \rangle \\
& \{[u_1] \odot [u_2] \mid [u_1] \in [L_1] \wedge [u_2] \in [L_2]\} \\
= & \qquad \langle \text{ From definition of } \odot \rangle \\
& \{[u_1 u_2] \mid [u_1] \in [L_1] \wedge [u_2] \in [L_2]\} \\
= & \qquad \langle \text{ From (6.1) } \rangle \\
& \{[u_1 u_2] \mid u_1 \in L_1 \wedge u_2 \in L_2\} \\
= & \qquad \langle \text{ From (6.3) } \rangle \\
& \{[u_1 u_2] \mid u_1 u_2 \in L_1 L_2\} \\
= & \qquad \langle \text{ From (6.1) } \rangle \\
& [L_1 L_2]
\end{aligned}$$

3. Assuming that $L_1 \subseteq L_2$, we want to show $[L_1] \subseteq [L_2]$. Assume $[t] \in [L_1]$. It suffices to show $[t] \in [L_2]$.

$$\begin{array}{ll}
 & [t] \in [L_1] \\
\Longrightarrow & \qquad \langle \text{ By } (6.1) \rangle \\
 & t \in L_1 \\
\Longrightarrow & \qquad \langle \text{ Since } L_1 \subseteq L_2 \rangle \\
 & t \in L_2 \\
\Longrightarrow & \qquad \langle \text{ By } (6.1) \rangle \\
 & [t] \in [L_2]
\end{array}$$

4. Assuming $t \in L$, we want to show $t \in \bigcup[L]$.

$$\begin{array}{ll}
 & t \in L \\
\Longrightarrow & \qquad \langle \text{ By the definition of comtraces } \rangle \\
 & t \in L \ \wedge \ t \in [t] \\
\Longrightarrow & \qquad \langle \text{ By the definition of set-theoretical union } \rangle \\
 & t \in \bigcup\{[u] \mid u \in L\} \\
\Longrightarrow & \qquad \langle \text{ By } (6.1) \rangle \\
 & t \in \bigcup[L]
\end{array}$$

5. We want to show that for any comtrace $[t]$, $[t] \in \mathcal{L}$ if and only if $[t] \in [\bigcup \mathcal{L}]$.

$$\begin{array}{ll}
 & [t] \in \mathcal{L} \\
\Longleftrightarrow & \qquad \langle \text{ By the definition of comtraces } \rangle \\
 & t \in [t] \in \mathcal{L} \\
\Longleftrightarrow & \qquad \langle \text{ By the definition of set-theoretical union } \rangle \\
 & t \in \bigcup \mathcal{L} \\
\Longleftrightarrow & \qquad \langle \text{ From } (6.2) \rangle \\
 & [t] \in \{[u] \mid u \in \bigcup \mathcal{L}\} \\
\Longleftrightarrow & \qquad \langle \text{ From } (6.1) \rangle \\
 & [t] \in [\bigcup \mathcal{L}]
\end{array}$$

6.

$[L_1] \cup [L_2]$

$=$ $\langle$ From (6.1) $\rangle$

$\{[u] \mid u \in L_1\} \cup \{[u] \mid u \in L_2\}$

$=$ $\langle$ By definition of set-theoretical union $\rangle$

$\{[u] \mid u \in L_1 \;\vee\; u \in L_2\}$

$=$ $\langle$ From definition of set-theoretical union $\rangle$

$\{[u] \mid u \in L_1 \cup L_2\}$

$=$ $\langle$ From (6.1) $\rangle$

$[L_1 \cup L_2]$

7. Notice I is the index set, so it has the form $I = \{i \mid 1 \leq i \leq n\}$. Hence, we will prove (7) by induction on n. When $n = 0$, it follows that $I = \emptyset$. Hence,

$\bigcup_{i\in\emptyset}[L_i]$

$=$ $\langle$ By definition of set-theoretical union $\rangle$

$\emptyset$

$=$ $\langle$ From (6.1) $\rangle$

$[\bigcup_{i\in\emptyset} L_i]$

When $n > 0$, we want to show that $\bigcup_{i=1}^{n}[L_i] = [\bigcup_{i=1}^{n} L_i]$.

$[\bigcup_{i=1}^{n} L_i]$

$=$ $\langle$ By definition of set-theoretical union $\rangle$

$[(\bigcup_{i=1}^{n-1} L_i) \cup L_n]$

$=$ $\langle$ From (6.1) $\rangle$

$\{[u] \mid u \in (\bigcup_{i=1}^{n-1} L_i) \cup L_n\}$

$=$ $\langle$ By the properties of set-theoretical union $\rangle$

$\{[u] \mid u \in \bigcup_{i=1}^{n-1} L_i\} \cup \{[u] \mid u \in L_n\}$

$=$ $\langle$ From (6.1) $\rangle$

$[\bigcup_{i=1}^{n-1} L_i] \cup [L_n]$

$=$ $\langle$ By induction hypothesis $\rangle$

$(\bigcup_{i=1}^{n-1}[L_i]) \cup [L_n]$

$=$ $\langle$ From (6) $\rangle$

$\bigcup_{i=1}^{n}[L_i]$

8. Observe that $[L]^* = \bigcup_{i=0}^{\infty}[L]^i$ and $[L^*] = [\bigcup_{i=0}^{\infty} L^i]$. Since we only deal with *finite* step sequences, it suffices to show that $[L]^i = [L^i]$ for every i. We proceed

by induction on i. When $i = 0$, it follows that

$$
\begin{array}{ll}
 & [L]^0 \\
= & \qquad \langle \text{ By (6.6) } \rangle \\
 & \{[\lambda]\} \\
= & \qquad \langle \text{ By (6.1) } \rangle \\
 & \{[u] | u \in \{\lambda\}\} \\
= & \qquad \langle \text{ By (6.1) } \rangle \\
 & [\{\lambda\}] \\
= & \qquad \langle \text{ By (6.5) } \rangle \\
 & [L^0]
\end{array}
$$

When $i > 0$, we want to show $[L]^i = [L^i]$.

$$
\begin{array}{ll}
 & [L]^i \\
= & \qquad \langle \text{ By (6.6) } \rangle \\
 & [L]^{i-1}[L] \\
= & \qquad \langle \text{ By induction hypothesis } \rangle \\
 & [L^{i-1}][L] \\
= & \qquad \langle \text{ By (2) } \rangle \\
 & [L^{i-1}L] \\
= & \qquad \langle \text{ By (6.5) } \rangle \\
 & [L^i]
\end{array}
$$

□

Comtrace languages provide a bridge between operational and structural, i.e. comtrace, semantics. In other words, if a step sequence language L describes an operational semantics of a given concurrent system, we only need to derive (E, sim, ser) from the system, and $[L]$ defines the structural semantics of the system.

Example 6.1. Consider the following simple concurrent system *Priority*, which comprises two sequential subsystems such that

- the first subsystem can cyclically engage in event a followed by event b,
- the second subsystem can cyclically engage in event b or in event c,
- the two systems synchronise by means of handshake communication,

- there is a priority constraint stating that if it is possible to execute event b then c must not be executed.

This example has often been analysed in the literature (cf. [16]), usually under the interpretation that a = *'Error Message'*, b = *'Stop And Restart'*, and c = *'Some Action'*. It can be formally specified in various notations including Priority and Inhibitor Nets (cf. [12, 15]). Its operational semantics (easily found in any model) can be defined by the following language of step sequences

$$L_{\mathit{Priority}} \stackrel{df}{=} \mathit{Pref}((\{c\}^* \cup \{a\}\{b\} \cup \{a,c\}\{b\})^*)\,,$$

where *Pref*(L) denotes the prefix closure of the language L, i.e.,

$$\mathit{Pref}(L) \stackrel{df}{=} \bigcup_{w \in L} \{u \in L \mid \exists v.\ uv = w\}.$$

The rules for deriving the concurrent alphabet $(E, \mathit{sim}, \mathit{ser})$ depend on the model, and for *Priority*, the set of possible steps is

$$\mathbb{S} = \{\{a\}, \{b\}, \{c\}, \{a, c\}\},$$

and $\mathit{ser} = \{(c,a)\}$ and $\mathit{ser} = \{(a,c),(c,a)\}$. Then, $[L_{\mathit{Priority}}]$ defines the structural comtrace semantics of *Priority*. For instance,

$$[\{a,c\}\{b\}] = \{\{c\}\{a\}\{b\}, \{a,c\}\{b\}\} \in [L_{\mathit{Priority}}].$$

Chapter 7

Paradigms of Concurrency

The general theory of concurrency developed in [13] 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, and 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 [14] 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:

Definition 7.1. Let (E, sim, ser, inl) be a generalised comtrace alphabet. The monoid of generalised comtraces (or comtraces when $inl = \emptyset$) $(\mathbb{S}^*/\equiv, \hat{\circ}, [\lambda])$ conforms to paradigm π_3 if and only if

$$\forall a, b \in E.\ (\{a\}\{b\} \equiv \{b\}\{a\} \Rightarrow \{a, b\} \in \mathbb{S}),$$

conforms to paradigm π_6 if and only if

$$\forall a, b \in E.\ (\{a, b\} \in \mathbb{S} \Rightarrow \{a\}\{b\} \equiv \{b\}\{a\}),$$

and conforms to paradigm π_8 if and only if

$$\forall a, b \in E.\ (\{a\}\{b\} \equiv \{b\}\{a\} \Leftrightarrow \{a, b\} \in \mathbb{S}).$$

Proposition 7.1. *Let $M = (\mathbb{S}^*/\equiv, \hat{\circ}, [\lambda])$ be a comtrace monoid over a comtrace alphabet (E, sim, ser). Then*

1. *M conforms to π_3.*

2. *If π_8 is satisfied, then $ind = ser = sim$.*

Proof. 1. Assume $\{a\}\{b\} \equiv \{b\}\{a\}$ for some $a, b \in E$. Hence, by Definition 3.5, $\{a\}\{b\} \approx^{-1} \{a, b\} \approx \{b\}\{a\}$, i.e. $\{a, b\} \in \mathbb{S}$.

2. Clearly $ind \subseteq ser \subseteq sim$. Let $(a, b) \in sim$. This means $\{a, b\} \in \mathbb{S}$, which, by π_8, implies $\{a\}\{b\} \equiv \{b\}\{a\}$. Hence, by Lemma 5.1(2), $(a, b) \in ind$. □

From Proposition 7.1(1), it follows that comtraces cannot model any concurrent behaviour (history) that does not conform to the paradigm π_3. Since any monoid of comtraces conforms to π_3, we know that if a monoid of comtraces conforms to π_6, then it also conforms to π_8. It also follows from Proposition 3.2 and Proposition 7.1(2) that all comtraces conforming to π_8 can be reduced to equivalent Mazurkiewicz traces.

Generalised comtraces does not conform to π_3. Example 3.5 works as a counterexample, since $\{a\}\{b\} \equiv \{b\}\{a\}$ but $\{a, b\} \notin \mathbb{S}$. In fact, as a language representation of generalised stratified order structures, generalised comtraces conform only to π_1, so they can model any concurrent history that is represented by a set of equivalent step sequences.

Chapter 8

Relational Structures Model of Concurrency

In this chapter, we review the theory of relational structures proposed by Janicki and Koutny [11, 14, 10, 15, 12] to specify concurrent behaviours by using *a pair of relations* instead of a single causality relation. The motivation is that partial orders can sufficiently model the "earlier than" relationship but cannot model the "not later than" relationship. We will give the definitions of *stratified order structure* and *generalised stratified order structure*, and then introduce the intuition and motivation behind these order structures using a detailed example.

8.1 Stratified Order Structure

By a *relational structure* we will mean a triple $T = (X, R_1, R_2)$, where X is a set and R_1 and R_2 are binary relations on X. A relational structure $T' = (X', R_1', R_2')$ is an *extension* of T, denoted as $T \subseteq T'$, if and only if $X = X'$, $R_1 \subseteq R_1'$ and $R_2 \subseteq R_2'$.

Definition 8.1 ([15]). A *stratified order structure* is a relational structure

$$S = (X, \prec, \sqsubset),$$

such that for all $a, b, c \in X$, the following hold:

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: $a \sqsubset b \prec c \vee a \prec b \sqsubset c \implies a \prec c$

When X is finite, S is called a *finite stratified order structure.*

Remark 8.1. The axioms C1–C4 imply that $(X, \prec)$ is a poset 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" relations, provided that the system runs are defined as stratified orders (step sequences).

Stratified order structures were independently introduced in [9] and [12] (the axioms are slightly different from C1–C4, although equivalent). Their comprehensive theory has been presented in [15]. They have been successfully applied to model inhibitor and priority systems, asynchronous races, synthesis problems, etc. (see for example [14, 18, 19, 20, 27] and others).

8.2 Generalised Stratified Order Structure

Stratified order structures can adequately model concurrent histories only when the paradigm π_3 of [13, 15] is satisfied. For the general case, we need *generalised stratified order structures* introduced by Guo and Janicki in [10] also under the assumption that the system runs are defined as stratified orders (step sequences).

Definition 8.2 ([10, 14]). A *generalised stratified order structure* is a relational structure

$$G = (X, <\!\!>, \sqsubset),$$

such that $\sqsubset$ is irreflexive, $<\!\!>$ is symmetric and irreflexive, and the triple

$$S_G = (X, \prec_G, \sqsubset),$$

where $\prec_G = <\!\!> \cap \sqsubset$, is a stratified order structure.

Such relational structure S_G is called the *stratified order structure induced by G.* When X is finite, G is called a *finite generalised stratified order structure.*

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

8.3 Motivating Example

To understand the main motivation and intuition behind the use of stratified order structures and generalised stratified order structures, we will consider the four simple programs in the following example taken from [11].

Example 8.1 ([11]). The programs are written using a mixture of cobegin, coend and a version of concurrent guarded commands.

P1:

```
begin
    int x,y;
    a: begin x:=0; y:=0 end;
    cobegin
        b: x:=x+1, c: y:=y+1
    coend
end.
```

P2:

```
begin
    int x,y;
    a: begin x:=0; y:=0 end;
    cobegin
        b: x=0 → y:=y+1, c: x:=x+1
    coend
end.
```

P3:

```
 begin
    int x,y;
    a: begin x:=0; y:=0 end;
    cobegin
        b: y=0 → x:=x+1, c: x=0 → y:=y+1
    coend
```

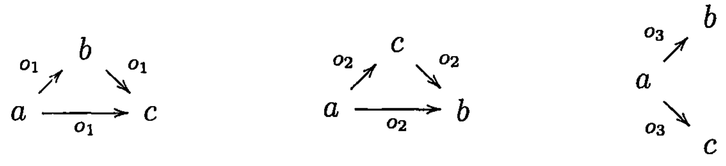
```
end.
```

P4:

```
begin
   int x;
   a: x:=0;
   cobegin
      b: x:=x+1, c: x:=x+2
   coend
end.
```

Each program is a different composition of three events (actions) called a, b, and c (a_i, b_i, c_i, $i = 1, ..., 4$, to be exact, but a restriction to a, b, c, does not change the validity of the analysis below, while simplifying the notation). Alternative models of these programs are shown Figure 8.1.

Let $obs(P_i)$ denote the set of all program runs involving the actions a, b, c that can be observed. Assume that simultaneous executions can be observed. In this simple case all runs (or observations) can be modelled by *step sequences* with simultaneous execution of a_1,...,a_n denoted by the step $\{a_1, \ldots, a_n\}$. Let us denote $o_1 = \{a\}\{b\}\{c\}$, $o_2 = \{a\}\{c\}\{b\}$, $o_3 = \{a\}\{b, c\}$. Each o_i can be equivalently seen as a stratified partial order $o_i = (\{a, b, c\}, \xrightarrow{o_i})$ (see Section 9.2 for formal discussion of the relationship between step sequences and stratified orders) where:



We can now write $obs(P_1) = \{o_1, o_2, o_3\}$, $obs(P_2) = \{o_1, o_3\}$, $obs(P_3) = \{o_3\}$, $obs(P_4) = \{o_1, o_2\}$. Note that for every $i = 1, ..., 4$, all runs from the set $obs(P_i)$ yield exactly the same outcome. Hence, each $obs(P_i)$ is called the *concurrent history* of P_i.

An abstract model of such an outcome is called a *concurrent behaviour*, and now we will discuss how causality, weak causality and commutativity relations are used to construct concurrent behaviour.

Program P_1:

In the set $obs(P_1)$, for each run, a always precedes both b and c, and there is no *causal* relationship between b and c. This *causality* relation, $\prec$, is the partial order defined as $\prec = \{(a,b),(a,c)\}$. In general $\prec$ is defined by: $x \prec y$ if and only if for each run o we have $x \overset{o}{\rightarrow} y$. Hence for P_1, $\prec$ is the intersection of o_1, o_2 and o_3, and $\{o_1,o_2,o_3\}$ is the set of all stratified extensions of the relation $\prec$.

Thus in this case the causality relation $\prec$ models the concurrent behaviour corresponding to the set of (equivalent) runs $obs(P_1)$. We will say that $obs(P_1)$ and $\prec$ are *tantamount* and write $obs(P_1) \asymp \{\prec\}$ or $obs(P_1) \asymp (\{a,b,c\},\prec)$. Having $obs(P_1)$ one may construct $\prec$ (as an intersection), and hence construct $obs(P_4)$ (as the set of all stratified extensions). This is a classical case of the "true" concurrency approach, where concurrent behaviour is modelled by a causality relation.

Before considering the remaining cases, note that the causality relation $\prec$ is exactly the same in all four cases, i.e. $\prec_i = \{(a,b),(a,c)\}$, for $i = 1,...,4$, so we may omit the index i.

Programs P_2 and P_3:

To deal with $obs(P_2)$ and $obs(P_3)$, $\prec$ is insufficient because $o_2 \notin obs(P_2)$ and $o_1, o_2 \notin obs(P_2)$. Thus, we need another relation, $\sqsubset$, called *weak causality*, defined in this context as $x \sqsubset y$ if and only if for each run o we have $\neg(y \overset{o}{\rightarrow} x)$ (*x is never executed after y*). For our four cases we have $\sqsubset_2 = \{(a,b),(a,c),(b,c)\}$, $\sqsubset_1 = \sqsubset_4 = \prec$, and $\sqsubset_3 = \{(a,b),(a,c),(b,c),(c,b)\}$. Notice again that for $i = 2,3$, the pair of relations $\{\prec,\sqsubset_i\}$ and the set $obs(P_i)$ are equivalent in the sense that each is definable from the other. (The set $obs(P_i)$ can be defined as the greatest set PO of partial orders built from a, b and c satisfying $x \prec y \Rightarrow \forall o \in PO.\ x \overset{o}{\rightarrow} y$ and $x \sqsubset_i y \Rightarrow \forall o \in PO.\ \neg(y \overset{o}{\rightarrow} x)$.)

Hence again in these cases ($i = 2,3$) $obs(P_i)$ and $\{\prec,\sqsubset_i\}$ are *tantamount*, $obs(P_i) \asymp \{\prec,\sqsubset_i\}$, and so the pair $\{\prec,\sqsubset_i\}$, $i = 2,3$, models the concurrent behaviour described by $obs(P_i)$. Note that $\sqsubset_i$ alone is not sufficient, since (for instance) $obs(P_2)$ and $obs(P_2) \cup \{\{a,b,c\}\}$ define the same $\sqsubset$.

Program P_4:

The causality relation $\prec$ does not model the concurrent behaviour of P_4 correctly[1] since o_3 does not belong to $obs(P_1)$. Let $\Diamond$ be a symmetric relation, called *commutativity*, defined as $x \Diamond y$ if and only if for each run o either $x \xrightarrow{o} y$ or $y \xrightarrow{o} x$. For the set $obs(P_4)$, the relation $\Diamond_4$ looks like $\Diamond_4 = \{(a,b),(b,a),(a,c),(c,a),(b,c),(c,b)\}$. The pair of relations $\{\Diamond_4, \prec\}$ and the set $obs(P_4)$ are equivalent in the sense that each is definable from the other. (The set $obs(P_4)$ is the greatest set PO of partial orders built from a, b and c satisfying $x \Diamond_4 y \Rightarrow \forall o \in PO.\ x \xrightarrow{o} y \vee y \xrightarrow{o} x$ and $x < y \Rightarrow \forall o \in PO.\ x \xrightarrow{o} y$.) In other words, $obs(P_4)$ and $\{\Diamond_4, \prec\}$ are *tantamount*, $obs(P_4) \asymp \{\Diamond_4, \prec\}$, so we may say that in this case the relations $\{\Diamond_4, <\}$ model the concurrent behaviour described by $obs(P_4)$.

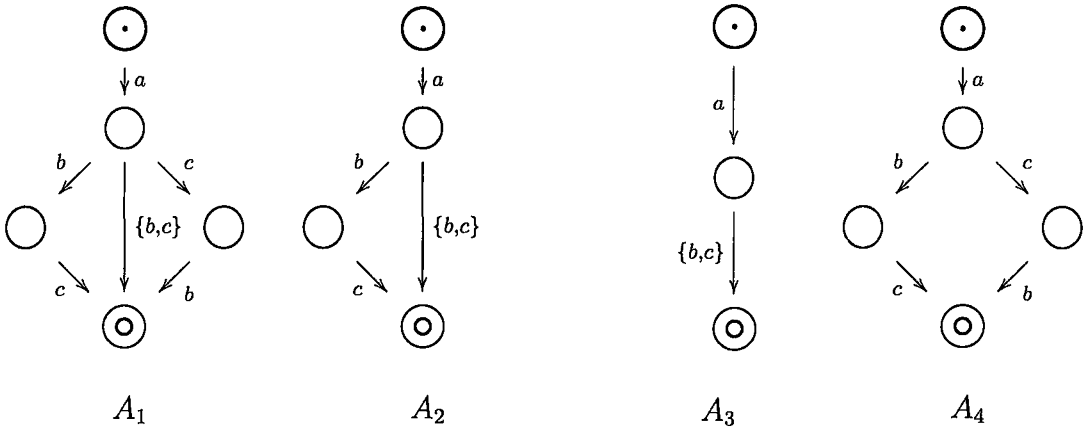
Note also that $\Diamond_1 = \prec \cup \prec^{-1}$ and the pair $\{\Diamond_1, \prec\}$ also models the concurrent behaviour described by $obs(P_1)$.

The state transition model A_i of each P_i and their respective concurrent histories and concurrent behaviours are summarised in Figure 8.1. Thus, we can make the following observations:

1. $obs(P_1)$ can be modelled by the relation $\prec$ alone, and $obs(P_1) \asymp \{\prec\}$.
2. $obs(P_i)$, for $i = 1,2,3$ can also be modelled by appropriate pairs of relations $\{\prec, \sqsubset_i\}$, and $obs(P_i) \asymp \{\prec, \sqsubset_i\}$.
3. all sets of observations $obs(P_i)$, for $i = 1,2,3,4$ are modelled by appropriate pairs of relations $\{\Diamond_i, \sqsubset_i\}$, and $obs(P_i) \asymp \{\Diamond_i, \sqsubset_i\}$.

Note that the relations $\prec$, $\Diamond$, $\sqsubset$ are not independent, since it can be proved (see [13]) that $< = \Diamond \cap \sqsubset$. The underlying idea is very intuitive. Since the relation $\Diamond$ means "earlier than or later than" and the relation $\sqsubset$ means "not later than", it follows the intersection means the "earlier than" relation $\prec$.

[1] Unless we assume that simultaneity is not allowed, or not observed, in which case $obs(P_1) = obs(P_4) = \{o_1, o_2\}$, $obs(P_2) = \{o_1\}$, $obs(P_3) = \emptyset$.



$\prec_1 = \{(a,b),(a,c)\}$
$\sqsubset_1 = \{(a,b),(a,c)\}$
$\diamond_1 = \sqsubset_1 \cup \sqsubset_1^{-1}$
$obs(P_1) \asymp obs(A_1)$
$\asymp \{\prec_1\} \asymp \{\prec_1, \sqsubset_1\}$
$\asymp \{\diamond_1, \sqsubset_1\}$

$\prec_2 = \{(a,b),(a,c)\}$
$\sqsubset_2 = \{(a,b),(a,c),(b,c)\}$
$\diamond_2 = \sqsubset_2 \cup \sqsubset_2^{-1}$
$obs(P_2) \asymp obs(A_2)$
$\asymp \{\prec_2, \sqsubset_2\}$
$\asymp \{\diamond_2, \sqsubset_2\}$

$\prec_3 = \{(a,b),(a,c)\}$
$\sqsubset_3 = \{(a,b),(a,c),$
$(b,c),(c,b)\}$
$\diamond_3 = \sqsubset_3 \cup \sqsubset_3^{-1}$
$obs(P_3) \asymp obs(A_3)$
$\asymp \{\prec_3, \sqsubset_3\}$
$\asymp \{\diamond_3, \sqsubset_3\}$

$\prec_4 = \{(a,b),(a,c)\}$
$\sqsubset_4 = \{(a,b),(a,c)\}$
$\diamond_4 = \{(a,b),(b,a),$
$(a,c),(c,a),(b,c),(c,b)\}$
$obs(P_4) \asymp obs(A_4)$
$\asymp \{\diamond_4, \sqsubset_4\}$

Figure 8.1: Examples of *causality*, *weak causality*, and *commutativity*. Each program P_i can be modelled by a labelled transition system (automaton) A_i. We use the step $\{b,c\}$ to denote *simultaneous* execution of a and b.

Chapter 9

Relational Representation of Mazurkiewicz Traces and Comtraces

It is well known that Mazurkiewicz traces can be interpreted as a formal language representation of partial orders. In fact, each comtrace uniquely determines a finite stratified order structure and each finite stratified order structure can be represented by a comtrace. In this chapter, we will study this relationship in more detail.

9.1 Partial Orders and Mazurkiewicz Traces

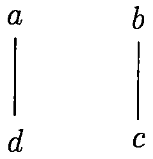
Each trace can be interpreted as a finite partial order. Let $t = \{x_1, ..., x_k\}$ be a trace, and let $\prec_{x_i}$ be a total order defined by a sequence x_i, $i = 1, ..., k$ then the set $\{\prec_{x_1}, ..., \prec_{x_n}\}$ is the set of all total order extensions of $\prec_t$. By the Szpilrajn Theorem, we know that the partial order generated by the trace t can then be defined as $\prec_t = \bigcap_{i=1}^{k} \prec_{x_i}$.

Conversely, each finite partial order uniquely determines a trace. 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 order 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 $(X, \frown_{\prec})$.

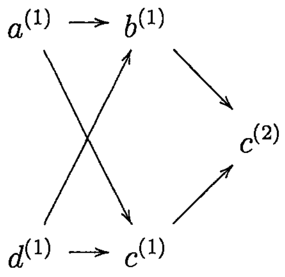
Example 9.1. Let $E = \{a, b, c, d\}$ where a, b, c and d are four atomic operations defined as follows:

$$a: \ x \leftarrow x + y, \quad b: \ y \leftarrow x + w, \quad c: \ y \leftarrow y + z, \quad d: \ w \leftarrow 2y + z.$$

Assuming simultaneous reading and exclusive writing, then a and d can be executed simultaneously, and so can the pair of actions b and c. The independency relation can be expressed as the following undirected graph:



Given a sequence of operations $s = dabcc$, we can enumerate the operations of s to get the enumerated sequence $\overline{s} = d^{(1)}a^{(1)}b^{(1)}c^{(1)}c^{(2)}$. By interpreting the lack of order as independency, we can build a causality partial order $\prec_{[s]}$ for $\overline{s}$ (for simplicity, we do not draw arrows resulting from transitivity):



For example, we have $a^{(1)} \frown_{\prec_t} d^{(1)}$ because a and d are independent operations.

The trace

$$[s] = \{dabcc, adbcc, dacbc, adcbc\}$$

defines all the *total order extensions* of the partial order $\prec_{[s]}$ because each sequence in $[s]$ induces a total order on the set of event occurrences $\{a^{(1)}, b^{(1)}, c^{(1)}, c^{(2)}, d^{(1)}\}$:

- *dabcc* induces $\prec_{dabcc}$: $d^{(1)} \rightarrow a^{(1)} \rightarrow b^{(1)} \rightarrow c^{(1)} \rightarrow c^{(2)}$
- *adbcc* induces $\prec_{adbcc}$: $a^{(1)} \rightarrow d^{(1)} \rightarrow b^{(1)} \rightarrow c^{(1)} \rightarrow c^{(2)}$
- *dacbc* induces $\prec_{dacbc}$: $d^{(1)} \rightarrow a^{(1)} \rightarrow c^{(1)} \rightarrow b^{(1)} \rightarrow c^{(2)}$
- *adcbc* induces $\prec_{adcbc}$: $a^{(1)} \rightarrow d^{(1)} \rightarrow c^{(1)} \rightarrow b^{(1)} \rightarrow c^{(2)}$

and we can verify that

$$\prec_{[s]} = \bigcap\{\prec_{dabcc}, \prec_{adbcc}, \prec_{dacbc}, \prec_{adcbc}\}.$$

9.2 Stratified Order Structure Representation of Comtraces

Analogous to the relationship between Mazurkiewicz traces and partial orders, comtraces can be seen as a formal language representation of finite stratified order structures. In [14], Janicki and Koutny showed that each comtrace uniquely determines a finite stratified order structure; however, it is not intuitive why their construction from comtraces to stratified order structures works. Hence, we will introduce more techniques to analyse this construction where the keys are the three notions of *non-serialisable steps* and the utilisation of the induction proof techniques.

Definition 9.1 ([15]). Let $S = (X, \prec, \sqsubset)$ be a stratified order structure. A *stratified* order $\lhd$ on X is a *stratified order extension* of S if for all $\alpha, \beta \in X$,

$$\alpha \prec \beta \implies \alpha \lhd \beta$$
$$\alpha \sqsubset \beta \implies \alpha \lhd^{\frown} \beta$$

The set of all stratified order extensions of S is denoted as $ext(S)$.

Proposition 9.1. *Let u, v be two step sequences over a comtrace alphabet (E, sim, ser) and $u \equiv v$. Then $\Sigma_u = \Sigma_v$.*

Proof. From Proposition 5.1(2), we know that $\equiv$ is *event-preserving*, i.e. for all $e \in E$, we have $|u|_e = |v|_e$. Since the enumeration of events in u and v depends on the multiplicity of event occurrences in u and v, it follows that $\Sigma_u = \Sigma_v$. □

Thus, for a comtrace $t = [u]$ we can define $\Sigma_t = \Sigma_u$.

The intuition of how a unique stratified order structure is constructed from a comtrace is provided in the following example which is analogous to the Example 9.1 for Mazurkiewicz traces.

Example 9.2. Consider a comtrace alphabet $\mathcal{C} = (\{a,b,c\}, sim, ser)$ where

- $sim = \{(a,b),(b,a),(a,c),(c,a)\}$
- $ser = \{(a,b),(b,a),(a,c)\}$

The set of all possible steps is $\{\{a,b\},\{a,c\},\{a\},\{b\},\{c\}\}$.

Consider a step sequence $s_1 = \{a,b\}\{c\}\{a\}$. With respect to the concurrent alphabet $\mathcal{C}$, we have:

$$t = [s_1] = \{\{a,b\}\{c\}\{a\}, \{a\}\{b\}\{c\}\{a\}, \{b\}\{a\}\{c\}\{a\}, \{b\}\{a,c\}\{a\}\}.$$

Since $\Sigma_t = \{a^{(1)}, a^{(2)}, b^{(1)}, c^{(1)}\}$, we can construct the corresponding stratified order for each of the element in t as following (the edges resulting from transitivity are omitted):

- $s_1 = \{a,b\}\{c\}\{a\}$ induces $\lhd_{s_1}$:

$$\begin{tikzcd} a^{(1)} \arrow[dr] & & \\ & c^{(1)} \arrow[r] & a^{(2)} \\ b^{(1)} \arrow[ur] & & \end{tikzcd}$$

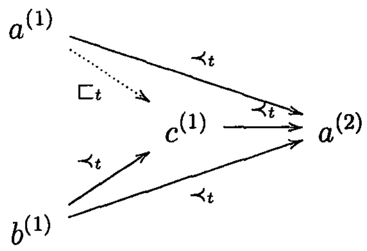
- $s_2 = \{a\}\{b\}\{c\}\{a\}$ induces $\lhd_{s_2}$: $\quad a^{(1)} \rightarrow b^{(1)} \rightarrow c^{(1)} \rightarrow a^{(2)}$
- $s_3 = \{b\}\{a\}\{c\}\{a\}$ induces $\lhd_{s_3}$: $\quad b^{(1)} \rightarrow a^{(1)} \rightarrow c^{(1)} \rightarrow a^{(2)}$
- $s_4 = \{b\}\{a,c\}\{a\}$ induces $\lhd_{s_4}$:

$$\begin{tikzcd} & a^{(1)} \arrow[dr] & \\ b^{(1)} \arrow[ur] \arrow[dr] & & a^{(2)} \\ & c^{(1)} \arrow[ur] & \end{tikzcd}$$

By observing all of the possible Mazurkiewicz traces and the order of event occurrences, we can build the following stratified order structure

$$S_t = (\Sigma_t, \prec_t, \sqsubset_t) = \left(\Sigma_t\, , \bigcap_{s\in t} \lhd_s\, , \bigcap_{s\in t} \lhd_s^{\frown}\right) \tag{9.1}$$

which can be graphically represented as follows (note that the directed edges labelled by $\prec_t$ also denote the $\sqsubset_t$ relation since $\prec_t \subseteq \sqsubset_t$):



We can also check that $ext(S_t) = \{\lhd_s \mid s \in t\}$.

In [14], Janicki and Koutny proposed the notion of $\diamond$*-closure* and used it to construct finite stratified order structures from comtraces. For a relation structure $S = (X, R_1, R_2)$, its $\diamond$-closure is defined as

$$S^{\diamond} = (X, R_1, R_2)^{\diamond} \overset{df}{=} (X, (R_1 \cup R_2)^* \circ R_1 \circ (R_1 \cup R_2)^*, (R_1 \cup R_2)^* \setminus id_X)$$

where $(R_1 \cup R_2)^*$ denotes the *reflexive transitive closure* of $(R_1 \cup R_2)$.

Definition 9.2 ([14]). Let $t = [s]$ be a comtrace over a comtrace alphabet (E, sim, ser). For $\alpha, \beta \in \Sigma_s$, we can define

$$\alpha \prec_s \beta \iff (l(\alpha), l(\beta)) \notin ser \ \wedge \ pos_s(\alpha) < pos_s(\beta),$$
$$\alpha \sqsubset_s \beta \iff (l(\beta), l(\alpha)) \notin ser \ \wedge \ pos_s(\alpha) \leq pos_s(\beta).$$

We let $\varphi_s \overset{df}{=} (\Sigma_s, \prec_s, \sqsubset_s)^{\diamond}$, then the *stratified order structure induced by the trace* $t = [s]$ is

$$\varphi_t = (\Sigma_t, \prec_t, \sqsubset_t) \overset{df}{=} \varphi_s.$$

The fact that φ_t is defined to be φ_s for any $s \in t$ makes sense because of the following results:

Proposition 9.2 (Proposition 4.4 of [14]). *Let s be step sequences over a comtrace alphabet (E, ser, sim). Then φ_s is a stratified order structure.* □

Theorem 9.1 ([14, Theorem 4.10]). *Let r and s be step sequences over a comtrace alphabet (E, ser, sim). Then $\varphi_r = \varphi_s$ if and only if $r \equiv s$.* □

We also know the following invariant properties of the step sequences that belong to the same comtrace:

Proposition 9.3 ([14, Proposition 4.2]). *Let $t = [s]$ be a comtrace over a comtrace alphabet (E, sim, ser). If $\alpha, \beta \in \Sigma_t$, then*

1. $\alpha \prec_s \beta \implies \forall u \in t.\ pos_u(\alpha) < pos_u(\beta)$
2. $\alpha \sqsubset_s \beta \implies \forall u \in t.\ pos_u(\alpha) \leq pos_u(\beta)$.

□

Proposition 9.4. *Let $t = [s]$ be a comtrace over a comtrace alphabet (E, sim, ser) and let $\varphi_t = (\Sigma_t, \prec_t, \sqsubset_t)$ be the stratified order structure induced by t. If $\alpha, \beta \in \Sigma_t$, then*

1. $\alpha \prec_t \beta \implies \forall u \in t.\ pos_u(\alpha) < pos_u(\beta)$
2. $\alpha \sqsubset_t \beta \implies (\alpha \neq \beta \ \wedge\ \forall u \in t.\ pos_u(\alpha) \leq pos_u(\beta))$.

Proof. 1. Assume $\alpha \prec_t \beta$ and let $R = (\prec_s \cup \sqsubset_s)$, then by definition of $\diamond$-closure, we have

$$\alpha \; R \; \alpha_1 \; R \; \ldots \; R \; \alpha_m \prec_s \beta_1 \; R \; \ldots \; R \; \beta_n \; R \; \beta$$

for some $m, n \geq 0$.

By Proposition 9.3, we know that if $\gamma \; R \; \delta$ then for all $u \in t$, we have $pos_u(\gamma) \leq pos_u(\delta)$ and if $\alpha_m \prec_s \beta_1$ then $pos_u(\alpha_m) < pos_u(\beta_1)$. Hence, for all $u \in t$, we have

$$pos_u(\alpha) \leq pos_u(\alpha_1) \leq \ldots \leq pos_u(\alpha_m) < pos_u(\beta_1) \leq \cdots \leq pos_u(\beta_n) \leq pos_u(\beta).$$

Hence, for all $u \in t$ we have $pos_u(\alpha) < pos_u(\beta)$ as desired.

2. Assume $\alpha \sqsubset_t \beta$, then by the definition of $\diamond$-closure, we have $\alpha \neq \beta$ and

$$\alpha \; R \; \alpha_1 \; R \; \ldots \; R \; \alpha_m \; R \; \beta$$

Similarly to (1), we can conclude that for all $u \in t$, we have $pos_u(\alpha) \leq pos_u(\beta)$ as desired. □

Although the implications of Proposition 9.4 are straightforward consequences of how φ_t is defined, the converses are non-trivial results, which we prove in Proposition 9.8. Before doing so, we need some new definitions and preliminary results.

Let A be a step over a comtrace alphabet (E, sim, ser) and let $a \in A$ then:

- The step A is called *serialisable* if and only if

$$\exists B, C \in \widehat{\mathscr{P}}A.\ (B \cup C = A \ \wedge\ B \times C \subseteq ser).$$

  The step A is called *non-serialisable* if and only if A is not serialisable, i.e.

$$\forall B, C \in \widehat{\mathscr{P}}A.\ (B \cup C = A \implies B \times C \nsubseteq ser).$$

  Obviously for a non-serialisable step, we have $[A] = \{A\}$. (Note that every non-serialisable step is a synchronous step as defined in Definition 3.6.)

- The step A is called *serialisable to the left of a* if and only if

$$\exists B, C \in \widehat{\mathscr{P}}A.\ (B \cup C = A \ \wedge\ a \in B \ \wedge\ B \times C \subseteq ser).$$

  The step A is called *non-serialisable to the left of a* if and only if A is not serialisable to the left of a, i.e.

$$\forall B, C \in \widehat{\mathscr{P}}A.\ ((B \cup C = A \ \wedge\ a \in B) \implies B \times C \nsubseteq ser).$$

- The step A is called *serialisable to the right of a* if and only if

$$\exists B, C \in \widehat{\mathscr{P}}A.\ (B \cup C = A \ \wedge\ a \in C \ \wedge\ B \times C \subseteq ser).$$

  The step A is called *non-serialisable to the right of a* if and only if A is not serialisable to the right of a, i.e.

$$\forall B, C \in \widehat{\mathscr{P}}A.\ ((B \cup C = A \ \wedge\ a \in C) \implies B \times C \nsubseteq ser).$$

For a step A, we know that $\varphi_A = (\Sigma_A, \prec_A, \sqsubset_A)^\diamond$ is the stratified order structure induced by the comtrace $[A]$. Then we can relate the non-serialisable step definitions to the relation $\sqsubset_A$ in the following proposition.

Proposition 9.5. *Let A be a step over a comtrace alphabet (E, sim, ser) then*

1. *If A is non-serialisable to the left of $l(\alpha)$ for some $\alpha \in \overline{A}$ then $\forall \beta \in \overline{A}.\ \alpha \sqsubset_A^* \beta$.*
2. *If A is non-serialisable to the right of $l(\beta)$ for some $\beta \in \overline{A}$ then $\forall \alpha \in \overline{A}.\ \alpha \sqsubset_A^* \beta$.*
3. *If A is non-serialisable then $\forall \alpha, \beta \in \overline{A}.\ \alpha \sqsubset_A^* \beta$.*

Proof. 1. For any $\beta \in \overline{A}$, we have to show that $\alpha \sqsubset_A^* \beta$. We define the $\sqsubset_A$-*right closure set* of α inductively as follows:

$$\begin{aligned} RC^0(\alpha) &\overset{df}{=} \{\alpha\} \\ RC^n(\alpha) &\overset{df}{=} \{\delta \in \overline{A} \mid \exists \gamma \in RC^{n-1}(\alpha) \ \wedge \ \gamma \sqsubset_A \delta\} \end{aligned}$$

We want to prove that if $\overline{A} \setminus RC^n(\alpha) \neq \emptyset$ then $|RC^{n+1}(\alpha)| > |RC^n(\alpha)|$. Assume that $\overline{A} \setminus RC^n(\alpha) \neq \emptyset$, and let us consider the set $\overline{A} \setminus RC^n(\alpha)$ and $RC^n(\alpha)$. Since A is non-serialisable to the left of $l(\alpha)$ and $\alpha \in \overline{A}$, we know that

$$l[\overline{A} \setminus RC^n(\alpha)] \times l[RC^n(\alpha)] \nsubseteq ser.$$

Thus there exists some $\gamma \in \overline{A} \setminus RC^n(\alpha)$ such that there is some $\delta \in RC^n(\alpha)$ satisfying $(l(\gamma), l(\delta)) \notin ser$. Hence, by Definition 9.2, we know that $\delta \sqsubset_A \gamma$. Thus, $\gamma \in RC^{n+1}(\alpha)$ where $\gamma \notin RC^n(\alpha)$. So $|RC^{n+1}(\alpha)| > |RC^n(\alpha)|$ as desired.

Since A is finite and if $\overline{A} \setminus RC^n(\alpha) \neq \emptyset$ then $|RC^{n+1}(\alpha)| > |RC^n(\alpha)|$, for some $n < |A|$, we must have $RC^n(\alpha) = \overline{A}$. Thus, $\beta \in RC^n(\alpha)$. By the way the $RC^n(\alpha)$ is defined, it follows that $\alpha \sqsubset_A^* \beta$.

2. The proof is dual to (1) by defining the $\sqsubset_A$-*left closure set* of β inductively as follows:

$$\begin{aligned} LC^0(\beta) &\overset{df}{=} \{\beta\} \\ LC^n(\beta) &\overset{df}{=} \{\delta \in \overline{A} \mid \exists \gamma \in LC^{n-1}(\beta) \ \wedge \ \delta \sqsubset_A \gamma\} \end{aligned}$$

We then prove that if $\overline{A} \setminus LC^n(\beta) \neq \emptyset$ then $|LC^{n+1}(\beta)| > |LC^n(\beta)|$. Thus, for some $n < |A|$, we must have $LC^n(\beta) = \overline{A}$ and hence $\alpha \in LC^n(\beta)$. By the way the

$LC^n(\beta)$ is defined, we conclude that $\alpha \sqsubset^*_A \beta$.

3. Since A is non-serialisable, it follows that A is non-serialisable to the left of $l(\alpha)$ for *every* $\alpha \in \overline{A}$. Hence, for every $\alpha \in \overline{A}$, we have $\forall \beta \in \overline{A}.\ \alpha \sqsubset^*_A \beta$ as desired. □

The existence of a non-serialisable sub-step of a step A to the left/right of an element $a \in A$ can be explained by the following proposition.

Proposition 9.6. *Let A be a step over a comtrace alphabet (E, sim, ser) and $a \in A$. Then*

1. *There exists a unique $B \subseteq A$ such that $a \in B$, B is non-serialisable to the left of a, and*
$$A \neq B \implies A \equiv (A \setminus B)B.$$

2. *There exists a unique $C \subseteq A$ such that $a \in B$, C is non-serialisable to the right of a, and*
$$A \neq C \implies A \equiv C(A \setminus C).$$

Proof. 1. If A is non-serialisable to the left of a, then $B = A$. If A is serialisable to the left of a, then the following set is not empty:

$$\zeta \overset{df}{=} \{D \in \widehat{\mathscr{P}}A \mid \exists C \in \widehat{\mathscr{P}}A.\,(C \cup D = A\ \wedge\ a \in D\ \wedge\ C \times D \subseteq ser)\}$$

Let $B \in \zeta$ such that B is a minimal element of the poset $(\zeta, \subset)$. We claim that B is non-serialisable to the left of a. Suppose for a contradiction that B is serialisable to the left of a, then there are some sets $E, F \in \widehat{\mathscr{P}}B$ such that

$$E \cup F = B\ \wedge\ a \in F\ \wedge\ E \times F \subseteq ser.$$

Since $B \in \chi$, there is some set $G \in \widehat{\mathscr{P}}A$ such that

$$G \cup B = A\ \wedge\ a \in B\ \wedge\ G \times B \subseteq ser.$$

Since $G \times B \subseteq ser$ and $F \subset B$, it follows that $G \times F \subseteq ser$. But since $E \times F \subseteq ser$, we have $(G \cup E) \times F \subseteq ser$. Hence,

$$(G \cup E) \cup F = A\ \wedge\ a \in F\ \wedge\ (G \cup E) \times F \subseteq ser.$$

So $E \in \zeta$ and $E \subset B$. This contradicts that B is minimal. Hence, B is non-serialisable to the left of a.

By the way the set ζ is defined, $A \equiv (A \setminus B)B$. It remains to prove the uniqueness of B. Let $B' \in \zeta$ such that B' is a minimal element of the poset $(\zeta, \subset)$. We want to show that $B = B'$.

We first show that $B \subseteq B'$. Suppose for a contradiction that there is some $b \in B$ such that $b \neq a$ and $b \notin B'$. Let α and β denote the event occurrences $a^{(1)}$ and $b^{(1)}$ in Σ_A respectively. Since $a \in B$ and B is non-serialisable to the left of a, it follows from Proposition 9.5(1) that $\alpha \sqsubset_A^* \beta$. But since $a \neq b$, $\alpha\ (\sqsubset_A^* \setminus id_{\Sigma_A})\ \beta$. From the definition of $\diamond$-closure, it follows that $\alpha \sqsubset_{[A]} \beta$. Hence, by Proposition 9.4(2), we have

$$\forall u \in [A].\ pos_u(\alpha) \leq pos_u(\beta) \tag{9.2}$$

By the way B' is chosen, we know $A \equiv (A \setminus B')B'$ and $b \notin B'$. So it follows that $b \in (A \setminus B')$. Hence, we have $(A \setminus B')B' \in [A]$ and $pos_{(A \setminus B')B'}(\beta) < pos_{(A \setminus B')B'}(\alpha)$, which contradicts (9.2). Thus, $B \subseteq B'$.

By reversing the role of B and B', we can prove that $B \supseteq B'$. Hence $B = B'$.

2. The proof is dual to (1) by considering the set

$$\psi \overset{df}{=} \{C \in \widehat{\mathscr{P}}A \mid \exists D \in \widehat{\mathscr{P}}A.(C \cup D = A\ \wedge\ a \in C\ \wedge\ C \times D \subseteq ser)\}.$$

□

Proposition 9.7. *Let $s = A_1 \ldots A_n$, where $n \geq 2$, be a canonical step sequence over a comtrace alphabet (E, sim, ser) and let $\overline{s} = \overline{A_1} \ldots \overline{A_n}$ be the enumerated step sequence of s. Then for every $\alpha \in \overline{A_n}$ there exist $\alpha_1 \in \overline{A_1}, \ldots, \alpha_{n-1} \in \overline{A_{n-1}}$ such that*

$$\alpha_1(\prec_s \circ \sqsubset_s^*) \ldots (\prec_s \circ \sqsubset_s^*)\alpha_{n-1}(\prec_s \circ \sqsubset_s^*)\alpha.$$

Proof. We proceed by induction on n, the number of steps of s.

When $n = 2$, we have $s = A_1A_2$. Let $C \subseteq A_2$ be non-serialisable to the right of $l(\alpha)$ as constructed in Proposition 9.6(2). Since s is canonical, by Corollary 4.1, $A_1 \times C \not\subseteq ser$. Hence, there is $\alpha_1 \in \overline{A_1}$ and $\alpha_2' \in \overline{A_2}$ such that $l(\alpha_2) \in C$ and

$(l(\alpha_1), l(\alpha_2)) \notin ser$. So it follows from Definition 9.2 that $\alpha_1 \prec_s \alpha_2$. Since C is non-serialisable to the right of $l(\alpha)$, by Proposition 9.5(2), $\alpha_2 \sqsubset_s^* \alpha$. Hence, $\alpha_1 \prec_s \alpha_2 \sqsubset_s^* \alpha$, which implies $\alpha_1 (\prec_s \circ \sqsubset_s^*) \alpha$.

When $n > 2$, we proceed similarly to the case of $n = 2$ to show that there is some $\alpha_{n-1} \in \overline{A_{n-1}}$ satisfying $\alpha_{n-1} (\prec_s \circ \sqsubset_s^*) \alpha$. By applying the induction hypothesis on α_{n-1}, there exist $\alpha_1 \in \overline{A_1}, \ldots, \alpha_{n-1} \in \overline{A_{n-1}}$ such that $\alpha_1(\prec_s \circ \sqsubset_s^*) \ldots (\prec_s \circ \sqsubset_s^*)\alpha_{n-1}$. Hence, $\alpha_1(\prec_s \circ \sqsubset_s^*) \ldots (\prec_s \circ \sqsubset_s^*)\alpha_{n-1}(\prec_s \circ \sqsubset_s^*)\alpha$. □

Proposition 9.8. *Let $t = [s]$ be a comtrace over a comtrace alphabet (E, sim, ser) and let $\varphi_t = (\Sigma_t, \prec_t, \sqsubset_t)$ be the stratified order structure induced by t. Then for any two event occurrences $\alpha, \beta \in \Sigma_t$:*

1. $(\forall u \in t.\ pos_u(\alpha) < pos_u(\beta)) \implies \alpha \prec_t \beta$,

2. $(\alpha \neq \beta\ \wedge\ \forall u \in t.\ pos_u(\alpha) \leq pos_u(\beta)) \implies \alpha \sqsubset_t \beta$.

Proof. 1. Let $w = A_1 \ldots A_n$ be the canonical representation of t, then by Theorem 9.1 we have

$$\varphi_t = (\Sigma_t, \prec_t, \sqsubset_t) = (\Sigma_w, \prec_w, \sqsubset_w)^{\diamond}.$$

We will prove using induction on n (the number of steps of w) that for all $\alpha, \beta \in \Sigma_{[A_1 \ldots A_n]}$

$$(\forall u \in t.\ pos_u(\alpha) < pos_u(\beta)) \implies \alpha \prec_t \beta.$$

When $n = 0$, we have the canonical step is λ and hence the implication is trivially true. When $n > 0$, we observe that $w' = A_1 \ldots A_{n-1}$ is the canonical step sequence of the comtrace $t' = [s \div_R A_n]$. For all $\alpha, \beta \in \Sigma_{t'}$, since $\forall u \in t.\ pos_u(\alpha) < pos_u(\beta)$, it follows that

$$\forall u \in \{vA_n \mid v \equiv A_1 \ldots A_{n-1}\}.\ pos_u(\alpha) < pos_u(\beta).$$

Thus, $\forall u \in t'.\ pos_u(\alpha) < pos_u(\beta)$. By induction hypothesis, we have $\alpha \prec_{t'} \beta$. Hence, from Definition 9.2 and $\diamond$-closure definition, $\alpha(\prec_{w'} \cup \sqsubset_{w'})^* \circ \prec_{w'} \circ (\prec_{w'} \cup \sqsubset_{w'})^* \beta$. But since $w' = w \div_R A_n$, it follows that $\alpha(\prec_w \cup \sqsubset_w)^* \circ \prec_w \circ (\prec_w \cup \sqsubset_w)^* \beta$. Thus, $\alpha \prec_t \beta$. We have just shown that:

$$\forall \alpha, \beta \in \Sigma_{[A_1 \ldots A_{n-1}]}.\ ((\forall u \in t.\ pos_u(\alpha) < pos_u(\beta)) \implies \alpha \prec_t \beta)$$

It remains to show that for all $\alpha \in \Sigma_{[A_1 \ldots A_{n-1}]}$ and $\beta \in (\Sigma_{[A_1 \ldots A_n]} \setminus \Sigma_{[A_1 \ldots A_{n-1}]})$, the following implication holds

$$(\forall u \in t.\ pos_u(\alpha) < pos_u(\beta)) \implies \alpha \prec_t \beta$$

We observe that for any $\alpha \in \Sigma_{[A_1 \ldots A_{n-1}]}$ and $\beta \in (\Sigma_{[A_1 \ldots A_n]} \setminus \Sigma_{[A_1 \ldots A_{n-1}]})$ satisfying

$$\forall u \in t.\ pos_u(\alpha) < pos_u(\beta),$$

by Proposition 9.6, there must be some $v \in t$ of the form $\overline{v} = \ldots \overline{B}\, \overline{C_1} \ldots \overline{C_k}\, \overline{D} \ldots$ where:

- $\alpha \in \overline{B}$ and B is non-serialisable to the left of $l(\alpha)$,
- $\beta \in \overline{D}$ and D is non-serialisable to the right of $l(\beta)$.

Let V be a set containing all such $\overline{v}$. Recall that for a step sequence $x = E_1 \ldots E_r$, we define

$$\mu(x) \overset{df}{=} 1 \cdot |E_1| + \ldots + r \cdot |E_r|.$$

We let $\overline{v_0} = \overline{x}\, \overline{B^0}\, \overline{C^0_1} \ldots \overline{C^0_{k_0}}\, \overline{D^0}\, \overline{y}$ in V such that $\mu(\overline{C^0_1} \ldots \overline{C^0_{k_0}})$ is the least among all $v_i \in V$, i.e.

$$\forall v_i \in V.\ \left(v_i = \ldots \overline{B^i}\, \overline{C^i_1} \ldots \overline{C^i_{k_i}}\, \overline{D^i} \ldots \implies \mu(\overline{C^0_1} \ldots \overline{C^0_{k_0}}) \leq \mu(\overline{C^i_1} \ldots \overline{C^i_{k_i}})\right).$$

Then there are two cases to consider:

Case (i):

If $\mu(\overline{C^0_1} \ldots \overline{C^0_{k_0}}) = 0$, then we have $\overline{v_0} = \overline{x}\, \overline{B^0}\, \overline{D^0}\, \overline{y}$. Since $\forall u \in t.\ pos_u(\alpha) < pos_u(\beta)$, we know $B^0 \times D^0 \not\subseteq ser$. Hence, there is some $\alpha_1 \in \overline{B^0}$ and $\beta_1 \in \overline{D^0}$ such that $(l(\alpha_1), l(\beta_1)) \notin ser$. But since $pos_{v_0}(\alpha_1) < pos_{v_0}(\beta_1)$, it follows that

$$\alpha_1 \prec_{v_0} \beta_1 \tag{9.3}$$

Since B^0 is non-serialisable to the left of $l(\alpha)$ and D^0 is non-serialisable to the right of $l(\beta)$, it follows from Proposition 9.5(1, 2) that

$$\alpha \sqsubset^*_{v_0} \alpha_1 \text{ and } \beta_1 \sqsubset^*_{v_0} \beta \tag{9.4}$$

From (9.3) and (9.4), we conclude that

$$\alpha \sqsubset_{v_0}^* \alpha_1 \prec_{v_0} \beta_1 \sqsubset_{v_0}^* \beta.$$

Hence,

$$\alpha \sqsubset_{v_0}^* \circ \prec_{v_0} \circ \sqsubset_{v_0}^* \beta \tag{9.5}$$

By Theorem 9.1, $\varphi_t = (\Sigma_t, \prec_t, \sqsubset_t) = (\Sigma_{v_0}, \prec_{v_0}, \sqsubset_{v_0})^{\diamond}$. Thus, it follows from Definition 9.2 and (9.5) that $\alpha \prec_t \beta$.

Case (ii):

If $\mu(\overline{C_1^0} \ldots \overline{C_{k_0}^0}) \neq 0$, then $\overline{v_0} = \overline{x}\ \overline{B^0}\ \overline{C_1^0} \ldots \overline{C_{k_0}^0}\ \overline{D^0}\ \overline{y}$ where $k_0 > 0$. We know that $C_{k_0}^0 \times D^0 \not\subseteq ser$, otherwise $\mu(\overline{C_1^0} \ldots \overline{C_{k_0}^0})$ is not the least. Hence, there is some $\gamma_{k_0} \in \overline{C_{k_0}^0}$ and $\beta_1 \in \overline{D^0}$ such that $(l(\gamma), l(\beta_1)) \notin ser$. Since $pos_{v_0}(\gamma) < pos_{v_0}(\beta)$, from Definition 9.2, it follows that

$$\gamma_{k_0} \prec_{v_0} \beta_1 \tag{9.6}$$

Since $\mu(\overline{C_1^0} \ldots \overline{C_{k_0}^0})$ is the least, by Corollary 4.1, $C_1^0 \ldots C_{k_0}^0$ is canonical. Hence, by Proposition 9.7, there exist a sequence $\gamma_1 \in \overline{C_1}, \ldots, \gamma_{k_0} \in \overline{C_{k_0}}$ $(k_0 \geq 1)$ such that

$$\gamma_1 (\prec_{v_0} \circ \sqsubset_{v_0}^*) \ldots (\prec_{v_0} \circ \sqsubset_{v_0}^*) \gamma_{k_0} \tag{9.7}$$

Let $C_1' \subseteq C_1$ be non-serialisable to the right of $l(\gamma_1)$ as given in Proposition 9.6(2). Clearly, since $\mu(\overline{C_1^0} \ldots \overline{C_{k_0}^0})$ is the least, $B^0 \times C_1' \not\subseteq ser$. Similarly to case (i), we can show that

$$\alpha \prec_t \gamma_1 \tag{9.8}$$

Since D^0 is non-serialisable to the right of $l(\beta)$, by Proposition 9.5(2), $\beta_1 \sqsubset_{v_0}^* \beta$. So it follows from (9.6) that $\gamma_{k_0} \prec_{v_0} \beta_1 \sqsubset_{v_0}^* \beta$. Thus, together with (9.7), we get

$$\gamma_1 (\prec_{v_0} \circ \sqsubset_{v_0}^*) \ldots (\prec_{v_0} \circ \sqsubset_{v_0}^*) \gamma_{k_0} \prec_{v_0} \beta_1 \sqsubset_{v_0}^* \beta$$

Hence, it follows from Definition 9.2 that

$$\gamma_1 \prec_t \beta \tag{9.9}$$

From (9.8) and (9.9), it follows that $\alpha \prec_t \gamma_1 \prec_t \beta$. Hence, $\alpha \prec_t \beta$ by transitivity of $\prec_t$.

2. For any $\alpha, \beta \in \Sigma_t$, if $\alpha \neq \beta$ and $\forall u \in t.\ pos_u(\alpha) < pos_u(\beta)$, then by (1) we have $\alpha \prec_t \beta$. Thus, $\alpha \sqsubset_t \beta$. Otherwise, there are some $u \in t$ such that $pos_u(\alpha) = pos_u(\beta)$. Hence, there is some step sequence u such that $\overline{u} = \overline{r}\ \overline{B}\ \overline{s}$ and $\alpha, \beta \in \overline{B}$. If B is non-serialisable to the left of $l(\alpha)$, by Proposition 9.5(1),

$$\alpha \sqsubset_v^* \beta \tag{9.10}$$

Otherwise, by Proposition 9.6(1), there are some steps $C, D \subset B$ such that $B \equiv CD$, $l(\alpha) \in D$, and D is non-serialisable to the left of $l(\alpha)$. Hence, there is some step sequence $v \in t$ such that $\overline{v} = \overline{r}\ \overline{C}\ \overline{D}\ \overline{s}$. Since $\forall u \in t.\ pos_u(\alpha) \leq pos_u(\beta))$ and $\alpha \in \overline{D}$, it follows that $\beta \in \overline{D}$. Since D is non-serialisable to the left of $l(\alpha)$, by Proposition 9.5(1),

$$\alpha \sqsubset_v^* \beta \tag{9.11}$$

Since $\alpha \neq \beta$, from (9.10) and (9.11), we have $\alpha\ (\sqsubset_v^* \setminus id_{\Sigma_v})\ \beta$. By $\diamondsuit$-closure definition, we conclude that $\alpha \sqsubset_t \beta$ as desired. □

Proposition 9.9. *Let $t = [s]$ be a comtrace over a comtrace alphabet (E, sim, ser) and let $\varphi_t = (\Sigma_t, \prec_t, \sqsubset_t)$ be the stratified order structure induced by t. Then for any two event occurrences $\alpha, \beta \in \Sigma_t$:*

1. $(\forall u \in t.\ pos_u(\alpha) < pos_u(\beta)) \iff \alpha \prec_t \beta$,
2. $(\alpha \neq \beta\ \wedge\ \forall u \in t.\ pos_u(\alpha) \leq pos_u(\beta)) \iff \alpha \sqsubset_t \beta$.

Proof. Follows directly from Propositions 9.4 and 9.8. □

According to the Szpilrajn Theorem, every poset can be reconstructed by taking the intersection of its total order extensions. A similar result holds for stratified order structures and stratified order extensions.

Theorem 9.2 ([15, Theorem 2.9]). *Let $S = (X, \prec, \sqsubset)$ be a stratified order structure. Then*

$$S = \left(X,\ \bigcap_{\lhd \in ext(S)} \lhd,\ \bigcap_{\lhd \in ext(S)} \lhd^{\frown} \right).$$

□

In the context of comtraces, the following theorem from [14] says that the stratified order extensions of φ_t are exactly those generated by the step sequences in $[t]$.

Theorem 9.3 ([14, Theorem 4.12]). *Let $t = [s]$ be a comtrace over a comtrace alphabet (E, sim, ser). Then $ext(\varphi_t) = \{\lhd_u \mid u \in t\}$.* □

Corollary 9.1. *Let t be a comtrace over a comtrace alphabet (E, sim, ser). Then*

$$\varphi_t = \left(\Sigma_t, \bigcap_{u \in t} \lhd_u, \bigcap_{u \in t} \lhd_u^{\frown}\right).$$

Proof. By Theorem 9.3, $ext(\varphi_t) = \{\lhd_u \mid u \in t\}$. Hence, by Theorem 9.2, we have

$$\varphi_t = \left(\Sigma_t, \bigcap_{\lhd \in ext(\varphi_t)} \lhd, \bigcap_{\lhd \in ext(\varphi_t)} \lhd^{\frown}\right) = \left(\Sigma_t, \bigcap_{u \in t} \lhd_u, \bigcap_{u \in t} \lhd_u^{\frown}\right).$$

□

Although Corollary 9.1 is equivalent to Proposition 9.9, we provided the alternative proofs of Propositions 9.4 and 9.8 without using Theorems 9.2 and 9.3. Firstly, it shows that Propositions 9.4 and 9.8 can be proved based on the construction from Definition 9.2 without using the sophisticated generalisation of the Szpilrajn Theorem for stratified order structures. Secondly, the proofs of Propositions 9.4 and 9.8 provide more intuition why any two event occurrences in a comtrace t cannot violate the invariants imposed by the generated stratified order structure φ_t. Moreover, we invented three different notions of non-serialisable steps, which are the key to explain how the causality and weak causality relations can be derived from *the relationships among the steps*[1] (sets of event occurrences) on a step sequence.

Even though Corollary 9.1 makes it simpler to construct a stratified order structure from a comtrace, the construction from Definition 9.2 has its own advantages. From a single step sequence s and a comtrace concurrent alphabet, the $\diamond$-closure construction can be used to construct the stratified order structure $\varphi_{[s]}$ without the need to construct all step sequences in $[s]$ and their generated stratified orders. Also

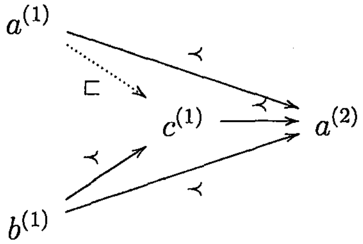
[1] This is different from the construction using $\diamond$-closure, which derives a stratified order structure by looking at *the relationship of every pair of event-occurrences* on a step sequence.

the $\diamond$-closure construction builds the relations $\prec_{[s]}$ and $\sqsubset_{[s]}$ from the relations $\prec_s$ and $\sqsubset_s$, which are often much simpler and easier to handle. The proof of Theorem 9.5 is one such example.

9.3 Comtrace Representation of Finite Stratified Order Structures

Although was shown in [14] that each finite stratified order structure can be represented by a comtrace, the converse which says each finite stratified order structure can be represented by a comtrace was not shown. The intuition of how to construct a finite stratified order structure from a comtrace can be shown in the following example, which is the converse of Example 9.2.

Example 9.3. Starting from the stratified order structure $S = (\Sigma, \prec, \sqsubset)$:



We can check that

$$\begin{aligned}\Delta &= \{u \mid \lhd_u \in ext(S)\}\\ &= \{\{a,b\}\{c\}\{a\}, \{a\}\{b\}\{c\}\{a\}, \{b\}\{a\}\{c\}\{a\}, \{b\}\{a,c\}\{a\}\}\end{aligned}$$

From Δ, we can build a comtrace alphabet $\theta = (E, sim, ser)$ where

- $E = l(\Sigma) = \{a, b, c\}$
- We define the relation sim such that

$$(a,b) \in sim \iff \exists \lhd \in ext(S).\ (l(\alpha) = a \wedge l(\beta) = b \wedge \alpha \frown_{\lhd} \beta)$$

Hence, $sim = \{(a,b), (b,a), (a,c), (c,a)\}$

- We define the relation *ser* such that

$$(a,b) \in ser \iff (a,b) \in sim \wedge \exists \lhd \in ext(S).\ (l(\alpha) = a \wedge l(\beta) = b \wedge \alpha \lhd \beta)$$

  Thus, $ser = \{(a,b),(b,a),(a,c)\}$

Clearly, Δ is a comtrace over θ. □

Before proving the main theorem of this chapter, we need several results from [15, 14] and their corollaries. The first result comes from the fact that stratified order structures conform to paradigm π_3.

Theorem 9.4 ([15, Theorem 3.6]). *Let $S = (X, \prec, \sqsubset)$ be a stratified order structure. Then*

$$((\exists \lhd \in ext(S).\ \alpha \lhd \beta) \wedge (\exists \lhd \in ext(S).\ \beta \lhd \alpha)) \implies (\exists \lhd \in ext(S).\ \beta \frown_{\lhd} \alpha).$$

□

Corollary 9.2. *Let $S = (X, \prec, \sqsubset)$ be a stratified order structure. Then*

$$(\forall \lhd \in ext(S).\ \alpha \lhd \beta \vee \beta \lhd \alpha) \implies ((\forall \lhd \in ext(S).\ \alpha \lhd \beta) \vee (\forall \lhd \in ext(S).\ \beta \lhd \alpha)).$$

Proof. Assume

$$\forall \lhd \in ext(S).\ \alpha \lhd \beta \vee \beta \lhd \alpha \tag{9.12}$$

and suppose for a contradiction that

$$\neg(\forall \lhd \in ext(S).\ \alpha \lhd \beta) \wedge \neg(\forall \lhd \in ext(S).\ \beta \lhd \alpha).$$

Hence, it follows that

$$(\exists \lhd \in ext(S).\ \alpha \lhd^{\frown} \beta) \wedge (\exists \lhd \in ext(S).\ \beta \lhd^{\frown} \alpha) \tag{9.13}$$

If $\exists \lhd \in ext(S).\ \beta \frown_{\lhd} \alpha$, then we get a contradiction with the assumption (9.12). Otherwise, suppose that $\neg(\exists \lhd \in ext(S).\ \beta \frown_{\lhd} \alpha)$. Then it follows from (9.13) that

$$(\exists \lhd \in ext(S).\ \alpha \lhd \beta) \wedge (\exists \lhd \in ext(S).\ \beta \lhd \alpha).$$

But this implies $\exists \lhd \in ext(S).\ \beta \frown_{\lhd} \alpha$ by Theorem 9.4, which again contradicts the assumption (9.12). □

Proposition 9.10 (Propositions 3.4 and 3.5 of [14]). *If $S = (X, \prec, \sqsubset)$ is a stratified order structure, and $S_0 = (X, \prec_0, \sqsubset_0)$ is a relational structure such that $S_0 \subseteq S$, then $S_0^{\diamond}$ is a stratified order structure satisfying $S_0^{\diamond} \subseteq S$.* □

Before proving the next lemma, we need a standard set-theoretic result.

Proposition 9.11. *If $X = \bigcap A$ and $Y = \bigcap B$ and $A \subseteq B$, then $Y \subseteq X$.*

Proof. Suppose that $x \in Y = \bigcap B$. Hence, $\forall C \in B.\ x \in C$. But since $A \subseteq B$, it follows that for all $\forall C \in A.\ x \in C$. Thus, $x \in X = \bigcap A$. Hence, $Y \subseteq X$. □

Lemma 9.1. *Let $S_0 = (X, \prec_0, \sqsubset_0)$ and $S_1 = (X, \prec_1, \sqsubset_1)$ be stratified order structures such that $ext(S_0) \subseteq ext(S_1)$. Then $S_1 \subseteq S_0$.*

Proof. By Theorem 9.2, we know $\prec_0 = \bigcap_{\lhd \in ext(S_0)} \lhd$ and $\prec_1 = \bigcap_{\lhd \in ext(S_1)} \lhd$. But since $ext(S_0) \subseteq ext(S_1)$, it follows from Proposition 9.11 that

$$\prec_1 \subseteq \prec_0 \tag{9.14}$$

By Theorem 9.2, we know $\sqsubset_0 = \bigcap_{\lhd \in ext(S_0)} \lhd^{\frown}$ and $\sqsubset_1 = \bigcap_{\lhd \in ext(S_1)} \lhd^{\frown}$. Since $ext(S_0) \subseteq ext(S_1)$, we have

$$\{\lhd^{\frown} \mid \lhd \in ext(S_0)\} \subseteq \{\lhd^{\frown} \mid \lhd \in ext(S_1)\}.$$

Thus, it follows from Proposition 9.11 that

$$\sqsubset_1 \subseteq \sqsubset_0 \tag{9.15}$$

From (9.14) and (9.15), we conclude $S_1 \subseteq S_0$. □

We will now show that we can build a comtrace from a finite stratified order structure using the construction from Example 9.3, where *sim* and *ser* are binary relations defined on the *labels* of the event occurrences. Although this method allows us to represent a labelled finite stratified order structure using a comtrace defined over a more concise comtrace alphabet, it does not work for *every* finite stratified order structure. For example, in the following stratified order structure

$$a^{(1)} \overset{\sqsubset}{\dashrightarrow} b^{(1)} \xrightarrow{\prec} a^{(2)} \xrightarrow{\prec} b^{(2)}$$

we cannot define $(a, b) \in ser$ since $a^{(2)} \prec b^{(2)}$. Also since *sim* is irreflexive, in the following stratified order structure, we cannot say that $(a, a) \in sim$.

$$\begin{tikzcd} a^{(1)} \arrow[r, dotted, "\sqsubset"] & a^{(2)} \end{tikzcd}$$

However, the construction works for a special kind of finite stratified order structures which we define next.

Definition 9.3. A finite stratified order structure $S = (\Sigma, \prec, \sqsubset)$ is a *proper stratified order structure* if it satisfies the following three conditions:

1. Σ is the set of event occurrences.

2. If $\alpha, \beta \in \Sigma$, $\alpha \neq \beta$, and $l(\alpha) = l(\beta)$, then $(l(\alpha), l(\beta)) \in \prec \cup \prec^{-1}$.

3. Let $\lhd_3, \lhd_4$ be stratified orders on Σ where $\Omega_{\lhd_3} = X_1 \ldots X_m(X \cup Y)Y_1 \ldots Y_n$ and $\Omega_{\lhd_3} = X_1 \ldots X_m XYY_1 \ldots Y_n$ and

$$\forall \alpha \in X.\ \forall \beta \in Y.\ \exists \lhd_1, \lhd_2 \in ext(S).\ \exists \alpha', \beta' \in \Sigma. \left(\begin{array}{ll} & l(\alpha) = l(\alpha') \\ \wedge & l(\beta) = l(\beta') \\ \wedge & \alpha' \lhd_1 \beta' \\ \wedge & \alpha' \frown_{\lhd_2} \beta' \end{array} \right).$$

   Then $\lhd_3 \in ext(S)$ if and only if $\lhd_4 \in ext(S)$.

Theorem 9.5. *Let $S = (\Sigma, \prec, \sqsubset)$ be a proper stratified order structure, $\Delta = \{u \mid \lhd_u \in ext(S)\}$, and $E = l(\Sigma)$. Let relations $sim, ser \subseteq E \times E$ be defined as follows:*

$$(l(\alpha), l(\beta)) \in sim \iff \exists \lhd \in ext(S).\ \alpha \frown_{\lhd} \beta \tag{9.16}$$

$$(l(\alpha), l(\beta)) \in ser \iff (l(\alpha), l(\beta)) \in sim \ \wedge\ \exists \lhd \in ext(S).\ \alpha \lhd \beta \tag{9.17}$$

Then we have:

1. *$\theta = (E, sim, ser)$ is a comtrace alphabet,*

2. *Δ is a comtrace over θ.*

Proof. 1. For any two labels $a, b \in l(\Sigma)$ we have $(a, b) \in sim$. Because S is a proper stratified order structure, by Condition (2) of Definition 9.3 we know that for all $\alpha, \beta \in \Sigma$,

$$l(\alpha) = l(\beta) \implies (l(\alpha), l(\beta)) \in \prec \cup \prec^{-1}.$$

This mean for all $\alpha, \beta \in \Sigma$,

$$l(\alpha) = l(\beta) \implies \forall \lhd \in ext(S).\ \neg(\alpha \frown_{\lhd} \beta).$$

But since $\frown_{\lhd}$ is irreflexive and symmetric, it follows that the relation sim is irreflexive and symmetric.

From (9.17), $(a, b) \in ser$ implies that $(a, b) \in sim$. So $ser \subseteq sim$.

It remains to show that for any pair of distinct element α, β satisfying $pos_u(\alpha) = pos_u(\beta)$ for some $u \in \Delta$ (α and β are in the same step of u), we have $(l(\alpha), l(\beta)) \in sim$. But $pos_u(\alpha) = pos_u(\beta)$ implies $\alpha \frown_{\lhd} \beta$ for some $\lhd \in ext(S)$. Hence, from (9.16), $(l(\alpha), l(\beta)) \in sim$.

Hence, (E, sim, ser) is a comtrace alphabet as desired.

2. We first need to check that all $u \in \Delta$ are step sequences *over* the alphabet θ. Let $u = A_1 \dots A_n \in \Delta$ and $\overline{u} = \overline{A_1} \dots \overline{A_n}$ be the enumerated step sequence of u. We want to show for any $\alpha, \beta \in A_i$ for any i, $(l(\alpha), l(\beta)) \in sim$. But since

$$\alpha, \beta \in A_i \implies \alpha \frown_{\lhd_u} \beta$$

and $\lhd_u \in ext(S)$, it follows from (9.16) that $(l(\alpha), l(\beta)) \in sim$.

Next we let u be a step sequence in Δ and $S_u = (\Sigma, \prec_u, \sqsubset_u)$ as from Definition 9.2. We want to show that that $\varphi_u = S_u^{\diamond} \subseteq S$. By Proposition 9.10, it suffices to show that $S_u \subseteq S$.

Assume $\alpha \prec_u \beta$, then from Definition 9.2, $\alpha \lhd_u \beta \wedge (l(\alpha), l(\beta)) \notin ser$. From (9.16) and (9.17), it follows that

$$\alpha \lhd_u \beta \wedge (\neg(\exists \lhd \in ext(S).\ \alpha \lhd \beta) \vee \neg(\exists \lhd \in ext(S).\ \alpha \frown_{\lhd} \beta)).$$

Since $\neg(\exists \lhd \in ext(S).\ \alpha \lhd \beta)$ contradicts that $\alpha \lhd_u \beta$, we have

$$\alpha \lhd_u \beta \wedge \neg(\exists \lhd \in ext(S).\ \alpha \frown_{\lhd} \beta).$$

Hence,

$$\alpha \lhd_u \beta \wedge (\forall \lhd \in ext(S).\ \alpha \lhd \beta \vee \beta \lhd \alpha).$$

Then, by Corollary 9.2,

$$\alpha \lhd_u \beta \wedge ((\forall \lhd \in ext(S).\ \alpha \lhd \beta) \vee (\forall \lhd \in ext(S).\ \beta \lhd \alpha)).$$

Since $\alpha \lhd_u \beta$ contradicts that $\forall \lhd \in ext(S).\ \beta \lhd \alpha$, it follows that

$$\forall \lhd \in ext(S).\ \alpha \lhd \beta \tag{9.18}$$

By Theorem 9.2, $\prec = \bigcap_{\lhd \in ext(S)} \lhd$. Hence, (9.18) implies $\alpha \prec \beta$.

Assume $\alpha \sqsubset_u \beta$, then by Definition 9.2, $\alpha \lhd_u^{\frown} \beta \wedge (l(\beta), l(\alpha)) \notin ser$. From (9.16) and (9.17), it follows that

$$\alpha \lhd_u^{\frown} \beta \wedge (\neg(\exists \lhd \in ext(S).\ \beta \lhd \alpha) \vee \neg(\exists \lhd \in ext(S).\ \beta \frown_{\lhd} \alpha)).$$

Hence,

$$\alpha \lhd_u^{\frown} \beta \wedge ((\forall \lhd \in ext(S).\ \alpha \lhd_u^{\frown} \beta) \vee (\forall \lhd \in ext(S).\ \alpha \lhd \beta \vee \beta \lhd \alpha)).$$

If $\forall \lhd \in ext(S).\ \alpha \lhd \beta \vee \beta \lhd \alpha$, then it must follow that $\alpha \lhd_u \beta$. This is the same to the case of $\alpha \prec_u \beta$. Hence, $\alpha \prec \beta$, which implies $\alpha \sqsubset \beta$. Otherwise, we have

$$\alpha \lhd_u^{\frown} \beta \wedge (\forall \lhd \in ext(S).\ \alpha \lhd_u^{\frown} \beta).$$

Thus,

$$\forall \lhd \in ext(S).\ \alpha \lhd_u^{\frown} \beta \tag{9.19}$$

By Theorem 9.2, $\sqsubset = \bigcap_{\lhd \in ext(S)} \lhd^{\frown}$. Hence, (9.19) implies $\alpha \sqsubset \beta$.

Thus, we have shown

$$\varphi_u \subseteq S \tag{9.20}$$

Our next goal is to prove $S \subseteq \varphi_u$. By Lemma 9.1, it suffices to show that $ext(\varphi_u) \subseteq ext(S)$.

We observe that from a step sequence $u \in \Delta$, by Definition 3.5, we can build the comtrace $[u]$ over the alphabet θ using the following inductive derivation sets:

$$\begin{aligned} D^0(u) &\overset{df}{=} \{u\} \\ D^n(u) &\overset{df}{=} \{w \mid w \in D^{n-1}(u) \ \vee\ \exists v \in D^{n-1}(u).\ (\ v \approx w \ \vee\ v \approx^{-1} w)\} \end{aligned}$$

Since u has finite event occurrences, $[u]$ is finite. Hence, $[u] = D^n(u)$ for some $n \geq 0$. We will prove by induction on n that if $w \in D^n(u)$ then $\lhd_w \in ext(S)$. When $n = 0$, $D^0(u) = \{u\}$. Since $u \in \Delta$, $\lhd_u \in ext(S)$. When $n > 0$, let w be an element of $D^n(u)$. Then either $w \in D^{n-1}(u)$ or $w \in (D^n(u) \setminus D^{n-1}(u))$. For the former case, by induction hypothesis, $\lhd_w \in ext(S)$. For the later case, there must be some element $v \in D^{n-1}(u)$ such that $v \approx w$ or $v \approx^{-1} w$. By induction hypothesis, $\lhd_v \in ext(S)$. We want to show that $\lhd_w \in ext(S)$.

Case (i): When $v \approx w$, by Definition 3.5, $v = yAz$ and $w = yBCz$ where A, B, C are steps satisfying $B \cap C = \emptyset$ and $B \cup C = A$ and $B \times C \subseteq ser$. Let $\overline{v} = \overline{y}\overline{A}\overline{z}$ and $\overline{w} = \overline{y}\overline{B}\,\overline{C}\overline{z}$ be enumerated step sequences of v and w respectively. Since $B \times C \subseteq ser$, it follows from (9.17) that

$$\forall \alpha \in \overline{B}.\ \forall \beta \in \overline{C}.\ \exists \lhd_1, \lhd_2 \in ext(S).\ \exists \alpha', \beta' \in \Sigma. \left(\begin{array}{ll} & l(\alpha) = l(\alpha') \\ \wedge & l(\beta) = l(\beta') \\ \wedge & \alpha' \lhd_1 \beta' \\ \wedge & \alpha' \frown_{\lhd_2} \beta' \end{array} \right).$$

Hence, by Condition (3) of Definition 9.3 and $\lhd_v \in ext(S)$, $\lhd_w \in ext(S)$.

Case (ii): When $v \approx^{-1} w$, by Definition 3.5, $w = yAz$ and $v = yBCz$ where A, B, C are steps satisfying $B \cap C = \emptyset$ and $B \cup C = A$ and $B \times C \subseteq ser$. Let $\overline{w} = \overline{y}\overline{A}\overline{z}$ and $\overline{v} = \overline{y}\overline{B}\ \overline{C}\overline{z}$ be enumerated step sequences of w and v respectively. Again similarly to the previous case, since $B \times C \subseteq ser$ and $\lhd_v \in ext(S)$, it follows from Condition (3) of Definition 9.3 that $\lhd_w \in ext(S)$.

Hence, we have shown that for all $n \geq 0$, if $w \in D^n(u)$ then $\lhd_w \in ext(S)$. Thus, $\{\lhd_w \mid w \in D^n(u)\} \subseteq ext(S)$ for every $n \geq 0$. But since Theorem 9.3 implies that

$$ext(\varphi_u) = \{\lhd_w \mid w \in [u]\} = \{\lhd_w \mid w \in D^n(u)\}$$

for some $n \geq 0$, we conclude $ext(\varphi_u) \subseteq ext(S)$. Thus, by Lemma 9.1, we have also shown

$$S \subseteq \varphi_u \tag{9.21}$$

From (9.20) and (9.21), we conclude that $\varphi_u = S$ for any $u \in \Delta$. Thus, for any $u \in \Delta$, it follows from Theorem 9.3 that

$$ext(S) = ext(\varphi_u) = \{\lhd_w \mid w \in [u]\},$$

which means $[u] = \{w \mid \lhd_w \in ext(S)\}$. So we conclude $\Delta = \{w \mid \lhd_w \in ext(S)\} = [u]$ is a comtrace over θ as desired. □

Although Theorem 9.5 only shows how proper stratified order structures can be represented using comtraces, any stratified order structure $(\Sigma, \sqsubset, \prec)$ can be represented by a comtrace by *redefining* the labelling function as

$$l \overset{df}{=} id_\Sigma.$$

In other words, we treat two occurrences of the same event as if they are two distinct events. The construction of Theorem 9.5 works because of the following proposition.

Proposition 9.12. *Let $S = (\Sigma, \sqsubset, \prec)$ be a finite stratified order structure and $l \overset{df}{=} id_\Sigma$. Then S is a proper stratified order structure.*

Proof. Since we redefine $l = id_\Sigma$, the Conditions (1) and (2) of Definition 9.3 are trivially satisfied since no "event" occurs more than once. To verify Condition (3), let $\lhd_3$ and $\lhd_4$ be stratified orders on Σ where $\Omega_{\lhd_3} = X_1 \ldots X_m(X \cup Y)Y_1 \ldots Y_n$ and $\Omega_{\lhd_3} = X_1 \ldots X_m XYY_1 \ldots Y_n$ and

$$\forall \alpha \in X.\ \forall \beta \in Y.\ \exists \lhd_1, \lhd_2 \in ext(S).\ \exists \alpha', \beta' \in \Sigma. \begin{pmatrix} & l(\alpha) = l(\alpha') \\ \wedge & l(\beta) = l(\beta') \\ \wedge & \alpha' \lhd_1 \beta' \\ \wedge & \alpha' \frown_{\lhd_2} \beta' \end{pmatrix}.$$

But since $l = id_\Sigma$, it follows that

$$\forall \alpha \in X.\ \forall \beta \in Y.\ \exists \lhd_1, \lhd_2 \in ext(S).\ (\alpha \lhd_1 \beta\ \wedge\ \alpha \frown_{\lhd_2} \beta) \tag{9.22}$$

We want to show that $\lhd_3 \in ext(S)$ if and only if $\lhd_4 \in ext(S)$.

($\Rightarrow$) Suppose for a contradiction that $\lhd_3 \in ext(S)$ and $\lhd_4 \notin ext(S)$. Hence, by Definition 9.1, there are some $\alpha, \beta \in \Sigma$ such that one of the following holds

$$\alpha \prec \beta \wedge \neg(\alpha \lhd_4 \beta) \tag{9.23}$$

$$\alpha \sqsubset \beta \wedge \neg(\alpha \lhd_4^{\frown} \beta) \tag{9.24}$$

Since $\lhd_4 = \lhd_3 \cup X \times Y$ and $\lhd_3 \in ext(S)$, (9.23) cannot be satisfied. Hence, (9.24) must hold. Since $\neg(\alpha \lhd_4^{\frown} \beta)$, we know $\beta \lhd_4 \alpha$. Because $\lhd_4 = \lhd_3 \cup X \times Y$, we must have $\beta \in X$ and $\alpha \in Y$. By (9.22), it follows that

$$\exists \lhd \in ext(S).\ \beta \lhd \alpha$$

Thus, $\exists \lhd \in ext(S).\ \neg(\alpha \lhd^{\frown} \beta)$. But by Theorem 9.2, $\sqsubset = \bigcap_{\lhd \in ext(S)} \lhd^{\frown}$. Hence, it follows that $\neg(\alpha \sqsubset \beta)$, , which contradicts (9.24).

($\Leftarrow$) Suppose for a contradiction that $\lhd_4 \in ext(S)$ and $\lhd_3 \notin ext(S)$. Hence, by Definition 9.1, there are some $\alpha, \beta \in \Sigma$ such that one of the following holds

$$\alpha \prec \beta \wedge \neg(\alpha \lhd_3 \beta) \tag{9.25}$$

$$\alpha \sqsubset \beta \wedge \neg(\alpha \lhd_3^{\frown} \beta) \tag{9.26}$$

Since $\lhd_3 = \lhd_4 \setminus X \times Y$, we know that if $\alpha \lhd_4^{\frown} \beta$ then $\alpha \lhd_3^{\frown} \beta$. But since $\lhd_4 \in ext(S)$, (9.26) cannot be satisfied. Hence, (9.25) must hold. Because $\lhd_3 = \lhd_4 \setminus X \times Y$, we must have $\alpha, \beta \in X \cup Y$. By (9.22), it follows that

$$\exists \lhd \in ext(S).\ \beta \frown_{\lhd} \alpha$$

But by Theorem 9.2, $\prec = \bigcap_{\lhd \in ext(S)} \lhd$. Hence, $\neg(\alpha \prec \beta)$, which contradicts (9.25). □

Chapter 10

Relational Representation of Generalised Comtraces

In this chapter, we analyse the relationship between generalised comtraces and generalised stratified order structures with the main result showing that each generalised comtrace uniquely defines a finite generalised stratified order structure.

10.1 Properties of Generalised Comtrace Congruence

In this section, we prove some basic properties of generalised comtrace congruence.

Proposition 10.1. *Let $\mathbb{S}$ be the set of all steps over a generalised comtrace alphabet (E, sim, ser, inl) and $u, v \in \mathbb{S}^*$. Then*

1. $u \equiv v \implies weight(u) = weight(v)$. (step sequence weight equality)
2. $u \equiv v \implies |u|_a = |v|_a$. (event-preserving)
3. $u \equiv v \implies u \div_R a \equiv v \div_R a$. (right cancellation)
4. $u \equiv v \implies u \div_L a \equiv v \div_L a$. (left cancellation)
5. $u \equiv v \iff \forall s, t \in \mathbb{S}^*.\ sut \equiv svt$. (step subsequence cancellation)

6. $u \equiv v \implies \pi_D(u) \equiv \pi_D(v)$. (projection rule)

Proof. For all except (5), it suffices to show that $u \approx v$ implies that the right hand side of (1)–(6) holds. Notice that when $u \approx v$, the case $u = xAy \approx v = xBCy$ follows from Proposition 5.1. So we only need to consider the case $u = xABy$ and $v = xBAy$, where $A \times B \subseteq inl$ and $A \cap B = \emptyset$.

1. We have:

$$\begin{aligned} weight(u) &= weight(x) + weight(A) + weight(B) + weight(z) \\ &= weight(x) + weight(B) + weight(A) + weight(z) = weight(v). \end{aligned}$$

2. $|u|_a = |x|_a + |A|_a + |B|_a + |z|_a = |x|_a + |B|_a + |A|_a + |z|_a = |v|_a$.

3. We want to show that $u \div_R a \approx v \div_R a$. There are four cases:

- $a \in \biguplus(y)$: Let $z = y \div_R a$. Then $u \div_R a = xABz \approx xBAz = v \div_R a$.
- $a \notin \biguplus(y)$, $a \in B$: Then $u \div_R a = xA(B \setminus \{a\})y \approx x(B \setminus \{a\})Ay = v \div_R a$.
- $a \notin \biguplus(By)$, $a \in A$: Then $u \div_R a = x(A \setminus \{a\})By \approx xB(A \setminus \{a\})Cy = v \div_R a$.
- $a \notin \biguplus(ABy)$: Let $z = x \div_R a$. Then $u \div_R a = zABy \approx zBAy = v \div_R a$.

4. Dually to (3).

5. ($\Rightarrow$) We want to show that $u \approx v \implies \forall s, t \in \mathbb{S}^*.\ sut \approx svt$. For any two step sequences $s, t \in \mathbb{S}^*$, we have $sut = sxAByt$ and $svt = sxBAyt$. But this clearly implies $sut \approx svt$ by how $\approx$ is defined in Definition 3.10.

($\Leftarrow$) For any two step sequences $s, t \in \mathbb{S}^*$, since $sut \equiv svt$, it follows that

$$(sut \div_R t) \div_L s = u \equiv v = (svt \div_R t) \div_L s.$$

Therefore, $u \equiv v$.

6. We want to show that $\pi_D(u) \approx \pi_D(v)$. Note that $\pi_D(A) \times \pi_D(B) \subseteq inl$, so

$$\pi_D(u) = \pi_D(x)\pi_D(A)\pi_D(B)\pi_D(y) \equiv \pi_D(x)\pi_D(B)\pi_D(A)\pi_D(C)\pi_D(y) = \pi_D(v).$$

□

Proposition 10.2. *If u and w are two step sequences over a generalised comtrace alphabet (E, sim, ser, inl) satisfying $u \equiv v$ then $\Sigma_u = \Sigma_v$.*

Proof. From Proposition 10.1(2), we know that $\equiv$ is *event-preserving*, i.e. for all $e \in E$, we have $|u|_e = |v|_e$. Since the enumeration of events in u and v depends only on the multiplicity of event occurrences in u and v, it follows that $\Sigma_u = \Sigma_v$. □

Thus, for a generalised comtrace $t = [u]$, we can define $\Sigma_t = \Sigma_u$. Furthermore, each enumeration of events specifies an invariant on the positions of any two event occurrences as shown in the next proposition.

Proposition 10.3. *Let u be a step sequence on a generalised comtrace alphabet (E, sim, ser, inl) and $\alpha, \beta \in \Sigma_u$ such that $l(\alpha) = l(\beta)$. Then*

1. *$pos_u(\alpha) \neq pos_u(\beta)$*
2. *If $pos_u(\alpha) < pos_u(\beta)$ and there is a step sequence v satisfying $v \equiv u$, then $pos_v(\alpha) < pos_v(\beta)$.*

Proof. 1. Follows from the fact that *sim* is irreflexive.

2. It suffices to show that if $pos_u(\alpha) < pos_u(\beta)$ and $\overline{v} \approx \overline{u}$, then $pos_v(\alpha) < pos_v(\beta)$. But this is clear from Definition 3.10 and the fact that *ser* and *inl* are irreflexive.

□

The following proposition ensures that if an invariant between the positions of two event occurrences is satisfied by the cancellation or projection of a generalised comtrace $[\overline{u}]$, then it is also satisfied by $[\overline{u}]$.

Proposition 10.4. *Let $\overline{u}$ be an enumerated step sequence on a generalised comtrace alphabet (E, sim, ser, inl) and $\alpha, \beta, \gamma \in \Sigma_u$ such that $\gamma \notin \{\alpha, \beta\}$. Then*

1. *$(\forall \overline{v} \in [\overline{u} \div_L \gamma].\, pos_{\overline{v}}(\alpha)\ \mathcal{R}\ pos_{\overline{v}}(\beta)) \implies (\forall \overline{w} \in [\overline{u}].\, pos_{\overline{w}}(\alpha)\ \mathcal{R}\ pos_{\overline{w}}(\beta))$*

2. $(\forall \overline{v} \in [\overline{u} \div_R \gamma].\, pos_{\overline{v}}(\alpha)\ \mathcal{R}\ pos_{\overline{v}}(\beta)) \implies (\forall \overline{w} \in [\overline{u}].\, pos_{\overline{w}}(\alpha)\ \mathcal{R}\ pos_{\overline{w}}(\beta))$

3. If $S \subseteq \Sigma_u$ *such that* $\{\alpha, \beta\} \subseteq S$, *then*

$$(\forall \overline{v} \in [\pi_S(\overline{u})].\, pos_{\overline{v}}(\alpha)\ \mathcal{R}\ pos_{\overline{v}}(\beta)) \implies (\forall \overline{w} \in [\overline{u}].\, pos_{\overline{w}}(\alpha)\ \mathcal{R}\ pos_{\overline{w}}(\beta))$$

where $\mathcal{R} \in \{\leq, \geq, <, >, =, \neq\}$.

Proof. 1. Assume that

$$\forall \overline{v} \in [\overline{v} \div_L \gamma].\, pos_{\overline{v}}(\alpha)\ \mathcal{R}\ pos_{\overline{v}}(\beta) \tag{10.1}$$

Suppose for a contradiction there is some $\overline{w} \in [\overline{v}]$ such that $\neg(pos_{\overline{w}}(\alpha)\ \mathcal{R}\ pos_{\overline{w}}(\beta)))$. Since $\gamma \notin \{\alpha, \beta\}$, we have $\neg(pos_{\overline{w} \div_L \gamma}(\alpha)\ \mathcal{R}\ pos_{\overline{w} \div_L \gamma}(\beta))$. But $\overline{w} \in [\overline{v}]$ implies $\overline{w} \div_L \gamma \equiv \overline{u} \div_L \gamma$. Hence, $\overline{w} \div_L \gamma \in [\overline{u} \div_L \gamma]$ and $\neg(pos_{\overline{w} \div_L \gamma}(\alpha)\ \mathcal{R}\ pos_{\overline{w} \div_L \gamma}(\beta))$, which contradicts the assumption (10.1).

2. Dually to (1).

3. Assume that

$$\forall \overline{v} \in [\pi_S(\overline{u})].\, pos_{\overline{v}}(\alpha)\ \mathcal{R}\ pos_{\overline{v}}(\beta) \tag{10.2}$$

Suppose for a contradiction there is some $\overline{w} \in [\overline{v}]$ such that $\neg(pos_{\overline{w}}(\alpha)\ \mathcal{R}\ pos_{\overline{w}}(\beta)))$. Since $\{\alpha, \beta\} \subseteq S$, we have $\neg(pos_{\pi_S(\overline{w})}(\alpha)\ \mathcal{R}\ pos_{\pi_S(\overline{w})}(\beta))$. But $\overline{w} \in [\overline{v}]$ implies $\pi_S(\overline{w}) \equiv \pi_S(\overline{u})$. Hence, $\pi_S(\overline{w}) \in [\pi_S(\overline{u})]$ and $\neg(pos_{\pi_S(\overline{w})}(\alpha)\ \mathcal{R}\ pos_{\pi_S(\overline{w})}(\beta))$, which contradicts the assumption (10.2).

□

10.2 Commutative Closure of Relational Structures

In this section, we develop the notion of *commutative closure* of a relational structure. It roughly corresponds to the notion of ◇-closure which is used to construct stratified order structure in Definition 9.2.

For a binary relation R on X, we let $R^{\leftrightarrows}$ denote the *symmetric closure* of R, i.e.,

$$R^{\leftrightarrows} \overset{df}{=} R \cup R^{-1}.$$

Definition 10.1. Let $G = (X, \Diamond, \sqsubset)$ be a relational structure and $\prec = \Diamond \cap \sqsubset^*$. Let $(X, \prec_0, \sqsubset_0) = (X, \prec, \sqsubset)^{\diamond}$. Then the *commutative closure* of the relational structure G is defined as

$$G^{\bowtie} \overset{df}{=} \left(X, {\prec_0}^{\leftrightarrows} \cup \Diamond, \sqsubset_0\right).$$

□

In the rest of this section, we will prove some useful properties of the commutative closure.

Proposition 10.5. *Let $G = (X, \Diamond, \sqsubset)$ be a relational structure and $\prec = \Diamond \cap \sqsubset^*$. If $(X, \prec_0, \sqsubset_0) = (X, \prec, \sqsubset)^{\diamond}$ is a stratified order structure then*

$$\prec_0 = ({\prec_0}^{\leftrightarrows} \cup \Diamond) \cap \sqsubset_0 .$$

Proof. ($\subseteq$) Since $(X, \prec_0, \sqsubset_0) = (X, \prec, \sqsubset)^{\diamond}$, by definition of $\diamond$-closure, $\prec_0 \subseteq \sqsubset_0$. Since we also have $\prec_0 \subseteq (\prec_0 \cup \Diamond)$, it follows that $\prec_0 \subseteq ({\prec_0}^{\leftrightarrows} \cup \Diamond) \cap \sqsubset_0$.

($\supseteq$) Suppose for a contradiction that $(x, y) \in ({\prec_0}^{\leftrightarrows} \cup \Diamond) \cap \sqsubset_0$ and $\neg(x \prec_0 y)$. There are two cases to consider:

- If $x \prec_0^{-1} y$ and $x \sqsubset_0 y$: Since $(X, \prec_0, \sqsubset_0)$ is a stratified order structure, it follows from Remark 8.1 that $y \prec_0 x \implies \neg(x \sqsubset_0 y)$, a contradiction.
- If $(x, y) \in \Diamond$ and $x \sqsubset_0 y$: Since $(X, \prec_0, \sqsubset_0) = (X, \prec, \sqsubset)^{\diamond}$, $\prec_0 = (\prec \cup \sqsubset)^* \circ \prec \circ (\prec \cup \sqsubset)^*$ and $\sqsubset_0 = (\prec \cup \sqsubset)^* \setminus id_X$. Since $x \sqsubset_0 y$ and $\neg(x \prec_0 y)$, it follows that $(x, y) \in (\sqsubset^* \setminus id_X)$. Since $(x, y) \in (\sqsubset^* \setminus id_X)$ and $(x, y) \in \Diamond$, we have $x \prec y$. Hence, $x \prec_0 y$, a contradiction.

Since either case leads to a contradiction, we get $\prec_0 \supseteq ({\prec_0}^{\leftrightarrows} \cup \Diamond) \cap \sqsubset_0$. □

Proposition 10.6 ([14, Proposition 3.3]). *Let S be a relational structure and $(X, \prec, \sqsubset) = S^{\diamond}$. Then $S^{\diamond}$ is a stratified order structure if and only if $\prec$ is irreflexive.* □

Proposition 10.7 ([14, Proposition 3.4]). *If S is a stratified order structure then $S = S^{\diamond}$.* □

Proposition 10.8. *If $G = (X, <\!\!>, \sqsubset)$ is a generalised stratified order structure then $G = G^{\bowtie}$.*

Proof. Since G is a generalised stratified order structure, by Definition 8.2, $S_G = (X, \prec_G, \sqsubset)$ is a stratified order structure. Hence, by Proposition 10.7, $S_G = S_G^{\diamond}$, which implies $\sqsubset = (\prec_G \cup \sqsubset)^* \setminus id_X$. But since S_G is a stratified order structure, $\prec_G \subseteq \sqsubset$. So $\sqsubset = \sqsubset^* \setminus id_X$. Let $\prec = <\!\!> \cap \sqsubset^*$. Then since $<\!\!>$ is irreflexive,

$$\prec = <\!\!> \cap \sqsubset^* = <\!\!> \cap (\sqsubset^* \setminus id_X) = <\!\!> \cap \sqsubset = \prec_G .$$

Hence, $(X, \prec, \sqsubset) = (X, \prec_G, \sqsubset)$ is a stratified order structure. By Proposition 10.7, $(X, \prec, \sqsubset) = (X, \prec, \sqsubset)^{\diamond}$. So from Definition 10.1, it follows that $G^{\bowtie} = \big(X, \prec^{\leftrightarrows} \cup <\!\!>, \sqsubset\big)$. Since $\prec \subseteq <\!\!>$ and (by Definition 8.2) $<\!\!>$ is symmetric, we have $\prec^{\leftrightarrows} \cup <\!\!> = <\!\!>$. Thus, $G = G^{\bowtie}$. □

Proposition 10.9. *If $G_1 = (X, <\!\!>_1, \sqsubset_1)$ and $G_2 = (X, <\!\!>_2, \sqsubset_2)$ are two relational structure such that $G_1 \subseteq G_2$ then $G_1^{\bowtie} \subseteq G_2^{\bowtie}$.*

Proof.

$G_1 \subseteq G_2$

$\Longrightarrow$ $\langle$ By definition of relational structure extension $\rangle$

$<\!\!>_1 \subseteq <\!\!>_2 \;\wedge\; \sqsubset_1 \subseteq \sqsubset_2$

$\Longrightarrow$ $\langle$ By properties of set-theoretical intersection $\rangle$

$(<\!\!>_1 \cap \sqsubset_1^*) \subseteq (<\!\!>_2 \cap \sqsubset_2^*) \wedge \sqsubset_1 \subseteq \sqsubset_2$

$\Longrightarrow$ $\langle$ By definition of $\diamond$-closure $\rangle$

$(X, <\!\!>_1 \cap \sqsubset_1^*, \sqsubset_1)^{\diamond} \subseteq (X, <\!\!>_2 \cap \sqsubset_2^*, \sqsubset_2)^{\diamond}$

$\Longrightarrow$ $\langle$ Let $(X, \prec_1', \sqsubset_1') = (X, <\!\!>_1 \cap \sqsubset_1^*, \sqsubset_1)^{\diamond}$ and $(X, \prec_2', \sqsubset_2') = (X, <\!\!>_2 \cap \sqsubset_2^*, \sqsubset_2)^{\diamond}$ $\rangle$

$(X, \prec_1', \sqsubset_1') \subseteq (X, \prec_2', \sqsubset_2')$

$\Longrightarrow$ $\langle$ By properties of $\cup$ and inverse operations and $<\!\!>_1 \subseteq <\!\!>_2$ $\rangle$

$(X, \prec_1'^{\leftrightarrows} \cup <\!\!>_1, \sqsubset_1') \subseteq (X, \prec_2'^{\leftrightarrows} \cup <\!\!>_2, \sqsubset_2')$

$\Longrightarrow$ $\langle$ By definition of commutative closure $\rangle$

$G_1^{\bowtie} \subseteq G_2^{\bowtie}$

□

10.3 Generalised Stratified Order Structures Generated by Step Sequences

We have seen how we can construct a stratified order structure from a step sequence over a comtrace alphabet in Definition 9.2. We will now introduce an analogous construction from a step sequence over a generalised comtrace alphabet to a generalised stratified order structure.

Let R be a binary relation on X. Then the *symmetric intersection* of R is defined as

$$si(R) \overset{df}{=} R \cap R^{-1}$$

And we define the *complement* of R to be

$$R^{\mathbf{C}} \overset{df}{=} (X \times X) \setminus R$$

Definition 10.2. Let s be a step sequence over a generalised comtrace alphabet (E, sim, ser, inl). For $\alpha, \beta \in \Sigma_s$

$$\alpha \Diamond_s \beta \iff (l(\alpha), l(\beta)) \in inl \tag{10.3}$$

$$\alpha \sqsubset_s \beta \iff (pos_s(\alpha) \leq pos_s(\beta) \wedge (l(\beta), l(\alpha)) \notin ser \cup inl) \tag{10.4}$$

$$\begin{array}{l} \alpha \prec_s \beta \iff pos_s(\alpha) < pos_s(\beta) \\ \wedge \left(\begin{array}{ll} & (l(\alpha), l(\beta)) \notin ser \cup inl \\ \vee & (\alpha, \beta) \in \Diamond_s \cap \left(si(\sqsubset_s^*) \circ \Diamond_s^{\mathbf{C}} \circ si(\sqsubset_s^*) \right) \\ \vee & \left(\begin{array}{ll} & (l(\alpha), l(\beta)) \in ser \\ \wedge & \exists \delta, \gamma \in \Sigma_s . \left(\begin{array}{ll} & pos_s(\delta) < pos_s(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser \\ \wedge & \alpha \sqsubset_s^* \delta \sqsubset_s^* \beta \wedge \alpha \sqsubset_s^* \gamma \sqsubset_s^* \beta \end{array} \right) \end{array} \right) \end{array} \right) \end{array} \tag{10.5}$$

We define the relational structure induced by s as

$$\xi_s \overset{df}{=} (\Sigma_s, \prec_s \cup \Diamond_s, \prec_s \cup \sqsubset_s)^{\bowtie}.$$

Proposition 10.10. *Let u, w are step sequences over a generalised comtrace alphabet (E, sim, ser, inl) such that $u(\approx \cup \approx^{-1})w$. Then*

1. *If $pos_u(\alpha) < pos_u(\beta)$ and $pos_w(\alpha) > pos_w(\beta)$ then there are x, y, A, B such that $\overline{u} = \overline{x}\overline{A}\,\overline{B}\overline{y}(\approx \cup \approx^{-1})\overline{x}\overline{B}\,\overline{A}\overline{y} = \overline{w}$ and $\alpha \in \overline{A}, \beta \in \overline{B}$.*
2. *If $pos_u(\alpha) = pos_u(\beta)$ and $pos_w(\alpha) > pos_w(\beta)$ then there are x, y, A, B, C such that $\overline{u} = \overline{x}\overline{A}\overline{y} \approx \overline{x}\overline{B}\,\overline{C}\overline{y} = \overline{w}$ and $\beta \in \overline{B}$ and $\alpha \in \overline{C}$.*

Proof. 1. Assume $pos_u(\alpha) < pos_u(\beta)$ and $pos_w(\alpha) > pos_w(\beta)$. Since $u(\approx \cup \approx^{-1})w$, we observe that

- If $\overline{u} = \overline{s}\overline{D}\overline{t} \approx \overline{s}\overline{E}\,\overline{F}\overline{t} = \overline{w}$, then $\forall \alpha, \beta \in \biguplus(\overline{u})$,
$$pos_u(\alpha) < pos_u(\beta) \implies pos_w(\alpha) < pos_w(\beta).$$
- If $\overline{u} = \overline{s}\overline{D}\,\overline{E}\overline{t} \approx \overline{s}\overline{F}\overline{t} = \overline{w}$, then $\forall \alpha, \beta \in \biguplus(\overline{u})$,
$$pos_u(\alpha) < pos_u(\beta) \implies pos_w(\alpha) \leq pos_w(\beta).$$

Either case contradicts the assumption that $pos_w(\alpha) > pos_w(\beta)$. Hence, it must be the case that
$$\overline{u} = \overline{x}\overline{A}\,\overline{B}\overline{y}(\approx \cup \approx^{-1})\overline{x}\overline{B}\,\overline{A}\overline{y} = \overline{w}$$
for some x, y, A, B. We will show that $\alpha \in \overline{A}$ and $\beta \in \overline{B}$. Suppose for a contradiction that $\alpha \notin \overline{A}$ or $\beta \notin \overline{B}$. Then

- If $\alpha \notin \overline{A}$, then $\forall \alpha, \beta \in \biguplus(\overline{x}) \cup \overline{B} \cup \biguplus(\overline{y})$,
$$pos_u(\alpha) < pos_u(\beta) \implies pos_w(\alpha) < pos_w(\beta),$$
a contradiction.
- If $\beta \notin \overline{B}$, then $\forall \alpha, \beta \in \biguplus(\overline{x}) \cup \overline{A} \cup \biguplus(\overline{y})$,
$$pos_u(\alpha) < pos_u(\beta) \implies pos_w(\alpha) < pos_w(\beta),$$
a contradiction.

Hence, $\overline{u} = \overline{xA}\,\overline{By}(\approx \cup \approx^{-1})\overline{xB}\,\overline{Ay} = \overline{w}$ where $\alpha \in \overline{A}$ and $\beta \in \overline{B}$ as desired.

2. Can be shown in a similar way to (1). □

Proposition 10.11. *Let s be a step sequence over a generalised comtrace alphabet (E, sim, ser, inl). If $\alpha, \beta \in \Sigma_s$, then*

1. $\alpha \diamond_s \beta \implies \forall u \in [s].\ pos_u(\alpha) \neq pos_u(\beta)$
2. $\alpha \sqsubset_s \beta \implies \forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$
3. $\alpha \prec_s \beta \implies \forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$

Proof. 1. Assume that $\alpha \diamond_s \beta$. Then, by (10.3), $(l(\alpha), l(\beta)) \in inl$. This implies that $l(\alpha) \neq l(\beta)$, so $\alpha \neq \beta$. Also since $inl \cap sim = \emptyset$, there is no step A where $\{l(\alpha), l(\beta)\} \in A$. Hence, $\forall u \in [s].\ pos_u(\alpha) \neq pos_u(\beta)$.

2. Assume that $\alpha \sqsubset_s \beta$. Suppose for a contradiction that $\exists u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$. Then must be some $u_1, u_1 \in [s]$ such that $u_1(\approx \cup \approx^{-1})u_2$ and $pos_{u_1}(\alpha) \leq pos_{u_1}(\beta)$ and $pos_{u_2}(\alpha) > pos_{u_2}(\beta)$. There are two cases:

- If $pos_{u_1}(\alpha) < pos_{u_1}(\beta)$ and $pos_{u_2}(\alpha) > pos_{u_2}(\beta)$, then it follows from Proposition 10.10(1) that there are x, y, A, B such that $\overline{u_1} = \overline{xA}\,\overline{By}(\approx \cup \approx^{-1})\overline{xB}\,\overline{Ay} = \overline{u_2}$ and $\alpha \in \overline{A}, \beta \in \overline{B}$. Hence, $(l(\alpha), l(\beta)) \in inl$. By (10.4), this contradicts that $\alpha \sqsubset_s \beta$.

- If $pos_{u_1}(\alpha) = pos_{u_1}(\beta)$ and $pos_{u_2}(\alpha) > pos_{u_2}(\beta)$, then it follows from Proposition 10.10(2) that there are x, y, A, B, C such that $\overline{u_1} = \overline{xAy} \approx \overline{xB}\,\overline{Cy} = \overline{u_2}$ and $\beta \in \overline{B}$ and $\alpha \in \overline{C}$. Thus, $(l(\beta), l(\alpha)) \in ser$. By (10.4), this again contradicts that $\alpha \sqsubset_s \beta$.

Since either case leads to a contradiction, we conclude $\forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$.

3. Assume that $\alpha \prec_s \beta$. Suppose for a contradiction that $\exists u \in [s].\ pos_u(\alpha) \geq pos_u(\beta)$. Then must be some $u_1, u_1 \in [s]$ such that $u_1(\approx \cup \approx^{-1})u_2$ and $pos_{u_1}(\alpha) < pos_{u_1}(\beta)$ and $pos_{u_2}(\alpha) \geq pos_{u_2}(\beta)$. There are two cases:

- If $pos_{u_1}(\alpha) < pos_{u_1}(\beta)$ and $pos_{u_2}(\alpha) = pos_{u_2}(\beta)$, then it follows from Proposition 10.10(2) that there are x, y, A, B, C such that $\overline{u_2} = \overline{x}\overline{A}\overline{y} \approx \overline{x}\overline{B}\ \overline{C}\overline{y} = \overline{u_1}$ and $\alpha \in \overline{B}$ and $\beta \in \overline{C}$. Thus, $(l(\alpha), l(\beta)) \in ser$ and $\neg(\alpha \Diamond_s \beta)$. Hence, it follows from (10.5) that
$$\exists \delta, \gamma \in \Sigma_s.\left(\begin{array}{ll} & pos_s(\delta) < pos_s(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser \\ \wedge & \alpha \sqsubset^*_s \delta \sqsubset^*_s \beta \wedge \alpha \sqsubset^*_s \gamma \sqsubset^*_s \beta \end{array}\right)$$
By (2) and transitivity of $\leq$, we have
$$\left(\begin{array}{ll} & \gamma \neq \delta \ \wedge \ (l(\delta), l(\gamma)) \notin ser \\ \wedge & (\forall u \in [s].\, pos_u(\alpha) \leq pos_u(\delta) \leq pos_u(\beta)) \\ \wedge & (\forall u \in [s].\, pos_u(\alpha) \leq pos_u(\gamma) \leq pos_u(\beta) \end{array}\right)$$
But since $\alpha, \beta \in \overline{B} \cup \overline{C} = \overline{A}$, it follows that $\{\gamma, \delta\} \subseteq \overline{A}$, which implies $pos_{u_2}(\gamma) = pos_{u_2}(\delta)$. Since we also have $pos_s(\delta) < pos_s(\gamma)$, it follows from Proposition 10.10(2) that there are z, w, D, E, F such that $\overline{z}\overline{D}\overline{w} \approx \overline{z}\overline{E}\ \overline{F}\overline{w}$ and $\delta \in \overline{E}$ and $\gamma \in \overline{F}$. Thus, $(l(\delta), l(\gamma)) \in ser$, a contradiction.

- If $pos_{u_1}(\alpha) < pos_{u_1}(\beta)$ and $pos_{u_2}(\alpha) > pos_{u_2}(\beta)$, then it follows from Proposition 10.10(1) that there are x, y, A, B such that $\overline{u_1} = \overline{x}\overline{A}\ \overline{B}\overline{y}(\approx \cup \approx^{-1})\overline{x}\overline{B}\ \overline{A}\overline{y} = \overline{u_2}$ and $\alpha \in \overline{A}, \beta \in \overline{B}$. Hence, $(l(\alpha), l(\beta)) \in inl$. Since we assume $\alpha \prec_s \beta$, by (10.5), it follows that $(\alpha, \beta) \in \Diamond_s \cap \left(si(\sqsubset^*_s) \circ \Diamond^{\mathbf{C}}_s \circ si(\sqsubset^*_s)\right)$. Hence, there must be some γ, δ such that $\alpha\ si(\sqsubset^*_s)\ \gamma \Diamond^{\mathbf{C}}_s \delta\ si(\sqsubset^*_s)\ \beta$. Observe that

$$\begin{array}{ll} & \alpha\ si(\sqsubset^*_s)\ \gamma \\ \Longrightarrow & \qquad \langle \text{ By definition of } si \ \rangle \\ & \alpha\ (\sqsubset^*_s)\ \gamma \ \wedge \ \gamma\ (\sqsubset^*_s)\ \alpha \\ \Longrightarrow & \qquad \langle \text{ By (2) and transitivity of } \leq \ \rangle \\ & (\forall u \in [s].\, pos_u(\alpha) \leq pos_u(\gamma)) \wedge (\forall u \in [s].\, pos_u(\gamma) \leq pos_u(\alpha)) \\ \Longrightarrow & \qquad \langle \text{ By logic } \rangle \\ & (\forall u \in [s].\, pos_u(\alpha) = pos_u(\gamma)) \\ \Longrightarrow & \qquad \langle \text{ Since } \alpha \in \overline{A} \ \rangle \\ & \{\alpha, \gamma\} \subseteq \overline{A} \end{array}$$

Similarly, since $\delta\ si(\sqsubset^*_s)\ \beta$, we can show that $\{\delta, \beta\} \subseteq \overline{B}$. Hence, since $\overline{x}\overline{A}\ \overline{B}\overline{y}(\approx \cup \approx^{-1})\overline{x}\overline{B}\ \overline{A}\overline{y}$, we get $A \times B \subseteq inl$. So $(l(\gamma), l(\delta)) \in inl$. But $\gamma \Diamond^{\mathbf{C}}_s \delta$ implies that $(l(\gamma), l(\delta)) \notin inl$, a contradiction.

Since either case leads to a contradiction, we conclude $\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$.

□

Proposition 10.12. *Let s be a step sequence over a generalised comtrace alphabet (E, sim, ser, inl) and $\xi_s = (\Sigma_s, \diamond, \sqsubset)$. If $\alpha, \beta \in \Sigma_s$, then*

1. $\alpha \diamond \beta \implies \forall u \in [s].\ pos_u(\alpha) \neq pos_u(\beta)$
2. $\alpha \sqsubset \beta \implies \forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$

Proof. 1. Let $\sqsubset_0 = \prec_s \cup \sqsubset_s$, $\diamond_0 = \prec_s \cup \diamond_s$ and $\prec_0 = \diamond_0 \cap \sqsubset_0^*$. We then let $\prec_1 = (\prec_0 \cup \sqsubset_0)^* \circ \prec_0 \circ (\prec_0 \cup \sqsubset_0)^*$. By Definitions 10.2 and 10.1, we have

$$\diamond = (\prec_1 \cup \diamond_0) \cup (\prec_1 \cup \diamond_0)^{-1}.$$

By Proposition 10.11, for $\alpha, \beta \in \Sigma_s$, we have

$$\alpha \sqsubset_0 \beta \implies \forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta) \tag{10.6}$$

$$\alpha \diamond_0 \beta \implies \forall u \in [s].\ pos_u(\alpha) \neq pos_u(\beta) \tag{10.7}$$

Hence, by transitivity of $\leq$, we have

$$\alpha \prec_0 \beta \implies \forall u \in [s].\ pos_u(\alpha) < pos_u(\beta) \tag{10.8}$$

But since $\prec_1 = (\prec_0 \cup \sqsubset_0)^* \circ \prec_0 \circ (\prec_0 \cup \sqsubset_0)^*$, by transitivity of $<$ and $\leq$, we have

$$\alpha \prec_1 \beta \implies \forall u \in [s].\ pos_u(\alpha) < pos_u(\beta) \tag{10.9}$$

Since $\diamond = (\prec_1 \cup \diamond_0) \cup (\prec_1 \cup \diamond_0)^{-1}$, from (10.7) and (10.9), it follows that

$$\alpha \diamond \beta \implies \forall u \in [s].\ pos_u(\alpha) \neq pos_u(\beta).$$

2. By Definitions 10.2 and 10.1, we have $\sqsubset = (\prec_0 \cup \sqsubset_0)^* \setminus id_{\Sigma_s}$. Hence, it follows from (10.7), (10.8) and transitivity of $<$ and $\leq$ that

$$\alpha \sqsubset \beta \implies \forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta).$$

□

Note that the definitions of non-serialisable steps, defined using only the relation *ser*, are still valid for the case of generalised comtraces. Moreover, the following results still hold.

Proposition 10.13. *Let A be a step over a generalised comtrace alphabet (E, sim, ser, inl), then*

1. *If A is non-serialisable to the left of $l(\alpha)$ for some $\alpha \in \overline{A}$, then*
$$\forall \beta \in \overline{A}.\ \alpha \sqsubset_A^* \beta.$$
2. *If A is non-serialisable to the right of $l(\beta)$ for some $\beta \in \overline{A}$, then*
$$\forall \alpha \in \overline{A}.\ \alpha \sqsubset_A^* \beta.$$
3. *If A is non-serialisable, then $\forall \alpha, \beta \in \overline{A}.\ \alpha \sqsubset_A^* \beta$.*

Proof. For all $\alpha, \beta \in \overline{A}$, $(l(\alpha), l(\beta)) \notin inl$. Hence, by (10.4),

$$\begin{aligned} \alpha \sqsubset_A \beta &\iff pos_A(\alpha) \leq pos_A(\beta) \wedge (l(\beta), l(\alpha)) \notin ser \cup inl \\ &\iff pos_A(\alpha) \leq pos_A(\beta) \wedge (l(\beta), l(\alpha)) \notin ser \end{aligned}$$

This is exactly the same to Definition 9.2. Hence, the proof is exactly the same to that of Proposition 9.5. □

Proposition 10.14. *Let A be a step over a generalised comtrace alphabet (E, sim, ser, inl) and $a \in A$. Then*

1. *There exists a unique $B \subseteq A$ such that $a \in B$, B is non-serialisable to the left of a, and*
$$A \neq B \implies A \equiv (A \setminus B)B.$$
2. *There exists a unique $C \subseteq A$ such that $a \in B$, C is non-serialisable to the right of a, and*
$$A \neq C \implies A \equiv C(A \setminus C).$$

Proof. Again since $\forall b, c \in A.\ (b, c) \notin inl$, $\sqsubset_A$ is defined in exactly the same way to Definition 9.2. Hence, the proof is the same to that of Proposition 9.6. □

Proposition 10.15. *Let s be a step sequence over a generalised comtrace alphabet* (E, sim, ser, inl) *and* $\xi_s = (\Sigma_s, \Diamond, \sqsubset)$. *Let* $\prec = \sqsubset \cup \Diamond$. *If* $\alpha, \beta \in \Sigma_s$, *then*

1. $$\begin{pmatrix} & (\forall u \in [s].\ pos_u(\alpha) \neq pos_u(\beta)) \\ \wedge & (\exists u \in [s].\ pos_u(\alpha) < pos_u(\beta)) \\ \wedge & (\exists u \in [s].\ pos_u(\alpha) > pos_u(\beta)) \end{pmatrix} \implies \alpha \Diamond \beta$$

2. $(\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)) \implies \alpha \prec \beta$

3. $(\alpha \neq \beta \ \wedge\ \forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)) \implies \alpha \sqsubset \beta$

Proof. 1. If $\begin{pmatrix} & (\forall u \in [s].\ pos_u(\alpha) \neq pos_u(\beta)) \\ \wedge & (\exists u \in [s].\ pos_u(\alpha) < pos_u(\beta)) \\ \wedge & (\exists u \in [s].\ pos_u(\alpha) > pos_u(\beta)) \end{pmatrix}$, then it follows from Proposition 10.10(1) that there are $u_1, u_2 \in [s]$ and x, y, A, B such that

$$\overline{u_1} = \overline{xA}\ \overline{By}(\approx \cup \approx^{-1})\overline{xB}\ \overline{Ay} = \overline{u_2}$$

and $\alpha \in \overline{A}, \beta \in \overline{B}$. Hence, $(l(\alpha), l(\beta)) \in inl$, which by (10.3) implies that $\alpha \Diamond_s \beta$. It then follows from Definitions 10.1 and 10.2 that $\alpha \Diamond \beta$.

2, 3. Assume $\forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$ and $\alpha \neq \beta$. Hence, we can choose $u_0 \in [s]$ where $\overline{u_0} = \overline{x_0}\ \overline{E_1} \ldots \overline{E_k}\ \overline{y_0}$ $(k \geq 1)$, E_1, E_k are non-serialisable, $\alpha \in \overline{E_1}$, $\beta \in \overline{E_k}$, and

$$\forall u_0' \in [s]. \begin{pmatrix} (\overline{u_0'} = \overline{x_0'}\ \overline{E_1'} \ldots \overline{E_{k'}'}\ \overline{y_0'} \ \wedge\ \alpha \in \overline{E_1'} \ \wedge\ \beta \in \overline{E_{k'}'}) \\ \implies weight(\overline{E_1} \ldots \overline{E_k}) \leq weight(\overline{E_1'} \ldots \overline{E_{k'}'}) \end{pmatrix} \tag{10.10}$$

We will prove by induction on $weight(\overline{E_1} \ldots \overline{E_k})$ that

$$(\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)) \implies \alpha \prec \beta \tag{10.11}$$

$$(\alpha \neq \beta \ \wedge\ \forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)) \implies \alpha \sqsubset \beta \tag{10.12}$$

Base Case:

When $weight(\overline{E_1}\dots\overline{E_k}) = 2$, then we consider two cases:

- If $\alpha \neq \beta$, $\forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$ and $\exists u \in [s].\ pos_u(\alpha) = pos_u(\beta)$, then it follows that

  - $\overline{u_0} = \overline{x_0}\{\alpha,\beta\}\overline{y_0}$, or
  - $\overline{u_0} = \overline{x_0}\{\alpha\}\{\beta\}\overline{y_0} \equiv \overline{x_0}\{\alpha,\beta\}\overline{y_0}$

  But since $\forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$, in either case, we must have $\{l(\alpha), l(\beta)\}$ is not serialisable to the right of $l(\beta)$. Hence, by Proposition 10.13(2), $\alpha \sqsubset_s^* \beta$. This by Definitions 10.1 and 10.2 implies that $\alpha \sqsubset \beta$.

- If $\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$, then it follows $\overline{u_0} = \overline{x_0}\{\alpha\}\{\beta\}\overline{y_0}$. Since $\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$, we must have $(l(\alpha), l(\beta)) \notin ser \cup inl$. This, by (10.3), implies that $\alpha \prec_s \beta$. Hence, from Definitions 10.1 and 10.2, we get $\alpha \prec \beta$.

From these two cases, since $\prec \subseteq \sqsubset$, it follows that (10.11) and (10.12) hold.

Inductive Step:

When $weight(\overline{E_1}\dots\overline{E_k}) > 2$, then $\overline{u_0} = \overline{x_0}\ \overline{E_1}\dots\overline{E_k}\ \overline{y_0}$ where $k \geq 1$. We need to consider two cases:

Case (i): If $\alpha \neq \beta$, $\forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$ and $\exists u \in [s].\ pos_u(\alpha) = pos_u(\beta)$, then there is some v_0 $\overline{v_0} = \overline{w_0}\ \overline{E}\ \overline{z_0}$ and $\alpha, \beta \in \overline{E}$. Either E is non-serialisable to the right of $l(\beta)$, or by Proposition 10.13(2) $\overline{v_0} = \overline{w_0}\ \overline{E}\ \overline{z_0} \equiv \overline{w_0'}\ \overline{E'}\ \overline{z_0'}$ where E' is non-serialisable to the right of $l(\beta)$. In either case, by Proposition 10.13(2), we have $\alpha \sqsubset_s^* \beta$. So it follows from Definitions 10.1 and 10.2 that $\alpha \sqsubset \beta$.

Case (ii): If $\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$, then it follows $\overline{u_0} = \overline{x_0}\ \overline{E_1}\dots\overline{E_k}\ \overline{y_0}$ where $k \geq 2$ and $\alpha \in \overline{E_1}, \beta \in \overline{E_k}$. We also know from the way we choose u_0 that E_1 is non-serialisable to the left of $l(\alpha)$ and E_2 is non-serialisable to the right of $l(\beta)$, otherwise condition (10.10) is not satisfied. If $(l(\alpha), l(\beta)) \notin ser \cup inl$, then by (10.3), $\alpha \prec_s \beta$. Hence, from Definitions 10.1 and 10.2, we get $\alpha \prec \beta$. Thus, we need to consider only when $(l(\alpha), l(\beta)) \in ser$ or $(l(\alpha), l(\beta)) \in inl$. There are three cases to consider:

- If $\overline{u_0} = \overline{x_0}\,\overline{E_1}\,\overline{E_2}\,\overline{y_0}$ where E_1 and E_2 are non-serialisable, then since we assume $\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$, it follows that $E_1 \times E_2 \not\subseteq ser$ and $E_1 \times E_2 \not\subseteq inl$. Hence, there are $\alpha_1, \alpha_2 \in \overline{E_1}$ and $\beta_1, \beta_2 \in \overline{E_2}$ such that $(l(\alpha_1), l(\beta_1)) \notin inl$ and $(l(\alpha_2), l(\beta_2)) \notin ser$. Since E_1 and E_2 are non-serialisable, by Proposition 10.13(3), $\alpha_1 \sqsubset_s^* \alpha_2$ and $\beta_2 \sqsubset_s^* \beta_1$. Also by 10.2, we know that $\alpha_1 <\!\!>_s \beta_2$ and $\alpha_2 {<\!\!>}_s^{\mathsf{C}} \beta_1$. Thus, by 10.2, we have $\alpha_1 \prec_s \beta_2$. Since E_1 and E_2 are non-serialisable, by Proposition 10.13(3), $\alpha \sqsubset_s^* \alpha_1 \prec_s \beta_2 \sqsubset_s^* \beta$. Hence, by Definitions 10.1 and 10.2, $\alpha \prec \beta$.

- If $\overline{u_0} = \overline{x_0}\,\overline{E_1} \ldots \overline{E_k}\,\overline{y_0}$ where $k \geq 3$ and $(l(\alpha), l(\beta)) \in inl$, then let $\gamma \in \overline{E_2}$. Observe that we have

  $$\overline{u_0} = \overline{x_0}\,\overline{E_1} \ldots \overline{E_k}\,\overline{y_0} \equiv \overline{x_1}\,\overline{E_1}\,\overline{w_1}\,\overline{F}\,\overline{z_1}\,\overline{E_k}\,\overline{y_1} \equiv \overline{x_2}\,\overline{E_1}\,\overline{w_2}\,\overline{F}\,\overline{z_2}\,\overline{E_k}\,\overline{y_2}$$

  such that F is non-serialisable and $weight(\overline{E_1}\,\overline{w_1}\,\overline{F})$, $weight(\overline{F}\,\overline{z_2}\,\overline{E_k})$ satisfy the minimal condition similarly to (10.10). Since from the way u_0 is chosen, we know that $\forall u \in [s].\ pos_u(\alpha) \leq pos_u(\gamma)$ and $\forall u \in [s].\ pos_u(\gamma) \leq pos_u(\beta)$, by applying the induction hypothesis, we can conclude that

  $$\alpha \sqsubset \gamma \sqsubset \beta \tag{10.13}$$

  So by transitivity of $\sqsubset$, we get $\alpha \sqsubset \beta$. But since we assume $(l(\alpha), l(\beta)) \in inl$, it follows that $\alpha <\!\!> \beta$. Hence, $(\alpha, \beta) \in \sqsubset \cap <\!\!> = \prec$.

- If $\overline{u_0} = \overline{x_0}\,\overline{E_1} \ldots \overline{E_k}\,\overline{y_0}$ where $k \geq 3$ and $(l(\alpha), l(\beta)) \in ser$, then we observe from how u_0 is chosen that

  $$\forall \gamma \in \biguplus(\overline{E_1} \ldots \overline{E_k}).\ (\forall u \in [s].\ pos_{u_0}(\alpha) \leq pos_{u_0}(\gamma) \leq pos_{u_0}(\beta))$$

  Similarly to how we show (10.13), we can prove that

  $$\forall \gamma \in \biguplus(\overline{E_1} \ldots \overline{E_k}) \setminus \{\alpha, \beta\}.\ \alpha \sqsubset \gamma \sqsubset \beta \tag{10.14}$$

  We next want to show that

  $$\exists \delta, \gamma \in \biguplus(\overline{E_1} \ldots \overline{E_k}).\ (pos_{u_0}(\delta) < pos_{u_0}(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser) \tag{10.15}$$

Suppose for a contradiction that (10.15) does not hold, then

$$\forall \delta, \gamma \in \biguplus(\overline{E_1} \ldots \overline{E_k}).\ (pos_{u_0}(\delta) < pos_{u_0}(\gamma) \implies (l(\delta), l(\gamma)) \in ser) \quad (10.16)$$

It follows that $\overline{u_0} = \overline{x_0}\,\overline{E_1} \ldots \overline{E_k}\,\overline{y_0} \equiv \overline{x_0}\,\overline{E}\,\overline{y_0}$, which contradicts that

$$\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$$

Hence, we have shown (10.15). Let $\delta, \gamma \in \biguplus(\overline{E_1} \ldots \overline{E_k})$ be event occurrences satisfying $pos_{u_0}(\delta) < pos_{u_0}(\gamma)$ and $(l(\delta), l(\gamma)) \notin ser$. By (10.14), we also have that $\alpha(\sqsubset \cup\, id_{\Sigma_s})\delta(\sqsubset \cup\, id_{\Sigma_s})\beta$ and $\alpha(\sqsubset \cup\, id_{\Sigma_s})\gamma(\sqsubset \cup\, id_{\Sigma_s})\beta$. If $\alpha \prec \delta$ or $\delta \prec \beta$ or $\alpha \prec \gamma$ or $\gamma \prec \beta$, then by (C4) of Definition 8.1, $\alpha \prec \beta$. Otherwise, by Definitions 10.1 and 10.2, we have $\alpha \sqsubset_s^* \delta \sqsubset_s^* \beta$ and $\alpha \sqsubset_s^* \gamma \sqsubset_s^* \beta$. But since $pos_{u_0}(\delta) < pos_{u_0}(\gamma)$ and $(l(\delta), l(\gamma)) \notin ser$, by 10.2, $\alpha \prec_s \beta$. So it follows from Definitions 10.1 and 10.2 that $\alpha \prec \beta$.

Thus, we have shown (10.11) and (10.12) as desired. □

Proposition 10.16. *Let s be a step sequence over a generalised comtrace alphabet (E, sim, ser, inl), $\xi_s = (\Sigma_s, \Diamond, \sqsubset)$, and $\prec = \Diamond \cap \sqsubset$. If $\alpha, \beta \in \Sigma_s$, then*

1. $\alpha \Diamond \beta \iff \forall u \in [s].\ pos_u(\alpha) \neq pos_u(\beta)$
2. $\alpha \sqsubset \beta \iff \forall u \in [s].\ pos_u(\alpha) \leq pos_u(\beta)$
3. $\alpha \prec \beta \iff \forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$
4. *If $l(\alpha) = l(\beta)$ and $pos_s(\alpha) < pos_s(\beta)$, then $\alpha \prec \beta$*

Proof. 1. Follows directly from Proposition 10.12(1) and Proposition 10.15(1, 2).

2. Follows directly from Proposition 10.12(2) and Proposition 10.15(3).

3.
$$\begin{aligned}
& \alpha \prec \beta \\
\iff & \qquad \langle \text{ Since } \prec_s = \Diamond_s \cap \sqsubset_s \ \rangle \\
& \alpha \Diamond_s \beta \wedge \alpha \sqsubset_s \beta \\
\iff & \qquad \langle \text{ From (1) and (2) } \rangle \\
& \forall u \in [s].\ (pos_u(\alpha) \neq pos_u(\beta) \wedge pos_u(\alpha) \leq pos_u(\beta)) \\
\iff & \qquad \langle \text{ By logic } \rangle \\
& \forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)
\end{aligned}$$

4. Assume that $l(\alpha) = l(\beta)$ and $pos_s(\alpha) < pos_s(\beta)$. Then, by Proposition 10.3(2), we know $\forall u \in [s].\ pos_u(\alpha) < pos_u(\beta)$. Hence, it follows from (3) that $\alpha \prec \beta$.

□

Theorem 10.1. *Let s be a step sequence over a generalised comtrace alphabet (E, sim, ser, inl). Then*

$$\xi_s = \left(\Sigma_s, \bigcap_{u \in [s]} \lhd_u^{\leftrightarrows}, \bigcap_{u \in [s]} \lhd_u^{\frown} \right). \tag{10.17}$$

Proof. Let $\xi_s = (\Sigma_s, <\!\!>, \sqsubset)$ and $\alpha, \beta \in \Sigma_s$. We have

$$
\begin{array}{ll}
 & \alpha <\!\!> \beta \\
\Longleftrightarrow & \qquad \langle \text{ By Proposition 10.16(1) } \rangle \\
 & \forall u \in [s].\ pos_u(\alpha) \neq pos_u(\gamma) \\
\Longleftrightarrow & \qquad \langle \text{ By logic } \rangle \\
 & \forall u \in [s].\ (pos_u(\alpha) < pos_u(\beta) \vee pos_u(\alpha) > pos_u(\beta)) \\
\Longleftrightarrow & \qquad \langle \text{ By definition of } \lhd_u \ \rangle \\
 & (\alpha, \beta) \in \bigcap_{u \in [s]} \lhd_u^{\leftrightarrows}
\end{array}
$$

We also have

$$
\begin{array}{ll}
 & \alpha \sqsubset \beta \\
\Longleftrightarrow & \qquad \langle \text{ By Proposition 10.16(2) } \rangle \\
 & \forall u \in [s].\ pos_u(\alpha) \leq pos_u(\gamma) \\
\Longleftrightarrow & \qquad \langle \text{ By definition of } \lhd_u^{\frown} \ \rangle \\
 & (\alpha, \beta) \in \bigcap_{u \in [s]} \lhd_u^{\frown\leftrightarrows}
\end{array}
$$

Hence, we conclude that

$$\xi_s = (\Sigma_s, <\!\!>, \sqsubset) = \left(\Sigma_s, \bigcap_{u \in [s]} \lhd_u^{\leftrightarrows}, \bigcap_{u \in [s]} \lhd_u^{\frown} \right)$$

□

Proposition 10.17. *Let s be a step sequence over a generalised comtrace alphabet (E, sim, ser, inl). Then $\xi_s = (\Sigma_s, <\!\!>, \sqsubset)$ is a generalised stratified order structure.*

Proof. Since $\Diamond = \bigcap_{u \in [s]} \lhd_u^{\leftrightarrows}$ and $\lhd_u^{\leftrightarrows}$ is irreflexive and symmetric, $\Diamond$ is irreflexive and symmetric. Since $\sqsubset = \bigcap_{u \in [s]} \lhd_u^{\frown}$ and $\lhd_u^{\frown}$ is irreflexive, $\sqsubset$ is irreflexive.

Let $\prec = \Diamond \cap \sqsubset$, it remains to show that $S = (\Sigma, \prec, \sqsubset)$ is a stratified order structure, i.e., S satisfies the conditions C1–C4 of Definition 8.1. Since $\sqsubset$ is irreflexive, C_1 is satisfied. Since $\prec = \Diamond \cap \sqsubset$ implies $\prec \subseteq \sqsubset$, C_2 is satisfied. Assume $\alpha \sqsubset \beta \sqsubset \gamma$ and $\alpha \neq \gamma$. Then

$$\begin{aligned}
& \alpha \sqsubset \beta \sqsubset \gamma \\
\Longrightarrow & \qquad \langle \text{ By (10.17) } \rangle \\
& (\alpha, \beta) \in \textstyle\bigcap_{u \in [s]} \lhd_u^{\frown} \wedge (\beta, \gamma) \in \bigcap_{u \in [s]} \lhd_u^{\frown} \\
\Longrightarrow & \qquad \langle \text{ By definition of } \lhd_u \rangle \\
& (\forall u \in [s].\, pos_u(\alpha) \leq pos_u(\beta)) \ \wedge\ (\forall u \in [s].\, pos_u(\alpha) \leq pos_u(\gamma)) \\
\Longrightarrow & \qquad \langle \text{ By transitivity of } \leq \text{ and the assumption that } \alpha \neq \gamma \rangle \\
& \forall u \in [s].\, pos_u(\alpha) \leq pos_u(\gamma) \wedge \alpha \neq \gamma \\
\Longrightarrow & \qquad \langle \text{ By definition of } \lhd_u \rangle \\
& (\alpha, \gamma) \in \textstyle\bigcap_{u \in [s]} \lhd_u^{\frown} \\
\Longrightarrow & \qquad \langle \text{ By (10.17) } \rangle \\
& \alpha \sqsubset \gamma
\end{aligned}$$

Hence, C3 is satisfied. Next we assume that $\alpha \prec \beta \sqsubset_s \gamma$. Then

$$\begin{aligned}
& \alpha \prec \beta \sqsubset \gamma \\
\Longrightarrow & \qquad \langle \text{ By (10.17) and } \prec = \Diamond \cap \sqsubset \rangle \\
& (\alpha, \beta) \in \textstyle\bigcap_{u \in [s]} (\lhd_u^{\frown} \cap \lhd_u^{\leftrightarrows}) \wedge (\beta, \gamma) \in \bigcap_{u \in [s]} (\lhd_u^{\frown} \cap \lhd_u^{\leftrightarrows}) \\
\Longrightarrow & \qquad \langle \text{ By definition of } \lhd_u \rangle \\
& (\forall u \in [s].\, (pos_u(\alpha) \leq pos_u(\beta) \wedge pos_u(\alpha) \neq pos_u(\beta))) \\
& \ \wedge (\forall u \in [s].\, (pos_u(\alpha) \leq pos_u(\gamma) \wedge pos_u(\alpha) \neq pos_u(\gamma))) \\
\Longrightarrow & \qquad \langle \text{ By logic } \rangle \\
& (\forall u \in [s].\, pos_u(\alpha) < pos_u(\beta)) \ \wedge\ (\forall u \in [s].\, pos_u(\alpha) < pos_u(\gamma)) \\
\Longrightarrow & \qquad \langle \text{ By transitivity of } < \rangle \\
& \forall u \in [s].\, pos_u(\alpha) < pos_u(\gamma) \\
\Longrightarrow & \qquad \langle \text{ By definition of } \lhd_u \text{ and logic } \rangle \\
& (\alpha, \gamma) \in \textstyle\bigcap_{u \in [s]} (\lhd_u^{\frown} \cap \lhd_u^{\leftrightarrows}) \\
\Longrightarrow & \qquad \langle \text{ By (10.17) } \rangle \\
& \alpha \prec \gamma
\end{aligned}$$

Similarly, we can show $\alpha \sqsubset \beta \prec \gamma \implies \alpha \prec \gamma$. Thus, C4 is satisfied. □

By Proposition 10.3, for each step sequence s over a generalised comtrace alphabet (E, sim, ser, inl), we will call ξ_s the *generalised stratified order structure* induced by the step sequence s.

10.4 Generalised Stratified Order Structures Generated by Generalised Comtraces

In this section, we want to show that the construction from Definition 10.2 indeed yields a generalised stratified order structure representation of comtraces. But before doing so, we need some preliminary definitions and results.

Definition 10.3 ([10, 11]). Let $G = (X, <\!\!>, \sqsubset)$ be a generalised stratified order structure. A stratified order $\lhd$ on X is an *stratified order extension* of G if for all $\alpha, \beta \in X$, the following hold

$$\alpha <\!\!> \beta \implies \alpha \lhd^{\leftrightarrows} \beta$$
$$\alpha \sqsubset \beta \implies \alpha \lhd^{\frown} \beta$$

The set of all stratified order extensions of G is denoted as $ext(G)$.

Proposition 10.18. *Let s be a step sequence over a generalised comtrace alphabet (E, sim, ser, inl). Then $\lhd_s \in ext(\xi_s)$.*

Proof. Let $\xi_s = (\Sigma, <\!\!>, \sqsubset)$. By Proposition 10.16, for all $\alpha, \beta \in \Sigma$,

$$\alpha <\!\!> \beta \implies pos_s(\alpha) \neq pos_s(\beta) \implies \alpha \lhd_s \beta \vee \beta \lhd_s \alpha \implies \alpha \lhd_s^{\leftrightarrows} \beta$$
$$\alpha \sqsubset \beta \implies pos_s(\alpha) \leq pos_s(\beta) \implies \alpha \lhd_s^{\frown} \beta$$

Hence, by Definition 10.3, we get $\lhd_s \in ext(\xi_s)$. □

Proposition 10.19. *Let s be a step sequence over a generalised comtrace alphabet $\theta = (E, sim, ser, inl)$. If $\lhd \in ext(\xi_s)$, then there is a step sequence u over θ such that $\lhd = \lhd_u$.*

Proof. Let $\xi_s = (\Sigma_s, \Diamond, \sqsubset)$ and $\Omega_\lhd = B_1 \ldots B_k$. We will show that $u = l(B_1) \ldots l(B_k)$ is a step sequence such that $\lhd = \lhd_u$.

Suppose $\alpha, \beta \in B_i$ are two distinct event occurrences such that $(l(\alpha), l(\beta)) \notin sim$. Then $pos_s(\alpha) \neq pos_s(\beta)$, which by Proposition 10.16 implies that $\alpha \Diamond \beta$. Since $\lhd \in ext(\xi_s)$, by Definition 10.3, $\alpha \lhd \beta$ or $\beta \lhd \alpha$ contradicting $\alpha, \beta \in B_i$. Thus, we have shown for all B_i $(1 \leq i \leq k)$,

$$\alpha, \beta \in B_i \wedge \alpha \neq \beta \implies (l(\alpha), l(\beta)) \notin sim \tag{10.18}$$

By Proposition 10.3(2), if $e^{(i)}, e^{(j)} \in \Sigma_s$ and $i \neq j$ then $\forall u \in [s]. \ pos_u(e^{(i)}) \neq pos_u(e^{(j)})$. So it follows from Proposition 10.16(1) that $e^{(i)} \Diamond e^{(j)}$. Since $\lhd \in ext(\xi_s)$, by Definition 10.3,

$$\text{If } e^{(k_0)} \in B_k \text{ and } e^{(m_0)} \in B_m \text{ then } k_0 \neq m_0 \iff k \neq m \tag{10.19}$$

From (10.18) it follows that u is a step sequence over θ. Also by (10.19), $pos_u^{-1}(i) = B_i$ and $|l(B_i)| = |B_i|$ for all i. Hence, $\Omega_\lhd = \Omega_{\lhd_u}$, which implies $\lhd = \lhd_u$. □

We next want to show that two step sequences over the same generalised comtrace alphabet induce the same generalised stratified order structure if and only if they belong to the same generalised comtrace (Theorem 10.2 below). The proof of an analogous result for comtraces from [14] is simpler because every comtrace has a unique canonical representation that can be easily constructed. Since generalised comtraces do not have a unique canonical representation as defined in Definition 4.2, to simplify our proofs, we have to find another unique representation of generalised comtraces which can be easily constructed.

Let R be a binary relation on a set X. We says R is a *well-ordering* on a set S if R is a total order on S and every non-empty subset of S has a least element in this ordering. When R is a well-ordering on X, we say that X is *well-ordered* by R or R *well-orders* X.

Proposition 10.20. *If R is a total order on a finite set X, then R is a well-ordering.*

Proof. We prove this by induction on $|X|$. If $|X| = 0$ then by definition R well-orders X. Now we want to show that it also holds for $|X| > 0$. For any non-empty $S \subset X$, we have $R|_{Y\times Y}$ is a total order on S. Hence, by induction hypothesis, S is well-ordered and hence it has a least element. It remains to show that X also has a least element. We pick an arbitrary element $x \in X$ and consider the set $Y = X \setminus \{x\}$. Since $R|_{Y\times Y}$ is a total order on Y, by induction hypothesis, Y is well-ordered and hence has a least element y. Since R is a total order on X, x and y are comparable. If xRy then x is the least element of X. Otherwise, y is the least element of X. □

Definition 10.4. Let $\mathbb{S}$ be the set of all possible steps of a generalised comtrace concurrent alphabet $\theta = (E, ser, sim, inl)$ and assume that we have a well-ordering $<_E$ on E. Then we can define a *step order* $<^{st}$ on $\mathbb{S}$ as following:

$$A <^{st} B \iff |A| > |B| \ \lor \ (|A| = |B| \land A \neq B \land min_{<_E}(A \setminus B) <_E min_{<_E}(B \setminus A)) \tag{10.20}$$

where $min_{<_E}(X)$ denotes the least element of the set $X \subseteq E$ with respect to $<_E$.

Let $A_1 \ldots A_n$ and $B_1 \ldots B_m$ be two sequences in $\mathbb{S}^*$. We define a *lexicographic order* $<^{lex}$ on step sequences as following:

$$A_1 \ldots A_n <^{lex} B_1 \ldots B_m \iff \exists k > 0. \big((\forall i < k.\ A_i = B_i) \land (A_k <^{st} B_k \lor k > n)\big) \tag{10.21}$$

Proposition 10.21. *Let $\mathbb{S}$ be the set of all possible steps of a generalised comtrace concurrent alphabet $\theta = (E, ser, sim, inl)$ and $<_E$ be a well-ordering on E. Then*

1. *$<^{st}$ well-orders $\mathbb{S}$*

2. *$<^{lex}$ well-orders $\mathbb{S}^*$*

Proof. 1. Since $\mathbb{S}$ is finite, by Proposition 10.20, we only need to show that if $A, B \in \mathbb{S}$ then $A <^{st} B$ or $B <^{st} A$ or $A = B$. Assume $A \neq B$. If $|A| < |B|$ or $|A| > |B|$ then it follows from (10.20) that $A <^{st} B$ or $B <^{st} A$. Otherwise, $|A| = |B|$ and $A \neq B$. Hence, $A \not\subseteq B$ and $B \not\subseteq A$, which implies $A \setminus B \neq \emptyset$ and $B \setminus A \neq \emptyset$ and $(A \setminus B) \cap (B \setminus A) = \emptyset$. Hence, $min_{<_E}(A \setminus B)$ and $min_{<_E}(B \setminus A)$ are comparable with respect to $<_E$. Since $(A \setminus B) \cap (B \setminus A) = \emptyset$, we also knows that $min_{<_E}(A \setminus B) \neq min_{<_E}(B \setminus A)$. Thus, $min_{<_E}(A \setminus B) <_E min_{<_E}(B \setminus A)$ or

$min_{<_E}(B \setminus A) <_E min_{<_E}(A \setminus B)$, which by (10.20) implies $A <^{st} B$ or $B <^{st} A$.

2. Since $\mathbb{S}^*$ is finite, by Proposition 10.20, we only need to show that if $u, v \in \mathbb{S}$ then $u <^{lex} v$ or $v <^{lex} u$ or $u = v$. Assume $u \neq v$, $u = A_1 \dots A_n$ and $v = B_1 \dots B_m$. Without loss of generality we can assume that $n \leq m$. We will prove the result by induction on n. When $n = \lambda$, then by (10.21) we have $u <^{lex} v$. When $n > 0$, by induction hypothesis, $u' = A_1 \dots A_n$ and v are comparable. If $v <^{lex} u'$, then by (10.21) $v <^{lex} u$. Otherwise, $u' <^{lex} v$, which implies that there is some k such that $0 < k \leq n$ and $(\forall i < k.\ A_i = B_i) \wedge (A_k <^{st} B_k \vee k > (n-1))$. If $k < n$, then by (10.21) we have $u <^{lex} v$. Otherwise, $k = n$, which implies $\forall i < n.\ A_i = B_i$. Since $u \neq v$, we have $A_n <_E B_n$ or $B_n <_E A_n$. Hence, it follows from (10.21) that $u <^{lex} v$ or $v <^{lex} u$. □

Lemma 10.1. *Let s be a step sequence over a generalised comtrace alphabet $\theta = (E, ser, sim, inl)$ and $<_E$ be a well-ordering on E. Let $u = A_1 \dots A_n$ be the least element of the generalised comtrace $[s]$ with respect to the well-ordering $<^{lex}$. Let $\xi_s = (\Sigma, \diamond, \sqsubset)$ and $\prec = \diamond \cap \sqsubset$. Let $mins_{\prec}(X)$ denote the set of all minimal elements of X with respect to $\prec$ and define*

$$Z(X) \overset{df}{=} \{Y \mid Y \subseteq mins_{\prec}(X) \wedge (\forall \alpha, \beta \in Y.\ \alpha \neq \beta \implies \neg(\alpha \diamond \beta)) \\ \wedge \forall \alpha \in Y.\ \forall \beta \in X \setminus Y.\ \neg(\beta \sqsubset \alpha)\} \qquad (10.22)$$

Let $\overline{u} = \overline{A_1} \dots \overline{A_n}$ be the enumerated step sequence of u. Then A_i is the least element of the set $\{l[Y] \mid Y \in Z(\Sigma \setminus \biguplus(\overline{A_1} \dots \overline{A_{i-1}}))\}$ with respect to the well-ordering $<^{st}$.

Proof. We first notice that by Proposition 10.16(4), if $e^{(i)}, e^{(j)} \in \Sigma$ and $i < j$ then $e^{(i)} \prec e^{(j)}$. Hence, for all $\alpha, \beta \in mins_{\prec}(X)$, where $X \subseteq \Sigma$, we have $l(\alpha) \neq l(\beta)$. This ensures that if $Y \in Z(X)$ and $X \subseteq \Sigma$ then $|Y| = |l(Y)|$.

For all $\alpha \in \overline{A_1}$ and $\beta \in \Sigma$, $pos_s(\beta) \geq pos_s(\alpha)$. Hence, by Proposition 10.16(3), $\neg(\beta \prec \alpha)$. Thus,

$$\overline{A_1} \subseteq mins_{\prec}(X) \qquad (10.23)$$

For all $\alpha, \beta \in \overline{A_1}$, since $pos_s(\beta) = pos_s(\alpha)$, by Proposition 10.16(1), we have

$$\neg(\alpha \diamond \beta) \qquad (10.24)$$

For any $\alpha \in \overline{A_1}$ and $\beta \in \Sigma \setminus \overline{A_1}$, since $pos_s(\beta) < pos_s(\alpha)$, by Proposition 10.16(2),

$$\neg(\beta \sqsubset \alpha) \tag{10.25}$$

From (10.23), (10.24) and (10.25), we know that $\overline{A_1} \in Z(\Sigma)$. Hence, $Z(\Sigma) \neq \emptyset$. This ensures the least element of $\{l[Y] \mid Y \in Z(\Sigma)\}$ with respect to $<^{st}$ is well-defined.

Let $Y_0 \in Z(\Sigma)$ such that $B_0 = l(Y_0)$ be the least element of $\{l[Y] \mid Y \in Z(\Sigma)\}$ with respect to $<^{st}$. We want to show that $A_1 = B_0$. Since $<^{st}$ is a well-ordering, we know that $A_1 <^{st} B_0$ or $B_0 <^{st} A_1$ or $A_1 = B_0$. But since $\overline{A_1} \in Z(\Sigma)$ and B_0 be the least element of the set $\{l[B] \mid B \in Z(\Sigma)\}$, $\neg(A_1 <^{st} B_0)$. Hence, to show that $A_1 = B_0$, it only remains to show that $\neg(B_0 <^{st} A_1)$.

Suppose for a contradiction that $B_0 <^{st} A_1$. We first want to show that for every $W \subseteq Y_0$ there is an enumerated step sequence v such that

$$\overline{v} = W_0\overline{v_0} \equiv \overline{A_1} \ldots \overline{A_n} \text{ and } W \subseteq W_0 \subseteq Y_0 \tag{10.26}$$

We will prove this by induction on $|W|$.

Base Case:

When $|W| = 1$, we let $\{\alpha_0\} = W$. We choose $\overline{v_1} = \overline{E_0} \ldots \overline{E_k}\overline{y_1} \equiv \overline{A_1} \ldots \overline{A_n}$ and $\alpha_0 \in \overline{E_k}$ ($k \geq 0$) such that for all $\overline{v'} = \overline{E'_0} \ldots \overline{E'_{k'}}\, \overline{y'_1} \equiv \overline{A_1} \ldots \overline{A_n}$ and $\alpha_0 \in \overline{E'_{k'}}$, we have

(i) $weight(\overline{E_0} \ldots \overline{E_k}) \leq weight(\overline{E'_0} \ldots \overline{E'_{k'}})$, and

(ii) $weight(\overline{E_{k-1}}\, \overline{E_k}) \leq weight(\overline{E'_{k'-1}}\, \overline{E'_{k'}})$.

We then consider only $\overline{w} = \overline{E_0} \ldots \overline{E_k}$. We observe that because of the way we chose $\overline{v_1}$, we have

$$\forall \beta \in \biguplus(\overline{w}).\ \beta \neq \alpha_0 \implies \forall t \in [w].\ pos_t(\beta) \leq pos_t(\alpha_0)$$

Hence, since $\overline{w} = \overline{u} \div_R \overline{v_0}$, it follows from Proposition 10.4(1, 2) that

$$\forall \beta \in \biguplus(\overline{w}).\ \beta \neq \alpha_0 \implies \forall t \in [A_1 \ldots A_n].\ pos_t(\beta) \leq pos_t(\alpha_0)$$

Then it follows from Proposition 10.16(2) that

$$\forall \beta \in \biguplus(\overline{w}).\ \beta \neq \alpha_0 \implies \beta \sqsubset \alpha_0 \tag{10.27}$$

By the way Y_0 was chosen, we know that

$$\forall \alpha \in Y_0.\ \forall \beta \in \Sigma \setminus Y_0.\ \neg(\beta \sqsubset \alpha).$$

This and (10.27) imply that

$$\biguplus(\overline{w}) = (\overline{E_0} \cup \ldots \cup \overline{E_k}) \subseteq Y_0 \tag{10.28}$$

We claim that for every $\alpha \in \overline{E_i}$ and $\beta \in \overline{E_j}$ $(0 \leq i < j \leq k)$,

$$\{\alpha\}\{\beta\} \equiv \{\alpha, \beta\} \tag{10.29}$$

Suppose not. Then either $[\{\alpha\}\{\beta\}] = \{\{\alpha\}\{\beta\}\}$ or $[\{\alpha\}\{\beta\}] = \{\{\alpha\}\{\beta\}, \{\beta\}\{\alpha\}\}$. In either case, we have $\forall t \in [\{l(\alpha)\}\{l(\beta)\}].\ pos_t(\alpha) \neq pos_t(\beta)$. Since $\{\alpha\}\{\beta\} \equiv \pi_{\{\alpha,\beta\}}(\overline{u})$, by Proposition 10.4(3), $\forall t \in [u].\ pos_t(\alpha) \neq pos_t(\beta)$, which by Proposition 10.16 implies $\alpha \diamond \beta$. This contradicts that $Y_0 \in Z(\Sigma)$ and $\alpha, \beta \in \Sigma(\overline{w}) \subseteq Y_0$. Thus, we have shown (10.29), which implies that for all $\alpha \in \overline{E_i}$ and $\beta \in \overline{E_j}$ $(0 \leq i < j \leq k)$, $(l(\alpha), l(\beta)) \in ser$. Then $\overline{E_0} \ldots \overline{E_k} \equiv \overline{E_0} \cup \ldots \cup \overline{E_k}$. Hence, there exists a step sequence v_1'' such that

$$\overline{v_1''} = (\overline{E_0} \cup \ldots \cup \overline{E_k})\overline{v_1} \equiv \overline{A_1} \ldots \overline{A_n},$$

where $\{\alpha_0\} \subseteq (\overline{E_0} \cup \ldots \cup \overline{E_k}) \subseteq Y_0$.

Inductive Step:

When $|W| > 1$, we pick an element $\beta_0 \in W$. By applying the induction hypothesis on $W \setminus \{\beta_0\}$, we get a step sequence v_2 such that

$$\overline{v_2} = \overline{F_0}\overline{y_2} \equiv \overline{A_1} \ldots \overline{A_n}$$

where $W \setminus \{\beta_0\} \subseteq \overline{F_0} \subseteq Y_0$. If $W \subseteq \overline{F_0}$, we are done. Otherwise, proceeding like the base case, we construct a step sequence v_3 such that

$$\overline{v_3} = \overline{F_0}\ \overline{F_1}\overline{y_3} \equiv \overline{A_1} \ldots \overline{A_n}$$

and $\{\beta_0\} \subseteq \overline{F_1} \subseteq Y_0$. Since $\overline{F_0} \subseteq Y_0$, $W \subseteq \overline{F_0} \cup \overline{F_1} \subseteq Y_0$.

Similarly to how we proved (10.29), we can show that

$$\forall \alpha \in \overline{F_0}.\ \forall \beta \in \overline{F_1}.\ \{\alpha\}\{\beta\} \equiv \{\alpha, \beta\}$$

This means that $\alpha \in \overline{F_0}$ and $\beta \in \overline{F_1}$, $(l(\alpha), l(\beta)) \in ser$. Hence, $\overline{F_0 F_1} \equiv \overline{F_0} \cup \overline{F_1}$. Hence, there is a step sequence v_4 such that

$$\overline{v_4} = (\overline{F_0} \cup \overline{F_1})\ \overline{y_4} \equiv \overline{A_1} \ldots \overline{A_n},$$

and $W \subseteq (\overline{F_0} \cup \overline{F_1}) \subseteq Y_0$.

We have shown (10.26), which implies that when we choose $W = Y_0$, we will get a step sequence v such that

$$\overline{v} = W_0 \overline{v_0} \equiv \overline{A_1} \ldots \overline{A_n} \tag{10.30}$$

where $Y_0 \subseteq W_0 \subseteq Y_0$. Since $Y_0 \subseteq W_0 \subseteq Y_0$ implies that $Y_0 = W_0$, from (10.30), we have v is the step sequence such that $\overline{v} = Y_0 \overline{v_0} \equiv \overline{A_1} \ldots \overline{A_n}$. Thus, $v = B_0 v_0 \equiv A_1 \ldots A_n$ But since $B_0 <^{st} A_1$, this contradicts the fact that $A_1 \ldots A_n$ is the least element of $[s]$ with respect to $<^{lex}$. Hence, we have shown that A_1 is the least element of $\{l(Y) \mid Y \in Z(\Sigma)\}$ with respect to $<^{st}$.

We now prove that A_i is the least element of $\{l[Y] \mid Y \in Z(\Sigma \setminus \biguplus(\overline{A_1} \ldots \overline{A_{i-1}}))\}$ with respect to $<^{st}$ by using induction on n, the number of steps of $A_1 \ldots A_n$. If $n = 0$, we are done. If $n > 0$, then we have just shown that A_1 is the least element of $\{l[Y] \mid Y \in Z(\Sigma)\}$ with respect to $<^{st}$. By applying the induction hypothesis on $p = \overline{A_2} \ldots \overline{A_n}$, $\Sigma_p = \Sigma \backslash \overline{A_1}$, and its stratified order structure $(\Sigma_p, \diamond |_{\Sigma_p \times \Sigma_p}, \sqsubset |_{\Sigma_p \times \Sigma_p})$, we get A_i is the least element of $\{l[Y] \mid Y \in Z(\Sigma \setminus \biguplus(\overline{A_1} \ldots \overline{A_{i-1}}))\}$ with respect to $<^{st}$ for all $i \geq 2$. Thus, we conclude A_i is the least element of $\{l[Y] \mid Y \in Z(\Sigma \setminus \biguplus(\overline{A_1} \ldots \overline{A_{i-1}}))\}$ with respect to $<^{st}$ for $1 \leq i \leq n$.

□

Theorem 10.2. *Let s, t be step sequences over a generalised comtrace alphabet $\theta = (E, sim, ser, inl)$. Then $s \equiv t$ if and only if $\xi_s = \xi_t$.*

Proof. ($\Rightarrow$) If $s \equiv t$, then $[s] = [t]$. Hence, by (10.17),

$$\xi_s = \left(\Sigma_s, \bigcap_{u \in [s]} \lhd_u^{\leftrightarrows}, \bigcap_{u \in [s]} \lhd_u^{\frown}\right) = \left(\Sigma_s, \bigcap_{u \in [t]} \lhd_u^{\leftrightarrows}, \bigcap_{u \in [t]} \lhd_u^{\frown}\right) = \xi_t.$$

($\Leftarrow$) By Lemma 10.1, we can use ξ_s to construct a unique element w_1 such that w_1 is the least element of both $[s]$ with respect to $<^{lex}$, and then use ξ_t to construct a unique element w_2 that is the least element of $[t]$ with respect to $<^{lex}$. But since $\xi_s = \xi_t$ and the construction is unique, we get $w_1 = w_2$. Hence, $s \equiv t$. □

By Theorem 10.2, for each step sequence s over a generalised comtrace alphabet $\theta = (E, sim, ser, inl)$, we will define the generalised stratified order structure induced by the generalised comtrace $[s]$ to be ξ_s.

To end this section, we prove two major results. Theorem 10.3 says that the stratified order extensions of the generalised stratified order structure induced by a generalised comtrace $[t]$ are exactly those generated by the step sequences in $[t]$. Theorem 10.4 says that the stratified order structure induced by a comtrace is uniquely identified by any of its extensions.

Lemma 10.2. *Let s, t be step sequences over a generalised comtrace alphabet $\theta = (E, sim, ser, inl)$ and $\lhd_s \in ext(\xi_t)$. Then $\xi_s = \xi_t$.*

Proof. Let $\xi_t = (\Sigma, <\!\!>, \sqsubset)$, $\xi_s = (\Sigma, <\!\!>', \sqsubset')$, $\prec = <\!\!> \cap \sqsubset$ and $\prec' = <\!\!>' \cap \sqsubset'$. We first want to show that $\xi_t \subseteq \xi_s$.

($<\!\!>_t = <\!\!>_s$) We have $\alpha <\!\!>_t \beta$ if and only if by Definition 10.2 $(l(\alpha), l(\beta)) \in inl$, which by Definition 10.2 means $\alpha <\!\!>_s \beta$. Hence,

$$<\!\!>_t = <\!\!>_s \tag{10.31}$$

($\sqsubset_t = \sqsubset_s$) If $\alpha \sqsubset_t \beta$, then by Definitions 10.1 and 10.2, $\alpha \sqsubset \beta$. But since $\lhd_s \in ext(\xi_t)$, we have $\alpha \lhd_s^\frown \beta$, which implies

$$pos_s(\alpha) \leq pos_s(\beta) \tag{10.32}$$

Since $\alpha \sqsubset_t \beta$, by Definition 10.2,

$$(l(\beta), l(\alpha)) \notin ser \cup inl \tag{10.33}$$

Hence, it follows from (10.32) and Definition 10.2 that $\alpha \sqsubset_s \beta$. Thus,

$$\sqsubset_t \subseteq \sqsubset_s \tag{10.34}$$

It remains to show that $\sqsubset_s \subseteq \sqsubset_t$. Let $\alpha \sqsubset_s \beta$, and we suppose for a contradiction that $\neg(\alpha \sqsubset_t \beta)$. Since $\alpha \sqsubset_s \beta$, by Definition 10.2, $pos_s(\alpha) \leq pos_s(\beta)$ and $(l(\beta), l(\alpha)) \notin ser \cup inl$. Since we assume $\neg(\alpha \sqsubset_t \beta)$, by Definition 10.2, we must have $pos_t(\beta) < pos_t(\alpha)$. But this by Definitions 10.1 and 10.2 implies that $\beta \prec_t \alpha$ and $\beta \prec \alpha$. But since $\lhd_s \in ext(\xi_t)$, we have $\beta \lhd_s \alpha$, which implies $pos_s(\beta) < pos_s(\alpha)$, a contradiction. Hence, $\sqsubset_s \subseteq \sqsubset_t$. Thus together with (10.34), we get

$$\sqsubset_t = \sqsubset_s \tag{10.35}$$

($\prec_t = \prec_s$) If $\alpha \prec_t \beta$, then by Definitions 10.1 and 10.2, $\alpha \prec \beta$. But since $\lhd_s \in ext(\xi_t)$, we have $\alpha \lhd_s \beta$, which implies

$$pos_s(\alpha) < pos_s(\beta) \tag{10.36}$$

Since $\alpha \prec_t \beta$, by Definition 10.2,

$$\left(\begin{array}{ll} & (l(\alpha), l(\beta)) \notin ser \cup inl \\ \vee & (\alpha, \beta) \in <\!\!>_t \cap \left(si(\sqsubset_t^*) \circ <\!\!>_t^{\mathbf{C}} \circ si(\sqsubset_t^*)\right) \\ \vee & \left(\begin{array}{ll} & (l(\alpha), l(\beta)) \in ser \\ \wedge & \exists \delta, \gamma \in \Sigma_t. \left(\begin{array}{ll} & pos_t(\delta) < pos_t(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser \\ \wedge & \alpha \sqsubset_t^* \delta \sqsubset_t^* \beta \wedge \alpha \sqsubset_t^* \gamma \sqsubset_t^* \beta \end{array} \right) \end{array} \right) \end{array} \right).$$

We want to show that $\alpha \prec_s \beta$.

- When $(l(\alpha), l(\beta)) \notin ser \cup inl$, it follows from (10.36) and Definition 10.2 that $\alpha \prec_s \beta$.

- When $(\alpha, \beta) \in <\!\!>_t \cap \left(si(\sqsubset_t^*) \circ <\!\!>_t^{\mathbf{C}} \circ si(\sqsubset_t^*)\right)$, then $\alpha <\!\!>_t \beta$ and there are $\delta, \gamma \in \Sigma$ such that $\alpha\ si(\sqsubset_t^*)\ \delta <\!\!>_t^{\mathbf{C}} \gamma\ si(\sqsubset_t^*)\ \beta$. Since $\sqsubset_t = \sqsubset_s$ and $<\!\!>_t = <\!\!>_s$, we also have $\alpha <\!\!>_s \beta$ and $\alpha\ si(\sqsubset_s^*)\ \delta <\!\!>_s^{\mathbf{C}} \gamma\ si(\sqsubset_s^*)\ \beta$. Thus, it follows from (10.36) and Definition 10.2 that $\alpha \prec_s \beta$.

- There remains only the case when $(l(\alpha), l(\beta)) \in ser$ and there are $\delta, \gamma \in \Sigma_t$ such that
$$\left(\begin{array}{ll} & pos_t(\delta) < pos_t(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser \\ \wedge & \alpha \sqsubset_t^* \delta \sqsubset_t^* \beta \wedge \alpha \sqsubset_t^* \gamma \sqsubset_t^* \beta \end{array} \right).$$
Since $\sqsubset_t = \sqsubset_s$, we also have $\alpha \sqsubset_s^* \delta \sqsubset_s^* \beta \wedge \alpha \sqsubset_s^* \gamma \sqsubset_s^* \beta$. Since $(l(\delta), l(\gamma)) \notin ser$, we either have $(l(\delta), l(\gamma)) \in inl$ or $(l(\delta), l(\gamma)) \notin ser \cup inl$.

- If $(l(\delta), l(\gamma)) \in inl$, then $pos_s(\delta) \neq pos_s(\gamma)$. This implies $(pos_s(\delta) < pos_s(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser)$ or $(pos_s(\gamma) < pos_s(\delta) \wedge (l(\gamma), l(\delta)) \notin ser)$. So it follows from (10.36) and Definition 10.2 that $\alpha \prec_s \beta$.
- If $(l(\delta), l(\gamma)) \notin inl$, then $(l(\delta), l(\gamma)) \notin ser \cup inl$. Hence, by Definition 10.2, $\delta \prec_t \gamma$, which by Definitions 10.1 and 10.2, $\delta \prec \gamma$. But since $\lhd_s \in ext(\xi_t)$, we have $\delta \lhd_s \gamma$, which implies $pos_s(\delta) < pos_s(\gamma)$. Since $pos_s(\delta) < pos_s(\gamma)$ and $(l(\delta), l(\gamma)) \notin ser$, it follows from (10.36) and Definition 10.2 that $\alpha \prec_s \beta$.

Thus, we have shown that $\alpha \prec_s \beta$. Hence,

$$\prec_t \subseteq \prec_s \tag{10.37}$$

It remains to show that $\prec_s \subseteq \prec_t$. Let $\alpha \prec_s \beta$, and we suppose for a contradiction that $\neg(\alpha \prec_t \beta)$. Since $\alpha \prec_s \beta$, by Definition 10.2, we have $pos_s(\alpha) < pos_s(\beta)$ and

$$\left(\begin{array}{ll} & (l(\alpha), l(\beta)) \notin ser \cup inl \\ \vee & (\alpha, \beta) \in \mathrel{<\!\!>}_s \cap \left(si(\sqsubset_s^*) \circ \mathrel{<\!\!>}_s^{\mathbf{C}} \circ si(\sqsubset_s^*) \right) \\ \vee & \left(\begin{array}{ll} & (l(\alpha), l(\beta)) \in ser \\ \wedge & \exists \delta, \gamma \in \Sigma_s. \left(\begin{array}{ll} & pos_s(\delta) < pos_s(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser \\ \wedge & \alpha \sqsubset_s^* \delta \sqsubset_s^* \beta \wedge \alpha \sqsubset_s^* \gamma \sqsubset_s^* \beta \end{array} \right) \end{array} \right) \end{array} \right).$$

We want to show that $\alpha \prec_t \beta$.

- When $(l(\alpha), l(\beta)) \notin ser \cup inl$, we suppose for a contradiction that $\neg(\alpha \prec_t \beta)$. This by Definition 10.2 implies that $pos_t(\beta) \leq pos_t(\alpha)$. By Definitions 10.1 and 10.2, it follows that $\beta \sqsubset_t \alpha$ and $\beta \sqsubset \alpha$. But since $\lhd_s \in ext(\xi_t)$, we have $\beta \lhd_s^{\frown} \alpha$, which implies $pos_s(\beta) \leq pos_s(\alpha)$, a contradiction.
- If $(\alpha, \beta) \in \mathrel{<\!\!>}_s \cap \left(si(\sqsubset_s^*) \circ \mathrel{<\!\!>}_s^{\mathbf{C}} \circ si(\sqsubset_s^*) \right)$, then since $\mathrel{<\!\!>}_s = \mathrel{<\!\!>}_t$ and $\sqsubset_s = \sqsubset_t$, we have $(\alpha, \beta) \in \mathrel{<\!\!>}_t \cap \left(si(\sqsubset_t^*) \circ \mathrel{<\!\!>}_t^{\mathbf{C}} \circ si(\sqsubset_t^*) \right)$. Since $\alpha \mathrel{<\!\!>}_t \beta$, we have $pos_t(\alpha) < pos_t(\beta)$ or $pos_t(\beta) < pos_t(\alpha)$. We want to show that $pos_t(\alpha) < pos_t(\beta)$. Suppose for a contradiction that $pos_t(\beta) < pos_t(\alpha)$. But since $(\alpha, \beta) \in \mathrel{<\!\!>}_t \cap \left(si(\sqsubset_t^*) \circ \mathrel{<\!\!>}_t^{\mathbf{C}} \circ si(\sqsubset_t^*) \right)$ and $\mathrel{<\!\!>}_t$ is symmetric, we have $(\beta, \alpha) \in \mathrel{<\!\!>}_t \cap \left(si(\sqsubset_t^*) \circ \mathrel{<\!\!>}_t^{\mathbf{C}} \circ si(\sqsubset_t^*) \right)$. Hence, it follows from Definitions 10.1 and 10.2 that $\beta \prec_t \alpha$ and $\beta \prec \alpha$. But since $\lhd_s \in ext(\xi_t)$, we have $\beta \lhd_s \alpha$, which implies

$pos_s(\beta) < pos_s(\alpha)$, a contradiction. We have just shown that $pos_t(\alpha) < pos_t(\beta)$. Since $(\alpha, \beta) \in \diamond_t \cap \left(si(\sqsubset_t^*) \circ \diamond_t^{\mathbf{C}} \circ si(\sqsubset_t^*)\right)$, we get $\alpha \prec_t \beta$

- There remains only the case when $(l(\alpha), l(\beta)) \in ser$ and there are $\delta, \gamma \in \Sigma_s$ such that
$$\left(\begin{array}{ll} & pos_s(\delta) < pos_s(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser \\ \wedge & \alpha \sqsubset_s^* \delta \sqsubset_s^* \beta \wedge \alpha \sqsubset_s^* \gamma \sqsubset_s^* \beta \end{array}\right).$$
Since $\sqsubset_s = \sqsubset_t$, we have $\alpha \sqsubset_t^* \delta \sqsubset_t^* \beta$ and $\alpha \sqsubset_t^* \gamma \sqsubset_t^* \beta$, which by Definition 10.2 and transitivity of $\leq$ implies that $pos_t(\alpha) \leq pos_t(\delta) \leq pos_t(\beta)$ and $pos_t(\alpha) \leq pos_t(\gamma) \leq pos_t(\beta)$. Since $(l(\delta), l(\gamma)) \notin ser$, we either have $(l(\delta), l(\gamma)) \in inl$ or $(l(\delta), l(\gamma)) \notin ser \cup inl$.

    - If $(l(\delta), l(\gamma)) \in inl$, then $pos_t(\delta) \neq pos_t(\gamma)$. This implies $(pos_t(\delta) < pos_t(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser)$ or $(pos_t(\gamma) < pos_t(\delta) \wedge (l(\gamma), l(\delta)) \notin ser)$. Since $pos_t(\delta) \neq pos_t(\gamma)$ and $pos_t(\alpha) \leq pos_t(\delta) \leq pos_t(\beta)$ and $pos_t(\alpha) \leq pos_t(\gamma) \leq pos_t(\beta)$, we also have $pos_t(\alpha) < pos_t(\beta)$. So it follows from Definition 10.2 that $\alpha \prec_t \beta$.
    - If $(l(\delta), l(\gamma)) \notin inl$, then $(l(\delta), l(\gamma)) \notin ser \cup inl$. We want to show that $pos_t(\delta) < pos_t(\gamma)$. Suppose for a contradiction that $pos_s(\delta) \geq pos_s(\gamma)$, then since $(l(\delta), l(\gamma)) \notin ser \cup inl$, by Definitions 10.1 and 10.2, we have $\gamma \sqsubset_t \delta$ and $\gamma \sqsubset \delta$. But since $\lhd_s \in ext(\xi_t)$, we have $\gamma \lhd_s^{\frown} \delta$, which implies $pos_s(\gamma) \leq pos_s(\delta)$, a contradiction. Since $pos_t(\delta) < pos_t(\gamma)$ and $pos_t(\alpha) \leq pos_t(\delta) \leq pos_t(\beta)$ and $pos_t(\alpha) \leq pos_t(\gamma) \leq pos_t(\beta)$, we have $pos_t(\alpha) < pos_t(\beta)$. Hence, we have $pos_t(\alpha) < pos_t(\beta)$ and
    $$\left(\begin{array}{ll} & pos_t(\delta) < pos_t(\gamma) \wedge (l(\delta), l(\gamma)) \notin ser \cup inl \\ \wedge & \alpha \sqsubset_t^* \delta \sqsubset_t^* \beta \wedge \alpha \sqsubset_t^* \gamma \sqsubset_t^* \beta \end{array}\right).$$
    Thus, it follows from Definition 10.2 that $\alpha \prec_t \beta$.

Thus, we have shown that $\alpha \prec_t \beta$, which implies $\sqsubset_t \subseteq \sqsubset_s$. Hence, by (10.37),

$$\prec_t = \prec_s \tag{10.38}$$

From (10.31), (10.35) and (10.38), we have

$$(\Sigma, \diamond_t \cup \prec_t, \sqsubset_t \cup \prec_t) = (\Sigma, \diamond_s \cup \prec_s, \sqsubset_s \cup \prec_s).$$

Thus, we conclude

$$\xi_t = (\Sigma, \Diamond_t \cup \prec_t, \sqsubset_t \cup \prec_t)^{\bowtie} = (\Sigma, \Diamond_s \cup \prec_s, \sqsubset_s \cup \prec_s)^{\bowtie} = \xi_s.$$

□

Theorem 10.3. *Let t be a step sequence over a generalised comtrace alphabet (E, sim, ser, inl). Then $ext(\xi_t) = \{\lhd_u \mid u \in [t]\}$.*

Proof. ($\subseteq$) Suppose $\lhd \in ext(\xi_t)$. By Proposition 10.19, there is a step sequence u such that $\lhd_u = \lhd$. Hence, by Lemma 10.2, we have $\xi_u = \xi_t$, which by Theorem 10.2 implies that $u \equiv t$. Hence, $ext(\xi_t) \supseteq \{\lhd_u \mid u \in [t]\}$.

($\supseteq$) If $u \in [t]$, then it follows from Theorem 10.2 that $\xi_u = \xi_t$. This and Proposition 10.18 imply $\lhd_u \in ext(\xi_t)$. Hence, $ext(\xi_t) \supseteq \{\lhd_u \mid u \in [t]\}$. □

Theorem 10.4. *Let s and t be step sequences over a generalised comtrace alphabet (E, sim, ser, inl) such that $ext(\xi_s) \cap ext(\xi_t) \neq \emptyset$. Then $s \equiv t$.*

Proof. Let $\lhd \in ext(\xi_s) \cap ext(\xi_t)$. By Proposition 10.19, there is a step sequence u such that $\lhd_u = \lhd$. By Lemma 10.2, we have $\xi_s = \xi_u = \xi_t$. This and Theorem 10.2 yields $s \equiv t$. □

Chapter 11

Conclusion and Future Works

The concepts of absorbing monoids over step sequences, partially commutative absorbing monoids over step sequences, absorbing monoids with compound generators, monoids of generalised comtraces and their canonical representations have been introduced and analysed. All of these quotient monoids are the generalisations of Mazurkiewicz trace and comtrace monoids. We have shown some algebraic and formal language properties of comtraces, and provided a new version of the proof of the existence of a unique canonical representation for comtraces. We then prove Theorem 9.5, which states that any finite stratified order structure can be represented by a comtrace.

One interesting observation is that the notions of non-serialisable steps are convenient for capturing the weak causality relationship induced not only by a comtrace but also by a generalised comtrace. The uses of non-serialisable steps for generalised comtraces were shown in Proposition 10.15, which was absolutely required for our proof of Theorem 10.1.

It is worth noticing that Theorems 9.3 and 10.3 can be seen as the generalisations of the Szpilrajn Theorem in the context of comtraces and generalised comtraces respectively. In other words, the (generalised) stratified order structure induced by a (generalised) comtrace $[t]$ can be uniquely reconstructed from the stratified orders generated by the step sequences in $[t]$.

Despite some obvious advantages, for instance very handy composition and no need to use labels, quotient monoids (perhaps with some exception of Mazurkiewicz traces) are much less popular for analysing issues of concurrency than their relational counterparts as 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.

An immediate task is to prove the analogue of Theorem 9.5 for generalised comtraces which says that each generalised stratified order structure can be represented by a generalised comtrace. This should not be difficult, thanks to the results from Chapter 10 and the analogy to the proof of Theorem 9.5.

Another interesting task is to study our novel notion of absorbing monoids with compound generators, which can model asymmetric synchrony. We believe the concept of compound generators might relate to another line of our research on the theory of part-whole relations in [22] which utilises the ideas from both *mereology* [29] and *category theory* [23, 6].

Much harder future tasks are in the area of comtrace and generalised comtrace languages with such major problems as recognisability [26], where the equivalences of Zielonka's Theorem[1] [33] for comtraces and generalised comtraces, etc., are still open.

[1] Zielonka's Theorem states that a trace language is recognisable if and only if it is accepted by some *finite asynchronous automaton.*

Bibliography

[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] B.A. Davey and H. A. Priestley, *Introduction to Lattices and Order.* Cambridge University Press 2002.

[4] 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.

[5] J. Desel, Private Information, Communicated to Ryszard Janicki by G. Juhás, 2007.

[6] J. L. Fiadeiro. *Categories for Software Engineering.* Springer, 2004.

[7] V. Diekert, *Combinatorics on Traces*, Springer 1990.

[8] V. Diekert and G. Rozenberg (eds.), *The Book of Traces.* World Scientific 1995.

[9] H. Gaifman and V. Pratt, Partial Order Models of Concurrency and the Computation of Function, *Proc. of LICS'87*, pp. 72-85.

[10] G. Guo and R. Janicki, Modelling Concurrent Behaviours by Commutativity and Weak Causality Relations, Proc. of AMAST'02, *Lecture Notes in Computer Science* 2422 (2002), 178-191.

[11] R. Janicki. Relational Structures Model of Concurrency. *Acta Informatica*, 45(4): 279-320, 2008.

[12] R. Janicki and M. Koutny, Invariants and Paradigms of Concurrency Theory, *Lecture Notes in Computer Science* 506, Springer 1991, pp. 59-74.

[13] R. Janicki and M. Koutny, Structure of Concurrency, *Theoretical Compututer Science*, 112(1):5–52, 1993.

[14] R. Janicki and M. Koutny, Semantics of Inhibitor Nets, *Information and Computation*, 123(1):1–16, 1995.

[15] R. Janicki and M. Koutny, Fundamentals of Modelling Concurrency Using Discrete Relational Structures, *Acta Informatica*, 34:367–388, 1997.

[16] R. Janicki and M. Koutny, On Causality Semantics of Nets with Priorities, *Fundamenta Informaticae* 34:222–255, 1999.

[17] R. Janicki and D. T. M. Lê, Modelling Concurrency with Quotient Monoids, Proc of PETRI NETS 2008, *Lecture Notes in Computer Science* 5062, Springer 2008, pp. 251–269.

[18] 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.

[19] G. Juhás, R. Lorenz, C. Neumair, Synthesis of Controlled Behavious with Modules of Signal Nets, *Lecture Notes in Comp. Science* 3099, Springer 2004, pp. 233-257.

[20] H. C. M. Kleijn and M. Koutny, Process Semantics of General Inhibitor Nets, *Information and Computation*, 190:18–69, 2004.

[21] K. Kunen. *Set Theory: An Introduction to Independence Proofs.* Elsevier Science Ltd, 1980.

[22] D. T. M. Lê and R. Janicki, A Categorical Approach to Mereology and Its Application to Modelling Software Components, *Transactions on Rough Sets* VIII, LNCS 5084, pp. 146–174, 2008.

[23] S. MacLane. *Categories for the Working Mathematician.* Springer, 1998.

[24] A. Mazurkiewicz, Concurrent Program Schemes and Their Interpretation, TR DAIMI PB-78, Comp. Science Depart., Aarhus University, 1977.

[25] A. Mazurkiewicz, Introduction to Trace Theory, in [8], pp. 3–42.

[26] E. Ochmański, Recognizable Trace Languages, in [8], pp. 167-204.

[27] M. Pietkiewicz-Koutny, The Synthesis Problem for Elementary Net Systems, *Fundamenta Informaticae* 40(2,3):310–327, 1999.

[28] M. W. Shields, Adequate Path Expressions, *Lecture Notes in Computer Science* 70, Springer 1979, pp. 249-265.

[29] P. Simons. *Parts: A Study in Ontology.* Oxford University Press, Oxford, 1987.

[30] P. Suppes, *Axiomatic Set Theory*, Dover, 1972.

[31] E. Szpilrajn, Sur l'extension de l'ordre partiel, *Fundamenta Mathematicae* 16 (1930), 386-389.

[32] J. D. Weston, A short proof of Zorn's lemma. *Arch. Math.*, 8:279, 1957.

[33] W. Zielonka, Notes on Finite Asynchronous Automata, *RAIRO Inform. Théor. Appl.* 21:99–135, 1987.