Report Number: WUCS-88-20

1988-06-01

The Mathematics of Directed Specifications

Jan Tijmen Udding and Tom Verhoeff

In this paper we lay a mathematical foundation for processes that communicate via directed communication channels. We start from a collection of primitive specifications. Particular correctness concerns partition this collection into equivalence classes, which can serve as abstract specifications. The theory is illustrated by taking as correctness concern absence of computation interference. In this case the abstract specification space can be identified with the space of delay-insensitive specifications.

THE MATHEMATICS OF DIRECTED SPECIFICATIONS

Jan Tijmen Udding and Tom Verhoeff

WUCS-88-20

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

The Mathematics of Directed Specifications

Jan Tijmen Udding

Department of Computer Science
Washington University
Campus Box 1045
St. Louis, MO 63130

*Tom Verhoeff**

Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands

June 1988

Abstract

In this paper we lay a mathematical foundation for processes that communicate via directed communication channels. We start from a collection of primitive specifications. Particular correctness concerns partition this collection into equivalence classes, which can serve as abstract specifications. The theory is illustrated by taking as correctness concern absence of computation interference. In this case the abstract specification space can be identified with the space of delay-insensitive specifications.

*Currently on leave of absence at Department of Computer Science, Washington University, Campus Box 1045, St. Louis, MO 63130.

Contents

0 Introduction

0.0 Research Context

The research reported in this paper springs from an interest in the design of mechanisms that communicate with their environments. Ideally, design starts with some form of initial specification and works its way towards a final implementation that satisfies the specification in a certain sense. We shall be concerned with the logical part of the design trajectory. In this case, the implementations we strive for are abstract networks of communicating processes that operate concurrently. The translation into physical circuitry is not addressed.

There is no general agreement on the semantics for systems with concurrency. The formalism CSP [1], for instance, is based on a symmetric (or undirected) synchronous communication primitive. The participants in a communication action all complete it at the same step (synchrony) and none will proceed until all partners are ready to engage in it (symmetry). Physics, however, does not provide phenomena to implement this type of communication directly on the circuit level. Therefore, it is more natural to consider asymmetric (or directed) asynchronous communication primitives. In this case, for each communication taking place there is an initiating party, which sends a signal to other parties regardless of their readiness (asymmetry). Each party is free to complete its part of the communication action before the others do so (asynchrony). This type of communication corresponds more closely to how physical systems influence each other's behavior. It is the designer's responsibility, however, to guarantee that the receiving ends are ready to process communications initiated by others.

The theory of directed specifications that we develop in this paper is mathematical in the usual ways. It is formal in that it tries to isolate, and where possible eliminate, intuition and empirical aspects. We are aware of the fact that the link with physics needs attention, but we shall ignore that issue here. It is abstract in that it transcends concrete applications, such as electrical communications in VLSI circuits. Nevertheless, it may be helpful to keep a particular interpretation in mind. It is concerned with foundational security in that we start with the very basics and carefully built up from there. Furthermore, the development of our theory for directed specifications uses the method of structural enrichment and subsequent abstraction from equivalence, which is often encountered in mathematics.

To illustrate this method we outline the key steps in the development of

number systems in mathematics. There, one starts, for example, with the set of natural numbers and the addition operator. Then it is noticed that certain "numbers" (viz. negative) are missing. Therefore, a new domain is created that is sufficiently rich to contain the objects with the desired properties, say, the set of pairs of natural numbers, representing their difference. Also a new addition operator is defined for this enriched domain. It turns out, however, that the new domain is too large and that certain distinct objects are equivalent for the intended purposes. The extended set of "numbers" is now obtained by abstracting from the equivalence. For instance, the set of integers is obtained as the set of equivalence classes of pairs of natural numbers. It is often interesting to find sets and operators that are constructed differently but that are isomorphic to the new domain to gain more insight in its structure. The cycle repeats itself from here, since the new number system may serve as the basis for further extensions along the same lines.

Specification domains can be constructed in a similar way. We start with a rich set of implementation-like objects obtained from primitive specifications. Then we define an equivalence based on some correctness concern, for instance, absence of computation interference. This resembles testing equivalence of [0]. The new abstract specification domain consists of the resulting equivalence classes. The cycle is repeated by incorporating other correctness concerns that call for more refined specifications. A similar hierarchical development of specification domains can also be found in [2].

We shall not be concerned with engineering applications in this paper. Our conviction is that we need a good foundation first. We should be careful not to introduce the analogon of the Roman numerals for engineering purposes.

0.1 Overview

This paper does not present a complete theory. Its intention is to exemplify the general paradigm. In section 1 we define two games. They capture the operational semantics for our primitive specifications. In both games the communications are directed. One game deals with synchronous communication, the other with asynchronous communication. Section 2 analyzes the two-person version of the synchronous game. It gives rise to an important partial order on directed specifications. Sections 3 proves some properties of this partial order when applied to the delay-insensitive specifications of [3]. Two isomorphic domains of directed specifications are presented in Section 4.

The two-person version of the asynchronous game is analyzed in Section 5. It shows that delay-insensitive specifications can be obtained as equivalence classes of directed specifications induced by absence of computation interference. Section 6 makes a first attempt at introducing a parallel composition operator for our specifications. Finally, in Section 7 we look back at what we have accomplished and what needs to be done in the future.

1 The Games

In this section we define two games that play a key role throughout this paper. One is called the synchronous game and the other one is called the asynchronous game. Both are modifications of the game underlying the operational semantics of CSP [1]. We start out with a number of notions and definitions.

Throughout this section we assume Ω to be a set of symbols. An *alphabet* is a subset of Ω. For alphabet A, A^* denotes the set of all finite-length sequences over symbols in A. A *trace* is an element of Ω^* and a *trace set* is a set of traces. Concatenation of traces is denoted by juxtaposition and the empty trace is denoted by ε.

Definition 0 Trace s is said to be a prefix of trace t, denoted by $s \leq t$, when $(\exists\, u :: su = t)$. Trace set S is said to be prefix-closed when

$$(\forall\, s,t : s \leq t \wedge t \in S : s \in S).$$

□

Definition 1 Projection, denoted by $\lceil$, of a trace on alphabet A is defined by

$$\begin{array}{lcll} \varepsilon\lceil A & = & \varepsilon, \text{ and for trace } s \text{ and symbol } a, & \\ (sa)\lceil A & = & s\lceil A & \text{if } a \notin A, \\ (sa)\lceil A & = & (s\lceil A)a & \text{if } a \in A. \end{array}$$

For trace set S and alphabet A, $S\lceil A$ is defined as $\{s : s \in S : s\lceil A\}$. □

Definition 2 The length of trace t, denoted by $\ell(t)$, is the number of symbols in t. The number of symbols a in trace t, denoted by $\#_a t$, is $\ell(t\lceil\{a\})$. □

Definition 3 A directed trace structure, or *DTS*, is a triple consisting of an input alphabet, an output alphabet and a prefix-closed trace set. For *DTS* S, the input alphabet is denoted by $\mathbf{i}S$, the output alphabet by $\mathbf{o}S$, and the trace set by $\mathbf{t}S$. The input and the output alphabets are disjoint subsets of Ω, and $\mathbf{t}S \subseteq (\mathbf{i}S \cup \mathbf{o}S)^*$. The alphabet of S is $\mathbf{i}S \cup \mathbf{o}S$ and is also denoted by $\mathbf{a}S$. □

Note 0 Not requiring the trace set of a *DTS* to be non-empty seems awkward at first. In the section on composition the crucial role of the empty *DTS* will become clear. □

Rather than saying that a trace belongs to the trace set of a *DTS* we usually say that it belongs to that *DTS*. In the same vein, we usually say that symbol a is an input or of type input in *DTS* S when we mean $a \in \mathbf{i}S$. Also, by the input alphabet of a collection of *DTS*'s we mean the union of the input alphabets of its elements.

Definition 4 The reflection of *DTS* S, denoted by $\tilde{S}$, is the trace structure T with

$$\mathbf{i}T = \mathbf{o}S,\ \mathbf{o}T = \mathbf{i}S, \text{ and } \mathbf{t}T = \mathbf{t}S.$$

□

Definition 5 The weave of a collection X of *DTS*'s, denoted by $\mathbf{w}X$, is the trace set

$$\{s : s \in (\bigcup S : S \in X : \mathbf{a}S)^* \wedge (\forall S : S \in X : s\lceil \mathbf{a}S \in S) : s\}.$$

□

Definition 6 A collection X of *DTS*'s is said to be closed, denoted by $\mathbf{cl}X$, when

$$(\bigcup S : S \in X : \mathbf{i}S) = (\bigcup S : S \in X : \mathbf{o}S) \wedge$$
$$(\forall S,T : S,T \in X \wedge S \neq T : \mathbf{i}S \cap \mathbf{i}T = \mathbf{o}S \cap \mathbf{o}T = \emptyset).$$

□

Every symbol in a closed collection of *DTS*'s occurs once as an output symbol and once as an input symbol of a *DTS*. The definition of a closed collection could have been more general so as to allow *DTS*'s to have common input symbols. We decided not to do so for reasons of clarity of exposition. For example, if X is a collection of *DTS*'s then all U's with the property that $\mathbf{cl}(X \cup \{U\})$ holds have equal input alphabets and equal output alphabets.

1.0 The Synchronous Game

Given is a closed set X of DTS's. The synchronous game is played on a game vector v of traces, one trace for each member S of X, denoted by $v.S$. A *move* consists of choosing a DTS in X, say S, and a symbol, say a, such that $a \in \mathbf{o}S$ and $(v.S)a \in S$ and replacing those traces $v.T$ by $(v.T)a$ for which $a \in \mathbf{a}T$. One move changes two traces in the game vector. The game starts with the vector that has ε in all of its components. A game vector is called *reachable* when it can be reached by a sequence of moves from the initial game vector.

When playing the game we start at the initial vector and move from one reachable game vector to another. Each move extends two traces with a certain symbol. We can record the execution of a particular game, by writing down from left to right and starting with the empty trace, those symbols that the traces of the game vector are subsequently extended with. We define the set of *game traces* as the set of the finite-length traces that can be brought about in this way.

The weave of a closed collection X of DTS's is related to the set of all reachable game vectors in the following way. For any trace t in $\mathbf{w}X$ the game vector v defined by $v.S = t\lceil\mathbf{a}S$ for all $S \in X$ is reachable. Moreover, the set of game traces contains $\mathbf{w}X$ as a subset.

1.1 The Asynchronous Game

The only difference between the synchronous and the asynchronous game is the definition of a move. In the asynchronous game each move changes one trace in the game vector. A *move* is either an output move or an input move. An *output move* consists of choosing a DTS in X, say S, and a symbol, say a, such that $a \in \mathbf{o}S$ and $(v.S)a \in S$ and replacing $v.S$ by $(v.S)a$ in the game vector. An *input move* consists of choosing a DTS in X, say S, and a symbol, say a, such that $(\exists T : T \in X : \#_a v.T > \#_a v.S)$, and replacing $v.S$ by $(v.S)a$ in the game vector. Notice that $a \in \mathbf{i}S$ when this condition holds. Also, if a game vector is reachable in the synchronous game then it is reachable in the asynchronous game by twice the number of moves.

1.2 The Correctness Concern

We want to define correctness criteria for the execution of a game on the union of two collections of DTS's X and U. One collection is generally interpreted as a set of modules and the other collection is viewed as the

environments of these modules. For this pair of collections we want a certain correctness predicate $P(X,U)$ to hold. This induces the **sat**-relation as follows. Specification X satisfies specification Y, or is at least as good as Y, when $P(X,U)$ holds whenever $P(Y,U)$ holds for all collections of environments U.

Definition 7 For correctness concern P and for collections X and Y of DTS's we say that X satisfies Y, denoted by X **sat** Y, when

$$(\forall U : P(Y,U) : P(X,U)).$$

□

Property 0 The relation **sat** is a pre-order, that is, it is reflexive and transitive. □

Generally, $P(X,U)$ is the conjunction of a number of predicates. One of these predicates will always be that $X \cup U$ is a closed collection, so that modules and environments "hook up" to one another correctly. Throughout the rest of the paper we choose as our additional correctness concern absence of computation interference, which is defined in the following way.

Definition 8 A closed collection X is said to have absence of computation interference when all reachable game vectors are safe. A game vector v is safe when $(\forall S : S \in X : v.S \in S)$. We denote the predicate that X is closed and has absence of computation interference by **nsi**X when playing the synchronous game and by **nai**X when playing the asynchronous game. □

When a move can extend a trace in a reachable game vector with an input symbol while the resulting trace does not belong to its corresponding DTS we have computation interference. For a closed collection X of DTS's absence of computation interference in the synchronous game is equivalent to the set of game traces of X being equal to $\mathbf{w}X$. This would have been an alternative way to define absence of computation interference in this game. Notice that the game vectors obtained from $\mathbf{w}X$ by projection onto the alphabets of the individual trace structures are exactly the safe reachable game vectors.

Property 1 A closed collection of DTS's X has absence of computation interference in the synchronous game when the initial game vector is safe and when for all safe reachable game vectors v

$$(\forall S,T,a : S,T \in X \wedge a \in \mathbf{o}S \cap \mathbf{i}T : (v.S)a \in S \Rightarrow (v.T)a \in T)$$

or, equivalently, when the initial game vector is safe and when

$$\begin{array}{ll}(\forall S,T,s,a : & S,T \in X \wedge a \in \mathbf{o}S \cap \mathbf{i}T \wedge s \in \mathbf{w}X : \\ & (s\lceil \mathbf{a}S)a \in \S \Rightarrow (s\lceil \mathbf{a}T)a \in T).\end{array}$$

It has absence of computation interference in the asynchronous game when the initial game vector is safe and when for all safe reachable game vectors v

$$(\forall S,T,a : S,T \in X \wedge \#_a v.S > \#_a v.T : (v.T)a \in T).$$

□

We can confine ourselves in this property to the *safe* reachable game vectors since any unsafe reachable vector can only be reached by a transition from a safe to an unsafe vector. Notice that if a closed collection of DTS's has absence of computation interference in the asynchronous game then it has absence of computation interference in the synchronous game. Notice also that a collection containing an empty DTS has computation interference since the initial game vector is not safe. It is unclear yet whether it is more elegant to define computation interference so that the empty trace structure does not result in interference.

1.3 Equivalence

Given the **sat**-relation we can define an equivalence relation on the space of all collections of DTS's. We call two specifications equal when no environment can distinguish the two. This equivalence is called testing-equivalence in [0].

Definition 9 Two collections X and Y of DTS's are called equivalent, denoted by $X \,\mathbf{equ}\, Y$, when $X \,\mathbf{sat}\, Y \wedge Y \,\mathbf{sat}\, X$. □

Property 2 For collections X and Y we have

$$X \,\mathbf{equ}\, Y = (\forall U :: P(X,U) = P(Y,U)).$$

□

It is obvious that **equ** is an equivalence relation. In the sequel we study, in detail, the situation in which both the collection of modules and the collection of environments contain one element each. The resulting game

is also referred to as the two-person game. In a closed collection with two elements the input alphabet of the one is the output alphabet of the other one. We conclude with a few abbreviations that will be used in the next sections.

Definition 10 We write S **nai** T and S **nsi** T for **nai**$\{S,\tilde{T}\}$ and **nsi**$\{S,\tilde{T}\}$ when playing a two-person game on a collection consisting of S and $\tilde{T}$. □

Notice that S **nai** T and S **nsi** T are false whenever S and T have unequal input alphabets or unequal output alphabets.

Definition 11 The (S,T)-game is the game played on the collection $\{S,\tilde{T}\}$. A game vector v in the (S,T)-game is denoted by the pair of traces $(v.S, v.\tilde{T})$. □

Property 3 The set of reachable vectors in the (S,T)-game is equal to the set of reachable vectors in the $(\tilde{T},\tilde{S})$-game. □

2 Analysis of the Synchronous 2-Person Game

In this section we analyze the synchronous two-person game in detail. We consider a space $\mathcal{T}$ of non-empty prefix-closed trace structures with input alphabet I and output alphabet O, the union of which we call A. We show that the **sat**-relation and the **nsi**-relation are equal on this space and that they make $\mathcal{T}$ a complete lattice. The equivalence classes under **equ** are singletons.

Definition 12 The space $\tilde{\mathcal{T}}$ is the space of trace structures obtained by reflecting the trace structures of $\mathcal{T}$. □

Property 4 The two traces constituting a reachable game vector in a synchronous two-person game are equal. □

On account of the last property we view the game vector as one single trace rather than as a pair of (equal) traces. Notice that this single trace is also the game trace as defined in the previous section. From Property 1 we infer for a synchronous two-person game the following property, due to the fact that the initial game vector is safe for elements of $\mathcal{T}$.

Property 5 For $S,T \in \mathcal{T}$ we have

$$S \,\mathbf{nsi}\, T = (\forall v, a : \; v \in S \cap T : \\ ((a \in O \wedge va \in S) \Rightarrow va \in T) \wedge \\ ((a \in I \wedge va \in T) \Rightarrow va \in S)).$$

□

Property 6 For $S \in \mathcal{T}$ we have $S \,\mathbf{nsi}\, S$. □

Property 7 For $S, T \in \mathcal{T}$ we have in the synchronous game

$$S \,\mathbf{sat}\, T = (\forall U : T \,\mathbf{nsi}\, U : S \,\mathbf{nsi}\, U).$$

□

2.0 The partial order on $\mathcal{T}$

On $\mathcal{T}$ we define the following relation.

Definition 13 For $S, T \in \mathcal{T}$ and $B \subseteq \Omega$ we define

$$S \subseteq_B T = (\forall s, m : m \in B \wedge sm \in S \wedge s \in T : sm \in T).$$

□

Lemma 0 For $S, T \in \mathcal{T}$ and $B, C \subseteq A$ such that $B \cup C = A$, we have

$$S \subseteq_B T \wedge S \subseteq_C T \Rightarrow S \subseteq T.$$

Proof Since $B \cup C = A$, the left hand side of the implication reduces to $(\forall s, m : sm \in S \wedge s \in T : sm \in T)$. Using $\varepsilon \in T$, it is straightforward to prove $S \subseteq T$ by mathematical induction on the length of the traces in T. □

We define the following relations on $\mathcal{T}$ and $\tilde{\mathcal{T}}$.

Definition 14 For $S, T \in \mathcal{T}$

$$S \sqsubseteq T = S \subseteq_O T \wedge T \subseteq_I S$$

and for $S, T \in \tilde{\mathcal{T}}$

$$S \tilde{\sqsubseteq} T = S \subseteq_I T \wedge T \subseteq_O S.$$

□

Property 8 For $S,T \in \mathcal{T}$ we have $S \sqsubseteq T = \tilde{T} \tilde{\sqsubseteq} \tilde{S}$. □

From Property 5 and Definition 14 we infer the following property, using some set theory and the fact that S and T are prefix-closed.

Property 9 For $S,T \in \mathcal{T}$ we have $S \,\mathbf{nsi}\, T = S \sqsubseteq T$. □

The relation $\sqsubseteq$ is also equal to the **sat**-relation of the previous section as we show in the following lemmata.

Lemma 1 For $S,T,U \in \mathcal{T}$ such that $S \sqsubseteq T$ and $T \sqsubseteq U$ and for trace s and symbol $a \in I$ we have

$$s \in S \wedge s \in T \wedge sa \in U \Rightarrow sa \in S \wedge sa \in T.$$

Proof We derive

$$\begin{array}{ll} & s \in S \wedge s \in T \wedge sa \in U \\ \Rightarrow & \quad \{\, T \sqsubseteq U, \text{ hence, } U \subseteq_I T \,\} \\ & s \in S \wedge sa \in T \\ \Rightarrow & \quad \{\, S \sqsubseteq T, \text{ hence, } T \subseteq_I S \,\} \\ & sa \in S \wedge sa \in T \end{array}$$

□

Since this lemma holds for any $\mathcal{T}$ with the proper ordering, it holds for $\tilde{\mathcal{T}}$ and $\tilde{\sqsubseteq}$. By Property 8 we then have the following corollary.

Corollary 0 For $S,T,U \in \mathcal{T}$ such that $S \sqsubseteq T$ and $T \sqsubseteq U$ and for trace s and symbol $a \in O$ we have

$$sa \in S \wedge s \in T \wedge s \in U \Rightarrow sa \in T \wedge sa \in U.$$

□

Lemma 2 For $S,T,U \in \mathcal{T}$ such that $S \sqsubseteq T$ and $T \sqsubseteq U$ we have for trace s

$$s \in S \cap U \Rightarrow s \in T.$$

Proof By induction on the length of s.

Base: $s = \varepsilon$. Trivial, since $\varepsilon \in T$.

Step: We assume the implication to hold for trace s and consider trace sa. We distinguish the cases $a \in O$ and $a \in I$.

Case $a \in O$. We derive

$$
\begin{array}{ll}
 & sa \in S \cap U \\
\Rightarrow & \{\ S \cap U \text{ is prefix-closed since } S \text{ and } U \text{ are; set theory } \} \\
 & sa \in S \wedge s \in S \cap U \\
\Rightarrow & \{\ \text{induction hypothesis } \} \\
 & sa \in S \wedge s \in T \\
\Rightarrow & \{\ S \sqsubseteq T, \text{ in particular } S \subseteq_O T\ \} \\
 & sa \in T
\end{array}
$$

Case $a \in I$. Now we derive

$$
\begin{array}{ll}
 & sa \in S \cap U \\
\Rightarrow & \{\ S \cap U \text{ is prefix-closed since } S \text{ and } U \text{ are; set theory } \} \\
 & sa \in U \wedge s \in S \cap U \\
\Rightarrow & \{\ \text{induction hypothesis } \} \\
 & sa \in U \wedge s \in T \\
\Rightarrow & \{\ T \sqsubseteq U, \text{ in particular } U \subseteq_I T\ \} \\
 & sa \in T
\end{array}
$$

□

Theorem 0 The $\sqsubseteq$-relation is transitive.

Proof Let $S,T,U \in \mathcal{T}$ be such that $S \sqsubseteq T$ and $T \sqsubseteq U$. We want to show $S \sqsubseteq U$, which falls apart into $S \subseteq_O U$ and $U \subseteq_I S$. For the former case we derive

$$
\begin{array}{ll}
 & a \in O \wedge sa \in S \wedge s \in U \\
\Rightarrow & \{\ \text{Lemma 2, using the prefix-closedness of } S\ \} \\
 & a \in O \wedge sa \in S \wedge s \in T \wedge s \in U \\
\Rightarrow & \{\ \text{Corollary 0}\ \} \\
 & sa \in U
\end{array}
$$

and for the latter case we derive

$$a \in I \wedge sa \in U \wedge s \in S$$

$\Rightarrow$ { Lemma 2, using the prefix-closedness of U }
$a \in I \wedge s \in S \wedge s \in T \wedge sa \in U$
$\Rightarrow$ { Lemma 1 }
$sa \in S$

□

Notice that the case analysis in the last two proofs could have been avoided by switching to the space $\tilde{\mathcal{T}}$, realizing that the trace sets of a trace structure and its reflection are equal and that $S \subseteq T \subseteq U$ is equivalent to $\tilde{S} \tilde{\sqsubseteq} \tilde{T} \tilde{\sqsubseteq} \tilde{U}$.

Theorem 1 For $S,T \in \mathcal{T}$ we have $S \,\mathbf{sat}\, T = S \sqsubseteq T$.

Proof We derive for the implication from left to right

$S \,\mathbf{sat}\, T$
$=$ { Property 7 }
$(\forall U : T \,\mathbf{nsi}\, U : S \,\mathbf{nsi}\, U)$
$\Rightarrow$ { instantiation }
$T \,\mathbf{nsi}\, T \Rightarrow S \,\mathbf{nsi}\, T$
$=$ { $T \,\mathbf{nsi}\, T$, on account of Property 6 }
$S \,\mathbf{nsi}\, T$
$=$ { Property 9 }
$S \sqsubseteq T$

The other way round, it follows immediately from Property 9 and Theorem 0, the transitivity of $\sqsubseteq$. □

Theorem 2 $(\mathcal{T}, \sqsubseteq)$ is partially ordered.

Proof Reflexivity: Obvious, using Properties 6 and 9.
Antisymmetry: For $S,T \in \mathcal{T}$ we derive

$S \sqsubseteq T \wedge T \sqsubseteq S$
$=$ { definition of $\sqsubseteq$ }
$S \subseteq_O T \wedge T \subseteq_I S \wedge T \subseteq_O S \wedge S \subseteq_I T$
$\Rightarrow$ { Lemma 0, $I \cup O = A$ }
$S \subseteq T \wedge T \subseteq S$
$=$ { set theory }
$S = T$

Transitivity: Theorem 0. □

2.1 $\mathcal{T}$ as a Complete Lattice

In this section we show that $\langle \mathcal{T}, \sqsubseteq \rangle$ is a complete lattice. We define operators *lub* and *glb* on subsets X of $\mathcal{T}$ which we show to coincide with the least upper bound and greatest lower bound of X respectively.

Definition 15 For $X \subseteq \mathcal{T}$ we define $lub.X$ recursively in the following way.

- $\varepsilon \in lub.X$
- for trace s and symbols $a \in I$ and $p \in O$

$$\begin{aligned} sa \in lub.X &= s \in lub.X \wedge (\forall S : S \in X \wedge s \in S : sa \in S) \\ sp \in lub.X &= s \in lub.X \wedge (\exists S : S \in X : sp \in S) \end{aligned}$$

□

It should be clear that *lub* is well-defined and yields an element of $\mathcal{T}$. The function *glb* is defined similarly by interchanging I and O.

Property 10 For non-empty $X \subseteq \mathcal{T}$ we have

$$lub.X, glb.X \subseteq (\bigcup S : S \in X : S)$$

and

$$(\bigcap S : S \in X : S) \subseteq lub.X, glb.X.$$

□

Theorem 3 $\langle \mathcal{T}, \sqsubseteq \rangle$ is a complete lattice.

Proof We show that $lub.X$ is the least upper bound of any $X \subseteq \mathcal{T}$. Therefore, each $X \subseteq \tilde{\mathcal{T}}$ has a least upper bound using the $\tilde{\sqsubseteq}$-relation. Using Property 8 we infer that any $X \subseteq \mathcal{T}$ has a greatest lower bound in the $\sqsubseteq$-relation, by which $\mathcal{T}$ is a lattice.

Let $X \subseteq \mathcal{T}$. We show that

0. $(\forall S : S \in X : S \sqsubseteq lub.X)$ and
1. $(\forall U : U \in \mathcal{T} \wedge (\forall S : S \in X : S \sqsubseteq U) : lub.X \sqsubseteq U)$

Ad 0. $(\forall S : S \in X : S \sqsubseteq lub.X)$

Let $S \in X$. We have to show $S \sqsubseteq lub.X$, which falls apart into $lub.X \subseteq_I S$ and $S \subseteq_O lub.X$. For any trace s and symbol $a \in I$ we derive

$$
\begin{array}{ll}
 & s \in S \wedge sa \in lub.X \\
= & \quad \{ \text{ definition of } lub \ \} \\
 & s \in S \wedge (\forall T : T \in X \wedge s \in T : sa \in T) \wedge s \in lub.X \\
\Rightarrow & \quad \{ \ S \in X, \text{ instantiation } \} \\
 & sa \in S
\end{array}
$$

which proves $lub.X \subseteq_I S$. Furthermore, we derive for any trace s and symbol $p \in O$

$$
\begin{array}{ll}
 & sp \in S \wedge s \in lub.X \\
\Rightarrow & \quad \{ \text{ predicate calculus, using } S \in X \ \} \\
 & s \in lub.X \wedge (\exists T : T \in X : sp \in T) \\
= & \quad \{ \text{ definition of } lub \ \} \\
 & sp \in lub.X
\end{array}
$$

which proves $S \subseteq_O lub.X$.

Ad 1. $(\forall U : U \in \mathcal{T} \wedge (\forall S : S \in X : S \sqsubseteq U) : lub.X \sqsubseteq U)$

Let $U \in \mathcal{T}$ be such that $(\forall S : S \in X : S \sqsubseteq U)$. We have to show $lub.X \sqsubseteq U$, i.e. $U \subseteq_I lub.X$ and $lub.X \subseteq_O U$. We derive for any trace s and symbol $a \in I$

$$
\begin{array}{ll}
 & s \in lub.X \wedge sa \in U \\
= & \quad \{ \ U \text{ is an upper bound of } X \ \} \\
 & s \in lub.X \wedge sa \in U \wedge (\forall S : S \in X : S \sqsubseteq U) \\
\Rightarrow & \quad \{ \text{ definition of } \sqsubseteq, \text{ in particular } U \subseteq_I S \ \} \\
 & s \in lub.X \wedge sa \in U \wedge (\forall S : S \in X \wedge s \in S \wedge sa \in U : sa \in S) \\
\Rightarrow & \quad \{ \text{ predicate calculus } \} \\
 & s \in lub.X \wedge (\forall S : S \in X \wedge s \in S : sa \in S) \\
= & \quad \{ \text{ definition of } lub \ \} \\
 & sa \in lub.X
\end{array}
$$

which proves $U \subseteq_I lub.X$. Moreover, we derive for any trace s and symbol $p \in O$

$$
\begin{array}{ll}
 & sp \in lub.X \wedge s \in U \\
= & \quad \{ \text{ definition of } lub \ \} \\
 & (\exists S : S \in X : sp \in S) \wedge s \in lub.X \wedge s \in U
\end{array}
$$

$\Rightarrow$ { predicate calculus }
$(\exists S : S \in X : sp \in S \wedge s \in U)$
$\Rightarrow$ { U is an upper bound of X, definition of $\sqsubseteq$, in particular $S \subseteq_O U$ }
$(\exists S : S \in X : sp \in U)$
$\Rightarrow$ { predicate calculus }
$sp \in U$

which proves $lub.X \subseteq_O U$. □

The reader may check that the bottom element of this lattice is I^* and that the top element is O^*.

2.2 The Empty Directed Trace Structure

As mentioned earlier, we do not want to confine ourselves to non-empty DTS's. So far, however, we have not taken the empty DTS into account. The reason is that it would only complicate the proofs in the previous sections. From now onwards we assume $\mathcal{T}$ to contain the empty DTS and add it as the new top of the lattice. More formally, the ordering becomes

$$S \sqsubseteq T = S \subseteq_O T \wedge T \subseteq_I S \wedge (\varepsilon \in T \Rightarrow \varepsilon \in S).$$

It is obvious that the new space is a complete lattice. Some properties do no longer hold, however. Properties 5, 6, and 9, and Lemma 0 hold only for non-empty DTS's.

3 Delay-Insensitive DTS's

Embedded in $\mathcal{T}$ are the delay-insensitive DTS's, the definition of which has been given in [3]. In this section we show that they almost form a sublattice in $\mathcal{T}$. Only the empty collection of delay-insensitive DTS's does not have a delay-insensitive lower or upper bound. For the sake of completeness we give here the definition of a delay-insensitive DTS. Symbols are said to be of the same type if they are both input or both output symbols.

Definition 16 A non-empty DTS $S \in \mathcal{T}$ is delay-insensitive when for all traces s and t, and symbols a, b, and c

$\mathbf{R_2}$ $saa \notin S$.

R3 $sabt \in S = sbat \in S$ for symbols a and b of the same type.

R4 $sabtc \in S \wedge sbat \in S \Rightarrow sbatc \in S$ for symbols a and c of another type than b.

R5 $sa \in S \wedge sb \in S \Rightarrow sab \in S$ for symbols a and b of different types.

□

Lemma 3 For $X \subseteq \mathcal{T}$ a collection of delay-insensitive *DTS*'s, U the greatest lower bound of X, $S \in X$, and for traces s and t and symbols $a \in I$ and $p \in O$

$$spat \in U \wedge sapt \in S \Rightarrow spat \in S.$$

Proof We prove this lemma by induction on the length of t.

Base: $t = \varepsilon$. We derive

$$\begin{array}{ll} & spat \in U \wedge sapt \in S \\ = & \quad \{\ t = \varepsilon\ \} \\ & spa \in U \wedge sap \in S \\ \Rightarrow & \quad \{\ \text{definition of } glb, \text{ which is the greatest lower bound, } p \in O\ \} \\ & (\forall T : T \in X \wedge s \in T : sp \in T) \wedge sap \in S \\ \Rightarrow & \quad \{\ \text{instantiation, } S \in X \text{ is prefix-closed}\ \} \\ & sp \in S \wedge sa \in S \\ \Rightarrow & \quad \{\ S \text{ is delay-insensitive, rule } \mathbf{R_5}\ \} \\ & spa \in S \\ = & \quad \{\ t = \varepsilon\ \} \\ & spat \in S \end{array}$$

Step: For the induction step we discern between the traces ending in an input and the ones ending in an output symbol. We derive for $b \in I$

$$\begin{array}{ll} & spatb \in U \wedge saptb \in S \\ \Rightarrow & \quad \{\ U \text{ and } S \text{ are prefix-closed}\ \} \\ & spat \in U \wedge sapt \in S \wedge saptb \in S \\ \Rightarrow & \quad \{\ \text{induction hypothesis}\ \} \\ & spat \in S \wedge saptb \in S \\ \Rightarrow & \quad \{\ S \text{ is delay-insensitive, rule } \mathbf{R_4}\ \} \end{array}$$

$spatb \in S$

and for $q \in O$ we derive

$spatq \in U \wedge saptq \in S$
$=$ { definition of *glb*, which is the greatest lower bound, $q \in O$ }
$spat \in U \wedge (\forall T : T \in X \wedge spat \in T : spatq \in T) \wedge saptq \in S$
$\Rightarrow$ { induction hypothesis, using that S is prefix-closed }
$spat \in S \wedge (\forall T : T \in X \wedge spat \in T : spatq \in T)$
$\Rightarrow$ { $S \in X$, instantiation }
$spatq \in S$

□

Theorem 4 The greatest lower bound of a non-empty collection of delay-insensitive trace sets is delay-insensitive.

Proof Let X be a a non-empty collection of delay-insensitive trace sets with greatest lower bound U. We show that U satisfies the rules $\mathbf{R_2}$ through $\mathbf{R_5}$.

$\mathbf{R_2}$: U is contained in $(\bigcup S : S \in X : S)$ on account of Property 10, using $X \neq \emptyset$. None of the trace sets of X contain traces of the form saa, so neither does U.

$\mathbf{R_3}$: Let s and t be traces and let a and b be input symbols and let p and q be output symbols. We prove the theorem by induction on the length of t.

Base: $t = \varepsilon$. For inputs we derive

$sabt \in U$
$=$ { $t = \varepsilon$ }
$sab \in U$
$=$ { definition of *glb*, which is the greatest lower bound; elements of X are prefix-closed; $a, b \in I$ }
$s \in U \wedge (\exists S : S \in X : sab \in S)$
$=$ { elements of X are delay-insensitive, rule $\mathbf{R_3}$ }
$s \in U \wedge (\exists S : S \in X : sba \in S)$
$=$ { elements of X are prefix-closed; definition of *glb*; $a, b \in I$ }
$sba \in U$

$$
\begin{array}{ll}
= & \{\ t = \varepsilon\ \} \\
& \mathit{sbat} \in U
\end{array}
$$

and for outputs

$$
\begin{array}{ll}
& \mathit{spqt} \in U \\
= & \{\ t = \varepsilon\ \} \\
& \mathit{spq} \in U \\
= & \{\ \text{definition of } \mathit{glb};\ p, q \in O\ \} \\
& s \in U \wedge (\forall S : S \in X \wedge s \in S : sp \in S) \wedge (\forall S : S \in X \wedge sp \in S : spq \in S) \\
= & \{\ \text{predicate calculus, elements of } X \text{ are prefix-closed}\ \} \\
& s \in U \wedge (\forall S : S \in X \wedge s \in S : spq \in S) \\
= & \{\ \text{elements of } X \text{ are delay-insensitive, rule } \mathbf{R_3}\ \} \\
& s \in U \wedge (\forall S : S \in X \wedge s \in S : sqp \in S) \\
= & \{\ \text{elements of } X \text{ are prefix-closed, predicate calculus}\ \} \\
& s \in U \wedge (\forall S : S \in X \wedge s \in S : sq \in S) \wedge (\forall S : S \in X \wedge sq \in S : sqp \in S) \\
= & \{\ \text{definition of } \mathit{glb};\ p, q \in O\ \} \\
& \mathit{sqp} \in U \\
= & \{\ t = \varepsilon\ \} \\
& \mathit{sqpt} \in U
\end{array}
$$

Step: Now we assume that U satisfies $\mathbf{R_3}$ for traces t of a certain length. For traces that are one longer we discern between the type of symbol in which they end. If the last symbol is in I we derive

$$
\begin{array}{ll}
& \mathit{sabtc} \in U \\
= & \{\ \text{definition of } \mathit{glb};\ c \in I\ \} \\
& \mathit{sabt} \in U \wedge (\exists S : S \in X : \mathit{sabtc} \in S) \\
= & \{\ \text{induction hypothesis, } \mathbf{R_3} \text{ holds for the elements of } X\ \} \\
& \mathit{sbat} \in U \wedge (\exists S : S \in X : \mathit{sbatc} \in S) \\
= & \{\ \text{definition of } \mathit{glb};\ c \in I\ \} \\
& \mathit{sbatc} \in U
\end{array}
$$

and if the last symbol is in O we derive

$$
\begin{array}{ll}
& \mathit{sabtc} \in U \\
= & \{\ \text{definition of } \mathit{glb};\ c \in O\ \}
\end{array}
$$

$$
\begin{array}{ll}
 & sabt \in U \wedge (\forall\, S : S \in X : sabtc \in S) \\
= & \quad \{ \text{ induction hypothesis and } \mathbf{R_3} \text{ holds for the elements of } X \} \\
 & sbat \in U \wedge (\forall\, S : S \in X : sbatc \in S) \\
= & \quad \{ \text{ definition of } glb;\ c \in O \} \\
 & sbatc \in U
\end{array}
$$

$\mathbf{R_4}$: For traces s and t, and input symbols a and b and output symbols p and q we derive

$$
\begin{array}{ll}
 & saptb \in U \wedge spat \in U \\
\Rightarrow & \quad \{ \text{ definition of } glb;\ b \in I \} \\
 & (\exists\, S : S \in X : saptb \in S) \wedge spat \in U \\
= & \quad \{ \text{ elements of } X \text{ are prefix-closed, predicate calculus } \} \\
 & spat \in U \wedge (\exists\, S : S \in X : saptb \in S \wedge sapt \in S \wedge spat \in U) \\
\Rightarrow & \quad \{ \text{ Lemma 3 } \} \\
 & spat \in U \wedge (\exists\, S : S \in X : saptb \in S \wedge spat \in S) \\
\Rightarrow & \quad \{ \text{ elements of } X \text{ are delay-insensitive, rule } \mathbf{R_4} \} \\
 & spat \in U \wedge (\exists\, S : S \in X : spatb \in S) \\
= & \quad \{ \text{ definition of } glb;\ b \in I \} \\
 & spatb \in U
\end{array}
$$

and

$$
\begin{array}{ll}
 & spatq \in U \wedge sapt \in U \\
= & \quad \{ \text{ definition of } glb;\ p \in O \} \\
 & spat \in U \wedge (\forall\, S : S \in X \wedge spat \in S : spatq \in S) \wedge sapt \in U \\
\Rightarrow & \quad \{ \text{ Lemma 3 } \} \\
 & (\forall\, S : S \in X \wedge sapt \in S : spat \in S) \wedge \\
 & (\forall\, S : S \in X \wedge spat \in S : spatq \in S) \wedge sapt \in U \\
\Rightarrow & \quad \{ \text{ predicate calculus } \} \\
 & (\forall\, S : S \in X \wedge sapt \in S : spatq \in S) \wedge sapt \in U \\
\Rightarrow & \quad \{ \text{ elements of } X \text{ are delay-insensitive, rule } \mathbf{R_4} \} \\
 & (\forall\, S : S \in X \wedge sapt \in S : saptq \in S) \wedge sapt \in U \\
= & \quad \{ \text{ definition of } glb;\ q \in O \} \\
 & saptq \in U
\end{array}
$$

$\mathbf{R_5}$: For trace s and symbols $a \in I$ and $p \in O$ we derive

$$
\begin{array}{ll}
 & sa \in U \wedge sp \in U \\
= & \quad \{ \text{ definition of } glb;\ U \text{ is prefix-closed; } a \in I \text{ and } p \in O \ \} \\
 & sa \in U \wedge sp \in U \wedge (\exists\, S : S \in X : sa \in S) \wedge \\
 & (\forall\, S : S \in X \wedge s \in S : sp \in S) \\
\Rightarrow & \quad \{ \text{ elements of } X \text{ are prefix-closed } \} \\
 & sa \in U \wedge sp \in U \wedge (\exists\, S : S \in X : sa \in S) \wedge \\
 & (\forall\, S : S \in X \wedge sa \in S : sp \in S) \\
\Rightarrow & \quad \{ \text{ predicate calculus } \} \\
 & sa \in U \wedge sp \in U \wedge (\exists\, S : S \in X : sa \in S \wedge sp \in S) \wedge \\
 & (\forall\, S : S \in X \wedge sa \in S : sa \in S \wedge sp \in S) \\
\Rightarrow & \quad \{ \text{ elements of } X \text{ are delay-insensitive, rule } \mathbf{R_5} \ \} \\
 & sa \in U \wedge sp \in U \wedge (\exists\, S : S \in X : spa \in S) \wedge \\
 & (\forall\, S : S \in X \wedge sa \in S : sap \in S) \\
= & \quad \{ \text{ definition of } glb;\ a \in I \text{ and } p \in O \ \} \\
 & spa \in U \wedge spa \in U
\end{array}
$$

□

4 Isomorphic Specification Domains

In this section we present two alternative representations for directed specifications. The advantages of the alternative representations are that the satisfaction relation (**sat**) is less complicated, that they are easier to extend when other correctness concerns are added, such as progress, and that composition may be expressed more concisely.

Let I and O be disjoint alphabets and $A = I \cup O$. The domain $\mathcal{T}$ of specifications introduced in the preceding section consists of all DTS's with input alphabet I and output alphabet O. Since the input and output alphabet are fixed we can restrict ourselves to the trace sets of these DTS's, that is, we identify prefix-closed trace sets over A with DTS's in $\mathcal{T}$. The empty trace set is included in this section's investigation. On $\mathcal{T}$ the relation $\sqsubseteq$ was defined by

$$S \sqsubseteq T \;=\; S \subseteq_O T \;\wedge\; T \subseteq_I S \;\wedge\; (\varepsilon \in T \Rightarrow \varepsilon \in S)$$

for all S and T in $\mathcal{T}$. This is the satisfaction relation for the synchronous game and, as was shown earlier, it is a partial order.

Definition 17 For alphabet B, trace set S is *B-extension-closed* when

$$(\forall\, s, a : s \in S \wedge a \in B : sa \in S)$$

and it is *B-chop-closed* when

$$(\forall\, s, a : sa \in S \wedge a \in B : s \in S).$$

□

Notice that prefix-closed and A-chop-closed are the same for trace sets over A.

Definition 18 The first alternative domain $\mathcal{T}'$ consists of all pairs (T, U) such that T is a non-empty prefix-closed I-extension-closed trace set over A and U is an A-extension-closed O-chop-closed subset of T. The relation $\sqsubseteq'$ on $\mathcal{T}'$ is defined by

$$(T, U) \sqsubseteq' (V, W) \;=\; T \subseteq V \;\wedge\; U \subseteq W$$

for all (T, U) and (V, W) in $\mathcal{T}'$. □

It is straightforward to show that $\sqsubseteq'$ is a partial order on $\mathcal{T}'$. (cf. failures model for CSP [1])

A mechanistic interpretation of specification (T, U) in $\mathcal{T}'$ is the following. Trace set T contains those traces that are not excluded from occurring by this specification; that is, only traces not in T are guaranteed not to occur. I-extension-closedness of T expresses that a specification can never exclude the extension with an input symbol (mechanisms are passive with regard to their inputs). The trace set U contains those traces for which the specification allows undefined behavior. Extension-closedness of U expresses that once undefined behavior is allowed it remains allowed. O-chop-closedness of U expresses that an extension with an output symbol does not lead to undefined behavior unless in the preceding state undefined behavior was already allowed (mechanisms are active with regard to their outputs). Hence, undefined behavior can occur only initially or from a state with well-defined behavior via the extension with an input symbol.

We claim that the posets $\langle \mathcal{T}, \sqsubseteq \rangle$ and $\langle \mathcal{T}', \sqsubseteq' \rangle$ are isomorphic. Before proving this we introduce some auxiliary concepts and discuss their properties.

Property 11 For $X \subseteq \mathcal{T}$ and $T \in \mathcal{T}$ we have

$$(\forall S : S \in X : S \sqsubseteq T) \Rightarrow (\bigsqcup S : S \in X : S) \sqsubseteq T.$$

□

Definition 19 For specification S in $\mathcal{T}$ we define trace sets $\mathbf{x}S$ and $\mathbf{u}S$ by

$$\begin{aligned} \mathbf{x}S &= (\bigcup X : X \sqsubseteq S : X), \\ \mathbf{u}S &= \mathbf{x}S \setminus S. \end{aligned}$$

We call $\mathbf{x}S$ the *extended* trace set of S and $\mathbf{u}S$ the *undefined* trace set of S. □

Since the union of prefix-closed trace sets is prefix-closed, we have $\mathbf{x}S \in \mathcal{T}$. From $S \sqsubseteq S$ it follows that $S \subseteq \mathbf{x}S$. From Property 11 we infer $\mathbf{x}S \sqsubseteq S$. Because $\emptyset$ is the top of $\mathcal{T}$, we have $A^* \sqsubseteq \emptyset$ and, hence, $\mathbf{x}\emptyset = A^*$ and $\mathbf{u}\emptyset = A^*$. Furthermore, because I^* is the bottom of $\mathcal{T}$, we have $X \sqsubseteq I^* \Rightarrow X = I^*$ and, thus, $\mathbf{x}I^* = I^*$ and $\mathbf{u}I^* = \emptyset$.

As an exercise the reader may prove such things as

$$\mathbf{x}S = (\bigcup X : X \sqsubseteq S \wedge S \subseteq X : X),$$

$$S \subseteq X \Rightarrow (X \sqsubseteq S = X \subseteq \mathbf{x}S),$$

$$\mathbf{x}(\mathbf{x}S) = \mathbf{x}S,$$

$$\mathbf{x}S = \{t : t \in A^* \wedge (\forall s, a : s \in S \wedge sa \leq t \wedge sa \notin S : a \in I) : t\}.$$

but we shall not rely on these properties later on.

Lemma 4 For specifications R and S in $\mathcal{T}$ such that $R \sqsubseteq S$ we have $\mathbf{x}R \subseteq \mathbf{x}S$ and $\mathbf{u}R \subseteq \mathbf{u}S$.

Proof Assume $R \sqsubseteq S$. We derive

$$\begin{aligned} & \mathbf{x}R \\ = & \quad \{ \text{ definition of } \mathbf{x} \} \\ & (\bigcup X : X \sqsubseteq R : X) \\ \subseteq & \quad \{ R \sqsubseteq S \text{ and transitivity of } \sqsubseteq \} \\ & (\bigcup X : X \sqsubseteq S : X) \end{aligned}$$

$$
\begin{array}{ll}
= & \{ \text{ definition of x } \} \\
& \mathrm{x}S
\end{array}
$$

To show the second inclusion it is now sufficient to prove $\mathrm{x}R \cap S \subseteq R$. We do this by mathematical induction on the length of the traces.

Base: $s = \varepsilon$. We derive

$$
\begin{array}{ll}
& s \in \mathrm{x}R \cap S \\
\Rightarrow & \{ \ s = \varepsilon \text{ and set theory } \} \\
& \varepsilon \in S \\
\Rightarrow & \{ \ R \sqsubseteq S \ \} \\
& \varepsilon \in R \\
= & \{ \ s = \varepsilon \ \} \\
& s \in R
\end{array}
$$

Step $s = s_0 a$. We derive

$$
\begin{array}{ll}
& s \in \mathrm{x}R \cap S \\
= & \{ \ s = s_0 a \text{ and the prefix-closedness of } \mathrm{x}R \text{ and } S \ \} \\
& s_0 a \in \mathrm{x}R \cap S \ \wedge \ s_0 \in \mathrm{x}R \cap S \\
\Rightarrow & \{ \text{ set theory and induction hypothesis, using } \ell(s_0) < \ell(s) \ \} \\
& s_0 a \in \mathrm{x}R \ \wedge \ s_0 a \in S \ \wedge \ s_0 \in R \\
\Rightarrow & \{ \ \mathrm{x}R \sqsubseteq R \text{ for } a \in O \text{ and } R \sqsubseteq S \text{ for } a \in I \ \} \\
& s_0 a \in R \\
= & \{ \ s_0 a = s \ \} \\
& s \in R
\end{array}
$$

□

Property 12 For specifications S and T with $S \sqsubseteq T$, trace s, and symbol $a \in I$ we have

$$s \in S \ \Rightarrow \ S \cup \{sa\} \sqsubseteq T,$$

and

$$s \in S \wedge s \notin T \ \Rightarrow \ S \cup \{t : t \in A^* : st\} \sqsubseteq T.$$

□

Theorem 5 The posets $\langle \mathcal{T}, \sqsubseteq \rangle$ and $\langle \mathcal{T}', \sqsubseteq' \rangle$ are isomorphic.

Proof Consider the mappings $f: \mathcal{T} \to \mathcal{T}'$ and $g: \mathcal{T}' \to \mathcal{T}$ defined by

$$\begin{aligned} f(S) &= (\mathbf{x}S, \mathbf{u}S), \\ g(T,U) &= T \setminus U \end{aligned}$$

for all $S \in \mathcal{T}$ and $(T,U) \in \mathcal{T}'$. Then f is an isomorphism between $\langle \mathcal{T}, \sqsubseteq \rangle$ and $\langle \mathcal{T}', \sqsubseteq' \rangle$ with inverse g. More specifically, we claim for R and S in $\mathcal{T}$ and for (T,U) and (V,W) in $\mathcal{T}'$

0. $f(S) \in \mathcal{T}'$,
1. $g(T,U) \in \mathcal{T}$,
2. $g(f(S)) = S$, hence, f is one-to-one,
3. $f(g(T,U)) = (T,U)$, hence, f is onto,
4. $R \sqsubseteq S \;\Rightarrow\; f(R) \sqsubseteq' f(S)$,
5. $(T,U) \sqsubseteq' (V,W) \;\Rightarrow\; g(T,U) \sqsubseteq g(V,W)$.

Ad 0. As noted earlier $\mathbf{x}S$ is prefix-closed. From $S \subseteq \mathbf{x}S$ and $\mathbf{x}\emptyset = A^*$ it follows that $\mathbf{x}S$ is non-empty. Property 12 implies that $\mathbf{x}S$ is I-extension-closed. Clearly $\mathbf{u}S$ is a subset of $\mathbf{x}S$ and it is A-extension-closed by Property 12. We prove that $\mathbf{u}S$ is O-chop-closed by deriving for trace s and symbol $p \in O$

$$\begin{aligned} & sp \in \mathbf{u}S \\ = \quad & \{ \text{ definition of } \mathbf{u} \text{ and set theory } \} \\ & sp \in \mathbf{x}S \wedge sp \notin S \\ = \quad & \{ \text{ definition of } \mathbf{x} \text{ and predicate calculus } \} \\ & (\exists X : X \sqsubseteq S : sp \in X \wedge sp \notin S) \\ \Rightarrow \quad & \{ \text{ definition of } \sqsubseteq, \text{ in particular } \subseteq_O, \text{ using } p \in O \} \\ & (\exists X : X \sqsubseteq S : sp \in X \wedge s \notin S) \\ \Rightarrow \quad & \{ X \text{ is prefix-closed } \} \\ & (\exists X : X \sqsubseteq S : s \in X \wedge s \notin S) \\ = \quad & \{ \text{ definition of } \mathbf{x} \text{ and predicate calculus } \} \\ & s \in \mathbf{x}S \wedge s \notin S \\ = \quad & \{ \text{ definition of } \mathbf{u} \text{ and set theory } \} \\ & s \in \mathbf{u}S \end{aligned}$$

Ad 1. The trace set $T \setminus U$ is prefix-closed because T is prefix-closed and U is A-extension-closed.

Ad 2. We show $\mathbf{x}S \setminus \mathbf{u}S = S$:

$$\begin{array}{ll} & \mathbf{x}S \setminus \mathbf{u}S \\ = & \quad \{ \text{ definition of } \mathbf{u}S \} \\ & \mathbf{x}S \setminus (\mathbf{x}S \setminus S) \\ = & \quad \{ \ S \subseteq \mathbf{x}S \ \} \\ & S \end{array}$$

Ad 3. We show $\mathbf{x}(T \setminus U) = T$ and $\mathbf{u}(T \setminus U) = U$. We first deal with the first equality. Since U is O-chop-closed and $T \setminus U \subseteq T$ it follows that $T \sqsubseteq T \setminus U$ and, hence, the definition of $\mathbf{x}$ yields $T \subseteq \mathbf{x}(T \setminus U)$. The other inclusion is readily proven by mathematical induction on the trace length and is left as an exercise to the reader. Now we derive the second equality:

$$\begin{array}{ll} & \mathbf{u}(T \setminus U) \\ = & \quad \{ \text{ definition of } \mathbf{u} \} \\ & \mathbf{x}(T \setminus U) \setminus (T \setminus U) \\ = & \quad \{ \ \mathbf{x}(T \setminus U) = T \ \} \\ & T \setminus (T \setminus U) \\ = & \quad \{ \ U \subseteq T \ \} \\ & U \end{array}$$

Ad 4. This follows immediately from Lemma 4.

Ad 5. Assume $(T,U) \sqsubseteq' (V,W)$. We show $T \setminus U \sqsubseteq V \setminus W$. For trace t and symbol a we derive

$$\begin{array}{ll} & a \in O \ \wedge \ ta \in T \setminus U \ \wedge \ t \in V \setminus W \\ \Rightarrow & \quad \{ \text{ set theory } \} \\ & a \in O \ \wedge \ ta \in T \ \wedge \ t \notin W \\ \Rightarrow & \quad \{ \ T \subseteq V \text{ and } W \text{ is } O\text{-chop-closed } \} \\ & ta \in V \ \wedge \ ta \notin W \\ = & \quad \{ \text{ set theory } \} \\ & ta \in V \setminus W \end{array}$$

and

$$
\begin{array}{ll}
 & a \in I \wedge t \in T \setminus U \wedge ta \in V \setminus W \\
\Rightarrow & \quad \{ \text{ set theory } \} \\
 & a \in I \wedge t \in T \wedge ta \notin W \\
\Rightarrow & \quad \{ T \text{ is } I\text{-extension-closed and } U \subseteq W \} \\
 & ta \in T \wedge ta \notin U \\
= & \quad \{ \text{ set theory } \} \\
 & ta \in T \setminus U
\end{array}
$$

and finally

$$
\begin{array}{ll}
 & \varepsilon \in V \setminus W \\
\Rightarrow & \quad \{ T \text{ is non-empty and prefix-closed, and set theory } \} \\
 & \varepsilon \in T \wedge \varepsilon \notin W \\
\Rightarrow & \quad \{ U \subseteq W \} \\
 & \varepsilon \in T \wedge \varepsilon \notin U \\
\Rightarrow & \quad \{ \text{ set theory } \} \\
 & \varepsilon \in T \setminus U
\end{array}
$$

□

Corollary 1 The poset $\langle \mathcal{T}', \sqsubseteq' \rangle$ is a complete lattice.

Proof According to Theorem 33 $\langle \mathcal{T}, \sqsubseteq \rangle$ is a complete lattice. Now apply the above Theorem. □

In $\langle \mathcal{T}', \sqsubseteq' \rangle$ the bottom is $(I^*, \emptyset)$ and the top is (A^*, A^*). The specification O^* in $\mathcal{T}$, which is the immediate predecessor of the top, corresponds to $(A^*, A^* \setminus O^*)$ in $\mathcal{T}'$. For non-empty subset X of $\mathcal{T}'$ it is readily verified that

$$((\bigcup T, U : (T, U) \in X : T), (\bigcup T, U : (T, U) \in X : U))$$

is in $\mathcal{T}'$ and that it is the least upper bound of X. Its greatest lower bound is obtained similarly by intersection.

Definition 20 The second alternative domain $\mathcal{T}''$ consists of all subsets F of $A^* \times \{\bot, \top\}$ such that for all traces s and t, and symbol a

0. $(\varepsilon, \top) \in F$,
1. $(st, \top) \in F \Rightarrow (s, \top) \in F$,

2. $(s, \bot) \in F \Rightarrow (s, \top) \in F$,
3. $(s, \bot) \in F \wedge t \in A^* \Rightarrow (st, \bot) \in F$,
4. $(s, \top) \in F \wedge a \in I \Rightarrow (sa, \top) \in F$,
5. $(sa, \bot) \in F \wedge a \in O \Rightarrow (s, \bot) \in F$.

The relation $\sqsubseteq''$ on $\mathcal{T}''$ is defined by

$$F \sqsubseteq'' G \;=\; F \subseteq G$$

for all F and G in $\mathcal{T}''$. □

The relation $\sqsubseteq''$ is obviously a partial order.

Theorem 6 The posets $\langle \mathcal{T}', \sqsubseteq' \rangle$ and $\langle \mathcal{T}'', \sqsubseteq'' \rangle$ are isomorphic.

Proof It is straightforward to check that the mapping $f: \mathcal{T}' \to \mathcal{T}''$ defined by

$$f(T, U) = \{t : t \in T : (t, \top)\} \cup \{u : u \in U : (u, \bot)\}$$

for all $(T, U) \in \mathcal{T}'$ is an isomorphism between $\langle \mathcal{T}', \sqsubseteq' \rangle$ and $\langle \mathcal{T}'', \sqsubseteq'' \rangle$ with inverse $g: \mathcal{T}'' \to \mathcal{T}'$ defined by

$$g(F) = (\{t : (t, \top) \in F : t\}, \{u : (u, \bot) \in F : u\})$$

for all $F \in \mathcal{T}''$. In fact, there is the following correspondence between the requirements for membership of $\mathcal{T}'$ and $\mathcal{T}''$.

0. T is non-empty (together with 1.),
1. T is prefix-closed,
2. U is a subset of T,
3. U is A-extension-closed,
4. T is I-extension-closed,
5. U is O-chop-closed.

□

Again $\langle \mathcal{T}'', \sqsubseteq'' \rangle$ is a complete lattice. Its bottom is $I^* \times \{\top\}$ and its top is $A^* \times \{\bot, \top\}$. Least upper bound and greatest lower bound of non-empty sets are obtained by union and intersection respectively.

Finally, we would like to suggest a definition for synchronous parallel composition (including hiding of internal communications) based on the domain $\mathcal{T}'$. A specification S is uniquely characterized by its four components $\mathbf{i}S$, $\mathbf{o}S$, $\mathbf{x}S$, and $\mathbf{u}S$, where the pair $(\mathbf{x}S, \mathbf{u}S)$ belongs to the domain $\mathcal{T}'$ with $I = \mathbf{i}S$ and $O = \mathbf{o}S$. The idea behind the definition is that undefinedness is inherited from either operand and is back-propagated over output symbols.

Definition 21 Given specifications S and T such that

$$\mathbf{i}S \cap \mathbf{i}T = \emptyset = \mathbf{o}S \cap \mathbf{o}T,$$

we define their synchronous parallel composite U by

$$\begin{aligned}
\mathbf{i}U &= (\mathbf{i}S \cup \mathbf{i}T) \setminus (\mathbf{o}S \cup \mathbf{o}T), \\
\mathbf{o}U &= (\mathbf{o}S \cup \mathbf{o}T) \setminus (\mathbf{i}S \cup \mathbf{i}T), \\
\mathbf{x}U &= \{u : u \in (\mathbf{a}S \cup \mathbf{a}T)^* \wedge u\lceil \mathbf{a}S \in \mathbf{x}S \wedge u\lceil \mathbf{a}T \in \mathbf{x}T : u\lceil(\mathbf{a}S \div \mathbf{a}T)\}, \\
\mathbf{u}U &= \{s,t,u : su \in (\mathbf{a}S \cup \mathbf{a}T)^* \wedge t \in (\mathbf{o}S \cup \mathbf{o}T)^* \wedge \\
&\qquad (((st)\lceil \mathbf{a}S \in \mathbf{x}S \wedge (st)\lceil \mathbf{a}T \in \mathbf{u}T) \vee \\
&\qquad ((st)\lceil \mathbf{a}S \in \mathbf{u}S \wedge (st)\lceil \mathbf{a}T \in \mathbf{x}T))) : (su)\lceil(\mathbf{a}S \div \mathbf{a}T)\}.
\end{aligned}$$

□

That this is a proper definition and what its relationship is with composition as discussed in Section 6 remains to be investigated.

5 Analysis of the Asynchronous 2-Person Game

In this section we analyze the asynchronous two-person game in detail. We begin by characterizing the safe reachable game vectors in this game and we proceed with a number of properties relating **nai** and $\sqsubseteq$. Then we study the asynchronous two-person versions of **sat** and **equ**. The major result of this section states that each equivalence class under **equ** contains exactly one specification that is delay-insensitive in a certain sense. For these delay-insensitive specifications the relations **nai** and **sat** are equal to their synchronous counterparts and, hence, equal to $\sqsubseteq$.

As usual, let I and O be disjoint alphabets and $A = I \cup O$, and let $\mathcal{T}$ be the set of non-empty DTS's with input alphabet I and output alphabet O. We can now define elements of $\mathcal{T}$ by giving just their trace set. We ignore the empty specification in this section for a reason to be explained at the end of this Section. We recall the partial order $\sqsubseteq$ on $\mathcal{T}$ as defined earlier:

$$S \sqsubseteq T \;=\; S \subseteq_O T \;\wedge\; T \subseteq_I S.$$

Least upper and greatest lower bounds are with respect to this order.

5.0 Safe Reachable Game Vectors

First we characterize the safe reachable game vectors in the asynchronous two-person game.

Definition 22 On A^* the relation $\mathbf{C}$ is defined as follows. For s and t traces over A, $s\,\mathbf{C}\,t$ holds when the game vector (s,t) is reachable in the asynchronous (A^*,A^*)-game. Notice that in this game all game vectors are safe. □

Example 0 Although the specification A^* does not exclude any traces from occurring in a game vector, not every pair of traces stands in the $\mathbf{C}$-relationship. For instance, for symbol $a \in I$ and symbol $p \in O$ we do have

$$\varepsilon\,\mathbf{C}\,a \;\wedge\; ap\,\mathbf{C}\,ap \;\wedge\; pa\,\mathbf{C}\,ap,$$

but also

$$\neg(a\,\mathbf{C}\,\varepsilon) \;\wedge\; \neg(ap\,\mathbf{C}\,pa).$$

□

The relation $\mathbf{C}$ may be characterized by the following

Property 13 For traces s and t over A and symbols $a \in I$ and $p \in O$, we have

$$\begin{aligned}
\varepsilon\,\mathbf{C}\,\varepsilon &= \text{true},\\
sa\,\mathbf{C}\,t &= s\,\mathbf{C}\,t \;\wedge\; \#_a s < \#_a t,\\
s\,\mathbf{C}\,tp &= s\,\mathbf{C}\,t \;\wedge\; \#_p s > \#_p t,\\
sp\,\mathbf{C}\,\varepsilon &= s\,\mathbf{C}\,\varepsilon,\\
\varepsilon\,\mathbf{C}\,ta &= \varepsilon\,\mathbf{C}\,t,\\
sp\,\mathbf{C}\,ta &= sp\,\mathbf{C}\,t \;\vee\; s\,\mathbf{C}\,ta.
\end{aligned}$$

□

Without proof we state

Property 14 The relation $\mathbf{C}$ is a pre-order, that is, it is reflexive and transitive. □

The reader can find more details about the relation $\mathbf{C}$ in [4]. For later use we also mention

Property 15 For traces s, t, and u, and symbol $a \in I$ we have

$$st\,\mathbf{C}\,u \;\Rightarrow\; (\exists\, u_0 : u_0 \leq u : s\,\mathbf{C}\,u_0)$$

and

$$ta\,\mathbf{C}\,u \;\Rightarrow\; (\exists\, b, u_0, u_1 : b \in I \wedge u = u_0 b u_1 : t\,\mathbf{C}\,u_0 b).$$

□

Of course, by I-O-duality of $\mathbf{C}$ we also have

$$s\,\mathbf{C}\,tu \;\Rightarrow\; (\exists\, s_0 : s_0 \leq s : s_0\,\mathbf{C}\,t).$$

The relation $\mathbf{C}$ can be used to characterize the safe reachable game vectors of other games besides the (A^*, A^*)-game.

Property 16 Let S and T be specifications in $\mathcal{T}$. In the asynchronous (S,T)-game, safe game vector (s,t), i.e. such that $s \in S$ and $t \in T$, is reachable if and only if $s\,\mathbf{C}\,t$ holds. □

5.1 Absence of Computation Interference

Now we can relate **nai** to $\sqsubseteq$. From the non-emptiness of specifications and Properties 1, 13, and 16 we derive

Lemma 5 For specifications S and T in $\mathcal{T}$, $S\,\mathbf{nai}\,T$ holds if and only if

$$(\forall\, s,t,p : s \in S \wedge t \in T \wedge p \in O \wedge s\,\mathbf{C}\,tp : tp \in T) \wedge$$
$$(\forall\, s,t,a : s \in S \wedge t \in T \wedge a \in I \wedge sa\,\mathbf{C}\,t : sa \in S).$$

□

From the reflexivity of $\mathbf{C}$ we immediately infer

Property 17 For S and T in $\mathcal{T}$ we have

$$S\,\mathbf{nai}\,T \;\Rightarrow\; S \sqsubseteq T.$$

□

Corollary 2 Relation **nai** is antisymmetric. □

A generalization of Lemma 2 is

Lemma 6 For S, T, and U in $\mathcal{T}$ such that S **nai** T and $T \sqsubseteq U$, and traces $s \in S$ and $u \in U$ with $s \mathbf{C} u$ we have $u \in T$.

Proof By mathematical induction on the length of u. Assume S **nai** T and $T \sqsubseteq U$.

Base: $u = \varepsilon$. Then $u \in T$, since T is non-empty and prefix-closed.

Step: $u = u_0 p$ for $p \in O$. We derive

$$
\begin{array}{ll}
 & s \in S \wedge s \mathbf{C} u \wedge u \in U \\
= & \quad \{\ u = u_0 p, \text{ Property 13 using } p \in O, \text{ and } U \text{ is prefix-closed}\ \} \\
 & s \in S \wedge s \mathbf{C} u_0 p \ \wedge \ s \mathbf{C} u_0 \wedge u_0 \in U \\
\Rightarrow & \quad \{\ \text{induction hypothesis, using } \ell(u_0) < \ell(u)\ \} \\
 & s \in S \wedge s \mathbf{C} u_0 p \ \wedge \ u_0 \in T \\
\Rightarrow & \quad \{\ S \mathbf{nai} T \text{ and Lemma 5 using } p \in O\ \} \\
 & u_0 p \in T \\
= & \quad \{\ u = u_0 p\ \} \\
 & u \in T
\end{array}
$$

Step: $u = u_0 a$ for $a \in I$. We derive

$$
\begin{array}{ll}
 & s \in S \wedge s \mathbf{C} u \wedge u \in U \\
= & \quad \{\ u = u_0 a\ \} \\
 & s \in S \wedge s \mathbf{C} u_0 a \wedge u_0 a \in U \\
\Rightarrow & \quad \{\ \text{Property 15 using } a \in I, \text{ and } S \text{ and } U \text{ are prefix-closed}\ \} \\
 & (\exists\, s_0 : s_0 \leq s : s_0 \in S \wedge s_0 \mathbf{C} u_0 \wedge u_0 \in U) \wedge u_0 a \in U \\
\Rightarrow & \quad \{\ \text{induction hypothesis, using } \ell(u_0) < \ell(u)\ \} \\
 & u_0 \in T \wedge u_0 a \in U \\
\Rightarrow & \quad \{\ T \sqsubseteq U, \text{ in particular } U \subseteq_I T \text{ using } a \in I\ \} \\
 & u_0 a \in T \\
= & \quad \{\ u = u_0 a\ \} \\
 & u \in T
\end{array}
$$

□

We can now prove the following substitution theorem.

Theorem 7 For S, T, and U in $\mathcal{T}$ we have

$$S \,\mathbf{nai}\, T \;\wedge\; T \sqsubseteq U \;\Rightarrow\; S \,\mathbf{nai}\, U.$$

Proof Assume $S \,\mathbf{nai}\, T$ and $T \sqsubseteq U$. We prove $S \,\mathbf{nai}\, U$ using Lemma 5. Let $s \in S$ and $u \in U$. We now derive

$$\begin{array}{ll} & p \in O \wedge s \,\mathbf{C}\, up \\ = & \quad \{ \text{ Property 13 } \} \\ & p \in O \wedge s \,\mathbf{C}\, up \wedge s \,\mathbf{C}\, u \\ \Rightarrow & \quad \{ \text{ Lemma 6, using } s \in S \text{ and } u \in U \} \\ & p \in O \wedge s \,\mathbf{C}\, up \wedge u \in T \\ \Rightarrow & \quad \{ S \,\mathbf{nai}\, T \text{ using } s \in S \} \\ & p \in O \wedge up \in T \\ \Rightarrow & \quad \{ T \sqsubseteq U, \text{ in particular } T \subseteq_O U, \text{ using } u \in U \} \\ & up \in U \end{array}$$

and

$$\begin{array}{ll} & a \in I \wedge sa \,\mathbf{C}\, u \\ = & \quad \{ \text{ Property 13 } \} \\ & a \in I \wedge sa \,\mathbf{C}\, u \wedge s \,\mathbf{C}\, u \\ \Rightarrow & \quad \{ \text{ Lemma 6, using } s \in S \text{ and } u \in U \} \\ & a \in I \wedge sa \,\mathbf{C}\, u \wedge u \in T \\ \Rightarrow & \quad \{ S \,\mathbf{nai}\, T \text{ using } s \in S \} \\ & sa \in S \end{array}$$

□

Corollary 3 For S, T, and U in $\mathcal{T}$ we have

$$\begin{array}{rcl} S \sqsubseteq T \;\wedge\; T \,\mathbf{nai}\, U & \Rightarrow & S \,\mathbf{nai}\, U \\ S \,\mathbf{nai}\, T \;\wedge\; T \,\mathbf{nai}\, U & \Rightarrow & S \,\mathbf{nai}\, U \end{array}$$

Proof The first implication follows from Theorem 7 by exploiting the fact that interchanging the role of I and O gives the dual relations for both $\sqsubseteq$ and **nai**. The second implication, expressing the transitivity of **nai**, follows by applying Property 17. □

5.2 Satisfaction and Equivalence

We recall the definition of **sat** for the asynchronous two-person game:

$$S \,\mathbf{sat}\, T \;=\; (\forall U : T \,\mathbf{nai}\, U : S \,\mathbf{nai}\, U).$$

Property 18 For S, T, and U in $\mathcal{T}$ we have

$$\begin{aligned} S \,\mathbf{sat}\, T \;\wedge\; T \,\mathbf{nai}\, U \;&\Rightarrow\; S \,\mathbf{nai}\, U \\ S \,\mathbf{sat}\, T \;\wedge\; T \sqsubseteq U \;&\Rightarrow\; S \,\mathbf{sat}\, U \\ S \sqsubseteq T \;\wedge\; T \,\mathbf{sat}\, U \;&\Rightarrow\; S \,\mathbf{sat}\, U \end{aligned}$$

□

Example 1 In general, we do not have

$$S \,\mathbf{nai}\, T \;\wedge\; T \,\mathbf{sat}\, U \;\Rightarrow\; S \,\mathbf{nai}\, U.$$

Take, for instance, $I = \{a, b\}$, $O = \emptyset$, $S = T = \{\varepsilon, a\}$, and $U = \{\varepsilon, a, ab\}$. In this case we do not even have

$$S \,\mathbf{nai}\, T \;\wedge\; T \,\mathbf{sat}\, U \;\Rightarrow\; S \sqsubseteq U.$$

Of course, we do have

$$S \,\mathbf{nai}\, T \;\wedge\; T \,\mathbf{sat}\, U \;\Rightarrow\; S \,\mathbf{sat}\, U.$$

□

Property 19 For S and T in $\mathcal{T}$ we have

$$S \sqsubseteq T \;\Rightarrow\; S \,\mathbf{sat}\, T.$$

Proof Assuming $S \sqsubseteq T$ we derive for $U \in \mathcal{T}$

$$\begin{aligned} & T \,\mathbf{nai}\, U \\ \Rightarrow \quad & \{\ S \sqsubseteq T \text{ and Theorem 7 } \} \\ & S \,\mathbf{nai}\, U \end{aligned}$$

hence, $S \,\mathbf{sat}\, T$. □

Property 20 For subset X of $\mathcal{T}$ and $U \in \mathcal{T}$ we have

$$(\forall S : S \in X : S \,\mathbf{nai}\, U) \;\Rightarrow\; lub.X \,\mathbf{nai}\, U.$$

Proof Let $T = lub.X$, then $T \in \mathcal{T}$. Assuming the left-hand side of the implication we show T **nai** U using Lemma 5. Let $t \in T$ and $u \in U$. In case X is non-empty then on account of Property 10 and $t \in T$, let $S \in X$ with $t \in S$. For non-empty X we now derive for symbol p

$$
\begin{array}{ll}
 & p \in O \wedge t \mathbf{C} up \\
\Rightarrow & \quad \{\ S \,\mathbf{nai}\, U \text{ by assumption, using } t \in S \text{ and } u \in U\ \} \\
 & up \in U
\end{array}
$$

If X is empty then $T = I^*$ and, hence, for $p \in O$ we cannot have $t \mathbf{C} up$. Furthermore, we derive for symbol a

$$
\begin{array}{ll}
 & a \in I \wedge ta \mathbf{C} u \\
\Rightarrow & \quad \{\ \text{assumption, using } u \in U\ \} \\
 & (\forall S : S \in X \wedge t \in S : ta \in S) \\
\Rightarrow & \quad \{\ \text{definition of } lub \text{ and } T, \text{ using } t \in T\ \} \\
 & ta \in T
\end{array}
$$

□

Corollary 4 For subset X of $\mathcal{T}$ we have

$$(\forall S : S \in X : S \,\mathbf{nai}\, S) \;\Rightarrow\; lub.X \,\mathbf{nai}\, lub.X.$$

Proof Let $T = lub.X$. We derive

$$
\begin{array}{ll}
 & (\forall S : S \in X : S \,\mathbf{nai}\, S) \\
= & \quad \{\ T = lub.X\ \} \\
 & (\forall S : S \in X : S \,\mathbf{nai}\, S \wedge S \sqsubseteq T) \\
\Rightarrow & \quad \{\ \text{Theorem 7}\ \} \\
 & (\forall S : S \in X : S \,\mathbf{nai}\, T) \\
\Rightarrow & \quad \{\ \text{Property 20}\ \} \\
 & lub.X \,\mathbf{nai}\, T \\
= & \quad \{\ \text{definition of } T\ \} \\
 & lub.X \,\mathbf{nai}\, lub.X
\end{array}
$$

□

5.3 The Subspace D_4

In this subsection we introduce the subspace D_4 of delay-insensitive specifications and we show how it relates to the equivalence classes.

Definition 23 The set of specifications D_4 is defined by

$$D_4 = \{S : S \in \mathcal{T} \wedge S \,\mathbf{nai}\, S : S\}.$$

□

The above Corollary can now be rephrased as: if X is a subset of D_4, then $lub.X \in D_4$. In a similar way (because of the I-O-duality of **nai**) one can prove that $glb.X \in D_4$. Hence, D_4 is a complete sublattice of $\mathcal{T}$.

There is an important relationship between the Rules given in Section 3 and membership in D_4, viz. $S \in D_4$ holds if and only if S satisfies Rules $\mathbf{R_3}$, $\mathbf{R_4}$, and $\mathbf{R_5}$. This is sometimes referred to as *The Fundamental Theorem of Delay-Insensitive Specifications* (also see [3,4]).

The equivalence class containing specification $S \in \mathcal{T}$ is denoted by $[S]$, that is,

$$[S] = \{U : U \in \mathcal{T} \wedge U \,\mathbf{equ}\, S : U\}.$$

We first show that each equivalence class contains at most one specification from D_4.

Property 21 For S and T in $\mathcal{T}$ we have

$$S \,\mathbf{equ}\, T \;\wedge\; S \in D_4 \;\wedge\; T \in D_4 \;\Rightarrow\; S = T.$$

Proof We derive

$$\begin{array}{ll} & S \,\mathbf{equ}\, T \wedge S \in D_4 \wedge T \in D_4 \\ = & \quad \{ \text{ Property of } \mathbf{equ} \text{ and definition of } D_4 \ \} \\ & (\forall\, U :: S \,\mathbf{nai}\, U = T \,\mathbf{nai}\, U) \wedge S \,\mathbf{nai}\, S \wedge T \,\mathbf{nai}\, T \\ \Rightarrow & \quad \{ \text{ predicate calculus } \} \\ & S \,\mathbf{nai}\, T \wedge T \,\mathbf{nai}\, S \\ \Rightarrow & \quad \{ \text{ antisymmetry of } \mathbf{nai} \ \} \\ & S = T \end{array}$$

□

Now we show that each equivalence class has a $\sqsubseteq$-maximum.

Property 22 For $S \in \mathcal{T}$ we have

$$lub.[S] \in [S].$$

Proof Let $T = lub.[S]$. We show $S \,\mathbf{equ}\, T$, hence $T \in [S]$. From $S \in [S]$ and the definition of least upper bound we infer $S \sqsubseteq T$. Using Property 17 then gives $S \,\mathbf{sat}\, T$. It remains to prove $T \,\mathbf{sat}\, S$. Let V be such that $S \,\mathbf{nai}\, V$ holds. Then for all $U \in [S]$ we have $U \,\mathbf{nai}\, V$ and, hence, Property 20 yields $T \,\mathbf{nai}\, V$, which completes the proof for $T \,\mathbf{sat}\, S$. □

The $\sqsubseteq$-maximum of an equivalence class can serve as a canonical representative. Next we want to show that this maximum is in D_4. Property 21 then implies that it is the only D_4 member of an equivalence class. First we need two more lemmata. Both lemmata construct a specification that is equivalent to and strictly $\sqsubseteq$-greater than a given specification not in D_4.

Lemma 7 Let $T \in \mathcal{T}$, $p \in O$, $t \in T$, $u \in T$ such that $up \notin T$ and $t \,\mathbf{C}\, up$. Define T' by

$$T' = T \cup \{v : v \in I^* : upv\}.$$

Then we have

$$T \neq T' \;\wedge\; T \sqsubseteq T' \;\wedge\; T \,\mathbf{equ}\, T'.$$

Proof The first conjunct is implied by $up \notin T$ and $up \in T'$. From $T \subseteq T'$ follows $T \subseteq_O T'$. For trace s and symbol $a \in I$ we derive

$$\begin{array}{ll} & sa \in T' \wedge s \in T \\ = & \quad \{ \text{ definition of } T' \ \} \\ & (sa \in T \vee (\exists\, v : v \in I^* : sa = upv)) \wedge s \in T \\ = & \quad \{ \ a \neq p \text{ because } a \in I,\ p \in O, \text{ and } I \cap O = \emptyset \ \} \\ & (sa \in T \vee (\exists\, v : v \in I^* : s = upv)) \wedge s \in T \\ \Rightarrow & \quad \{ \ up \notin T \text{ and } T \text{ is prefix-closed} \ \} \\ & (sa \in T \vee s \notin T) \wedge s \in T \\ \Rightarrow & \quad \{ \text{ predicate calculus } \} \\ & sa \in T \end{array}$$

This shows $T' \subseteq_I T$, which completes the proof of $T \sqsubseteq T'$. Hence, we know from Property 19 that $T \,\mathbf{sat}\, T'$ holds. All that remains to be shown for $T \,\mathbf{equ}\, T'$ is $T' \,\mathbf{sat}\, T$. Assuming $T \,\mathbf{nai}\, U$ holds for some $U \in \mathcal{T}$ we show that $T' \,\mathbf{nai}\, U$ holds as well using Lemma 5. Let $s \in T'$ and $w \in U$.

First we consider the case with symbol $q \in O$ such that $s \,\mathbf{C}\, wq$, and we want to show $wq \in U$. If $s \in T$ then $wq \in U$ on account of $T \,\mathbf{nai}\, U$. Otherwise, $s = upv$ for some $v \in I^*$. Property 13 tells us that $up \,\mathbf{C}\, upv$ and, hence, we have

$$t \,\mathbf{C}\, up \,\mathbf{C}\, upv \,\mathbf{C}\, wq.$$

From the transitivity of $\mathbf{C}$ we then infer $t \,\mathbf{C}\, wq$ and, therefore, $wq \in U$ again on account of $T \,\mathbf{nai}\, U$.

Finally, we consider the case with symbol $a \in I$ such that $sa \,\mathbf{C}\, w$, and we want to show $sa \in T'$. If $s \in T$, then $sa \in T$ on account of $T \,\mathbf{nai}\, U$, and thus $sa \in T'$. Otherwise, $s = upv$ for some $v \in I^*$. Hence, $sa = upva$, which is in T' by definition of T', a being an input symbol.

This completes the proof of $T' \,\mathbf{nai}\, U$ and thus that of $T' \,\mathbf{sat}\, T$. □

Lemma 8 Let $T \in \mathcal{T}$, $a \in I$, $t \in T$, $u \in T$ such that $ta \notin T$ and $ta \,\mathbf{C}\, u$. On account of Property 15 we can write $u = u_0 b u_1$ for some $b \in I$ such that $ta \,\mathbf{C}\, u_0 b$. Define T' by

$$T' = T \setminus \{v : v \in A^* : u_0 b v\}.$$

Then we have

$$T \neq T' \;\wedge\; T \sqsubseteq T' \;\wedge\; T \,\mathbf{equ}\, T'.$$

Proof The first conjunct is implied by $u_0 b \in T$ and $u_0 b \notin T'$. From $T' \subseteq T$ follows $T' \subseteq_I T$. For trace s and symbol $p \in O$ we derive

$$\begin{aligned}
& sp \in T \wedge s \in T' \\
= \quad & \{ \text{ definition of } T' \} \\
& sp \in T \wedge s \in T \wedge \neg(u_0 b \leq s) \\
= \quad & \{ \ T \text{ is prefix-closed } \} \\
& sp \in T \wedge \neg(u_0 b \leq s) \\
= \quad & \{ \ b \neq p \text{ because } b \in I, p \in O, \text{and } I \cap O = \emptyset \ \} \\
& sp \in T \wedge \neg(u_0 b \leq sp) \\
= \quad & \{ \text{ definition of } T' \}
\end{aligned}$$

$$sp \in T'$$

This shows $T \subseteq_O T'$, which completes the proof of $T \sqsubseteq T'$. Hence, we know from Property 19 that $T \,\mathbf{sat}\, T'$ holds. All that remains to be shown for $T \,\mathbf{equ}\, T'$ is $T' \,\mathbf{sat}\, T$. Assuming $T \,\mathbf{nai}\, U$ holds for some $U \in \mathcal{T}$ we show that $T' \,\mathbf{nai}\, U$ holds as well using Lemma 5. Let $s \in T'$ and $w \in U$.

First we consider the case with symbol $p \in O$ such that $s \,\mathbf{C}\, wp$, and we want to show $wp \in U$. Since $T' \subseteq T$, we have $s \in T$. From $T \,\mathbf{nai}\, U$ now follows $wp \in U$.

Finally, we consider the case with symbol $c \in I$ such that $sc \,\mathbf{C}\, w$, and we want to show $sc \in T'$. Since $T' \subseteq T$, we have $s \in T$. From $T \,\mathbf{nai}\, U$ now follows $sc \in T$. It remains to show that $\neg(u_0 b \leq sc)$. Assuming $u_0 b \leq sc$ we derive a contradiction. On account of this assumption, $sc \,\mathbf{C}\, w$, and Property 15 let $w_0 \leq w$ be such that $u_0 b \,\mathbf{C}\, w_0$. We now have $ta \,\mathbf{C}\, u_0 b \,\mathbf{C}\, w_0$ and, thus, $ta \,\mathbf{C}\, w_0$. From $t \in T$, $w_0 \in U$, and $T \,\mathbf{nai}\, U$ then follows $ta \in T$, contradicting $ta \notin T$.

This completes the proof of $T' \,\mathbf{nai}\, U$ and thus that of $T' \,\mathbf{sat}\, T$. □

Theorem 8 For all $S \in \mathcal{T}$ we have $lub.[S] \in D_4$.

Proof Let $T = lub.[S]$, we show $T \,\mathbf{nai}\, T$ using Lemma 5. Let $t \in T$ and $u \in T$. First we consider the case with symbol $p \in O$ such that $t \,\mathbf{C}\, up$, and we show $up \in T$. Assuming $up \notin T$ leads to a contradiction as follows. According to Lemma 7, which is now applicable, let $T' \in \mathcal{T}$ be such that

$$T \neq T' \wedge T \sqsubseteq T' \wedge T \,\mathbf{equ}\, T'.$$

From Property 22 we know $S \,\mathbf{equ}\, T$ and, thus, $T' \in [S]$. Since T is an upper bound of $[S]$ we have $T' \sqsubseteq T$. Combined with $T \sqsubseteq T'$ this yields $T = T'$, contradicting $T \neq T'$.

The case with symbol $a \in I$ such that $ta \,\mathbf{C}\, u$ is similar using Lemma 8. □

Corollary 5 For $S \in \mathcal{T}$ we have $S \,\mathbf{nai}\, lub.[S]$.

Proof From $S \in [S]$ and the definition of least upper bound we infer $S \sqsubseteq lub.[S]$. Theorem 8 tells us that $lub.[S] \,\mathbf{nai}\, lub.[S]$. Application of Corollary 3 now yields $S \,\mathbf{nai}\, lub.[S]$. □

The following theorem gives a different expression for the maximum of an equivalence class.

Theorem 9 For $S \in \mathcal{T}$ let $T = glb.\{U : S \,\mathbf{nai}\, U : U\}$. Then we have

$$S \,\mathbf{nai}\, T \;\wedge\; S \,\mathbf{equ}\, T \;\wedge\; T \in D_4$$

and, hence, $T = lub.[S]$.

Proof Let $T' = lub.[S]$. In the light of the preceding Theorem and its Corollary it is sufficient to show $T = T'$. We do so by proving that T' is the greatest lower bound of $\{U : S \,\mathbf{nai}\, U : U\}$. For $U \in \mathcal{T}$ we derive

$$\begin{array}{ll} & S \,\mathbf{nai}\, U \\ = & \quad \{\ S \,\mathbf{equ}\, T', \text{ on account of Theorem 8 }\} \\ & T' \,\mathbf{nai}\, U \\ \Rightarrow & \quad \{\ \text{Property 17 }\} \\ & T' \sqsubseteq U \end{array}$$

and, hence, T' is a lower bound. Now we derive for $V \in \mathcal{T}$

$$\begin{array}{ll} & (\forall\, U : S \,\mathbf{nai}\, U : V \sqsubseteq U) \\ \Rightarrow & \quad \{\ \text{instantiation, using } S \,\mathbf{nai}\, T' \text{ from Corollary 5 }\} \\ & V \sqsubseteq T' \end{array}$$

and, hence, T' is greatest among the lower bounds. □

Corollary 6 For $S \in \mathcal{T}$ let $T = glb.\{U : U \in D_4 \wedge S \sqsubseteq U : U\}$. Then we have

$$S \,\mathbf{nai}\, T \;\wedge\; S \,\mathbf{equ}\, T \;\wedge\; T \in D_4$$

and, hence, $T = lub.[S]$. □

We can now give a number of equivalent characterizations of D_4.

Theorem 10 For $S \in \mathcal{T}$ the following expressions are equivalent:

0. $S \in D_4$,
1. $(\forall\, U :: S \,\mathbf{nai}\, U = S \sqsubseteq U)$,
2. $(\forall\, U :: U \,\mathbf{nai}\, S = U \sqsubseteq S)$,
3. $(\forall\, U :: S \,\mathbf{nai}\, U = S \,\mathbf{sat}\, U)$,
4. $(\forall\, U :: U \,\mathbf{nai}\, S = U \,\mathbf{sat}\, S)$,
5. $(\forall\, U :: U \,\mathbf{sat}\, S = U \sqsubseteq S)$,

6. $S = lub.[S]$,
7. $S = glb.\{U : S \,\mathbf{nai}\, U : U\}$,
8. $glb.\{U : S \,\mathbf{nai}\, U : U\} = lub.\{U : U \,\mathbf{nai}\, S : U\}$,
9. $lub.[S] = glb.\{T : (\forall U : U \,\mathbf{nai}\, S = U \,\mathbf{nai}\, T) : T\}$.

□

See Example 1 for a reason why

$$(\forall U :: S \,\mathbf{sat}\, U = S \sqsubseteq U)$$

is missing among the expressions in the Theorem. The last two expressions in the Theorem derive from an analogous investigation of the relations $\mathbf{sat}'$ and $\mathbf{equ}'$ defined by

$$\begin{aligned} S \,\mathbf{sat}'\, T &= (\forall U : U \,\mathbf{nai}\, S : U \,\mathbf{nai}\, T) \\ S \,\mathbf{equ}'\, T &= S \,\mathbf{sat}'\, T \wedge T \,\mathbf{sat}'\, S \end{aligned}$$

for all S and T in $\mathcal{T}$.

Corollary 7 On D_4 the relations **nai**, **sat**, and $\sqsubseteq$ are all the same. □

The Corollary is a special case of what is sometimes called *The Fundamental Theorem of Delay-Insensitive Composition*, viz. that for delay-insensitive specifications absence of computation interference in the asynchronous and in the synchronous game are the same.

Theorem 11 The structures $\langle \mathcal{T}, \mathbf{sat} \rangle / \mathbf{equ}$, $\langle \mathcal{T}, \mathbf{nai} \rangle / \mathbf{equ}$, and $\langle D_4, \sqsubseteq \rangle$ are isomorphic. □

Note 1 The treatment of the empty specification is still unsatisfactory. When $\emptyset$ is included in the present framework it would end up in an equivalence class by its self and it would not belong to D_4. The reason for this is that

$$(\forall S : S \in \mathcal{T} \cup \{\emptyset\} : \neg(\emptyset \,\mathbf{nai}\, S))$$

and, hence, $\neg(\emptyset \,\mathbf{nai}\, \emptyset)$, i.e. $\emptyset \notin D_4$; and also $\neg(\emptyset \,\mathbf{nai}\, O^*)$. Furthermore,

$$(\forall S : S \in \mathcal{T} \wedge S \neq \emptyset : S \,\mathbf{nai}\, O^*)$$

and, hence, $\neg(S \,\mathbf{equ}\, \emptyset)$ for $S \neq \emptyset$. Thus, $\emptyset$ would require a special treatment in Theorem 8. There is, however, a case to be made for including the empty specification in D_4 anyway, but we could not find a natural way to do so. □

6 Composition

In this section we briefly touch upon the issue of synchronous parallel composition of DTS's. We define the composite of two DTS's and show two monotonicity theorems.

By composing two DTS's, they communicate with each other via their common symbols and with the environment via the non-common symbols. The common symbols should be of type input in the one and of type output in the other DTS, so we can add a third DTS, the environment, to play the game on a closed collection. Absence of interference being our correctness criterion, we are interested in what signals an environment of the composite can maximally send without causing interference, neither in the communications with the composite nor in the communications between the components. Moreover, we are interested in the outputs that such an environment should minimally be prepared for in order to guarantee absence of interference.

It is clear that we can no longer assume the alphabets of the trace structures under consideration to be equal and to have the same input and output partitioning as we did in the previous sections. we extend the $\sqsubseteq$-relation to the space of all trace structures by defining it to be false whenever two trace structures differ in their input or output alphabets.

Definition 24 The synchronous parallel composition of two trace structures S and T, $S \parallel T$, is defined as

$$glb.\{U : \mathbf{nsi}\{S,T,U\} : \tilde{U}\}.$$

□

From this definition, involving three trace structures, we see that we should gain insight in the three-person game before actually trying to attack the problem of composition. The area of the three-person game is unexplored and we have not yet had the opportunity to study this game. Since we want to give the reader some feel for the issues involved we prove two, rather arbitrary, theorems, one having to do with the three-person game and the other one with composition.

In the next theorem we prove some sort of monotonicity property of the weave.

Theorem 12 For DTS's S_0, S_1, T, and U such that $S_0 \sqsubseteq S_1 \wedge \mathbf{nsi}\{S_1,T,U\}$ we have

$$S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \subseteq S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U.$$

Proof First of all, notice that $S_0 \sqsubseteq S_1$ implies that the input alphabets and output alphabets of S_0 and S_1 are equal. Since $\mathbf{nsi}\{S_1, T, U\}$ holds, we therefore have that $\{S_0, T, U\}$ is closed. We prove the theorem by induction on the length of the traces in $S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U$.

Base: This case is obvious, since $\mathbf{nsi}\{S_1, T, U\}$ implies that ε is in all the trace structures and, hence, in $\varepsilon \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U$.

Step: In the induction step we assume the theorem to hold for traces of a certain length and prove it to hold for traces that are extended by one symbol a. We distinguish three cases: $a \notin \mathbf{a}S_0$, $a \in \mathbf{i}S_0$, and $a \in \mathbf{o}S_0$.

In the case that $a \notin \mathbf{a}S_0$ we derive

$$\begin{array}{ll}
& sa \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \\
\Rightarrow & \quad \{ \text{ weaving preserves prefix-closedness, definition of weaving } \} \\
& s \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \wedge a \in \mathbf{a}S_0 \cup \mathbf{a}T \cup \mathbf{a}U \\
\Rightarrow & \quad \{ \text{ induction hypothesis } \} \\
& s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \wedge a \in \mathbf{a}S_0 \cup \mathbf{a}T \cup \mathbf{a}U \\
\Rightarrow & \quad \{ \text{ definition of weaving, } a \notin \mathbf{a}S_0,\ \mathbf{a}S_0 = \mathbf{a}S_1 \ \} \\
& s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (sa)\lceil \mathbf{a}S_1 \in S_1 \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \wedge \\
& a \in \mathbf{a}S_1 \cup \mathbf{a}T \cup \mathbf{a}U \\
= & \quad \{ \text{ definition of weaving } \} \\
& sa \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U
\end{array}$$

In the case that $a \in \mathbf{i}S_0$ we derive

$$\begin{array}{ll}
& sa \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \\
\Rightarrow & \quad \{ \text{ weaving preserves prefix-closedness, definition of weaving } \} \\
& s \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \wedge a \in \mathbf{a}S_0 \cup \mathbf{a}T \cup \mathbf{a}U \\
\Rightarrow & \quad \{ \text{ induction hypothesis, } \mathbf{a}S_0 = \mathbf{a}S_1 \ \} \\
& s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \wedge a \in \mathbf{a}S_1 \cup \mathbf{a}T \cup \mathbf{a}U \\
\Rightarrow & \quad \{ \ a \in \mathbf{i}S_0, \text{ hence, } a \in \mathbf{o}T \cup \mathbf{o}U, \text{ since } \mathbf{cl}\{S_0, T, U\}; \text{ moreover,} \\
& \qquad \mathbf{nsi}\{S_1, T, U\} \ \} \\
& s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (sa)\lceil \mathbf{a}S_1 \in S_1 \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \wedge \\
& a \in \mathbf{a}S_1 \cup \mathbf{a}T \cup \mathbf{a}U \\
= & \quad \{ \text{ definition of weaving } \}
\end{array}$$

$$sa \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U$$

In the case that $a \in \mathbf{o}S_0$ we derive

$$\begin{array}{ll} & sa \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \\ \Rightarrow & \{ \text{ definition of weaving, weaving preserves prefix-closedness, and } a \in \mathbf{a}S_0 \ \} \\ & s \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}S_0)a \in S_0 \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \\ \Rightarrow & \{ \text{ induction hypothesis, } \mathbf{a}S_0 = \mathbf{a}S_1 \ \} \\ & s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}S_1)a \in S_0 \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \\ \Rightarrow & \{ \ S_0 \sqsubseteq S_1, \ a \in \mathbf{o}S_0, \text{ and } s\lceil \mathbf{a}S_1 \in S_1 \text{ by the definition of weaving } \} \\ & s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}S_1)a \in S_1 \wedge (sa)\lceil \mathbf{a}T \in T \wedge (sa)\lceil \mathbf{a}U \in U \\ = & \{ \text{ definition of weaving, using } a \in \mathbf{a}S_1 \ \} \\ & sa \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \end{array}$$

□

In the next theorem we prove that $\|$ is monotonic with respect to $\sqsubseteq$.

Theorem 13 For *DTS*'s S_0, S_1, and T such that $S_0 \sqsubseteq S_1$ we have $S_0 \parallel T \sqsubseteq S_1 \parallel T$.

Proof We prove that

$$\{U : \mathbf{nsi}\{S_1, T, U\} : U\} \subseteq \{U : \mathbf{nsi}\{S_0, T, U\} : U\}.$$

Hence, the greatest lower bound of the latter is at most ($\sqsubseteq$) that of the former set. This inclusion also holds when we reflect the trace structures that we collect and switch to the dual ordering. This then establishes the proof. Let *DTS* U be such that

$$\mathbf{nsi}\{S_1, T, U\} \tag{0}$$

Since the alphabets of S_0 and S_1 are equal and partitioned in the same way, we have $\mathbf{cl}\{S_0, T, U\}$. We show that $\{S_0, T, U\}$ has absence of computation interference by showing (cf. Property 1

$$\begin{array}{l} (\forall V, W, s, a : V, W \in \{S_0, T, U\} \wedge a \in \mathbf{o}V \cap \mathbf{i}W \wedge \quad s \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U : \\ \qquad (s\lceil \mathbf{a}V)a \in V \Rightarrow (s\lceil \mathbf{a}W)a \in W) \end{array} \tag{1}$$

Because of the symmetry between T and U, it suffices to instantiate the pair of dummies (V, W) by (S_0, T), (T, S_0), and (T, U).

Instantiating V and W by S_0 and T respectively we derive

$a \in \mathbf{o}S_0 \cap \mathbf{i}T \wedge s \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}S_0)a \in S_0$

$\Rightarrow$ { Theorem 12, using $s_0 \sqsubseteq S_1$, and (0) }

$a \in \mathbf{o}S_0 \cap \mathbf{i}T \wedge s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}S_0)a \in S_0$

$\Rightarrow$ { $\mathbf{a}S_0 = \mathbf{a}S_1$, $\mathbf{o}S_0 = \mathbf{o}S_1$, and $S_0 \sqsubseteq S_1$, $s\lceil \mathbf{a}S_1 \in S_1$ on account of the definition of weaving }

$a \in \mathbf{o}S_1 \cap \mathbf{i}T \wedge s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}S_1)a \in S_1$

$\Rightarrow$ { (0), using Property 1 }

$(s\lceil \mathbf{a}T)a \in T$

Instantiating V and W by T and S_0 respectively we derive

$a \in \mathbf{o}T \cap \mathbf{i}S_0 \wedge s \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}T)a \in T$

$\Rightarrow$ { $\mathbf{i}S_0 = \mathbf{i}S_1$, definition of weaving, Theorem 12, using $S_0 \sqsubseteq S_1$ and (0) }

$a \in \mathbf{o}T \cap \mathbf{i}S_1 \wedge (s\lceil \mathbf{a}S_0) \in S_0 \wedge s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}T)a \in T$

$\Rightarrow$ { (0) using Property 1 }

$(s\lceil \mathbf{a}S_1)a \in S_1 \wedge a \in \mathbf{i}S_1 \wedge s\lceil \mathbf{a}S_0 \in S_0$

$\Rightarrow$ { $\mathbf{a}S_0 = \mathbf{a}S_1$, $S_0 \sqsubseteq S_1$ }

$(s\lceil \mathbf{a}S_0)a \in S_0$

Finally, we instantiate V and W by T and U and we derive

$a \in \mathbf{o}T \cap \mathbf{i}U \wedge s \in S_0 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}T)a \in T$

$\Rightarrow$ { Theorem 12 }

$a \in \mathbf{o}T \cap \mathbf{i}U \wedge s \in S_1 \,\mathbf{w}\, T \,\mathbf{w}\, U \wedge (s\lceil \mathbf{a}T)a \in T$

$\Rightarrow$ { (0), using Property 1 }

$(s\lceil \mathbf{a}U)a \in U$

□

7 Concluding Remarks

In this section we summarize the results of the previous sections, relate it to some other work in this area, and mention future research.

Given the space of all *DTS*'s as our primitive objects, we have discussed a general approach to suppress irrelevant distinctions between objects in the presence of particular correctness concerns. We have demonstrated this by

choosing computation interference as our correctness concern, which led to the introduction of the $\sqsubseteq$-relation. The mathematical framework associated with this ordering turned out to be so rich that we touched only upon a small number of issues: the two-person games, the relation between the equivalence classes introduced by $\sqsubseteq$, delay-insensitivity, and, very superficially, composition.

The important distinction between our work and CSP [1], is that our communications are directed. We believe that directed (asynchronous) communications, as opposed to the synchronous communications in CSP, provide a mathematically tractable, rich and formal basis as a first abstraction of the operation of electrical circuits. This work supports this point of view and one of the next research topics is to see how well this point of view can be upheld when other correctness concerns play a role.

Additional correctness concerns in which we are interested are transmission interference, progress (e.g. liveness and deadlock), and fairness. Preliminary research in these areas looks promising, but it is too early to report on it. When progress is one of our additional concerns, we introduce equivalence classes of a finer grain than with only the concern of computation interference. It is unlikely in this case that we can still choose one canonical representative in the class capturing all properties of that class. Presumably, we have to extend our specifications with additional information, each new specification then representing one class. This is similar to what one does when introducing the integers from the natural numbers, or when extending the trace model to the failures model as in [2].

An issue which has been brought up, but which has not been resolved very satisfactorily is that of the empty DTS. The need for it is clear. Not allowing DTS's to be empty, there are two distinct ways in which the DTS with trace set O^* comes about. On the one hand it represents a composite that will exhibit no computation interference as long as its environment is willing to accept anything. On the other hand, it is the result of composing two DTS's that, no matter how the environment behaves, has (internal) computation interference. Without the empty DTS we cannot distinguish these two situations, since the greatest lower bound of the empty collection of DTS's is O^* in the absence of the empty DTS. This is very similar to the need to introduce divergence into the semantics of CSP. Therefore, we add the empty DTS as the top of our lattice. Once done so, there are at least two ways to define interference in a closed collection containing an empty DTS: it does or does not have computation interference. In this paper we chose for interference in this situation. One of the reasons is that we believe a

certain substitution theorem to hold, when done this way. This substitution theorem states that we may replace, in a collection X of DTS's, a collection $Y \subseteq X$ by an equivalent one without changing any interference properties of the collection X (also cf. Theorem 7, esp. its Corollary). However, the consequence of this choice is that another theorem we would like to hold, viz. $\mathbf{nai}\{S,T,\tilde{U}\}$, where U is the composite of S and T, does not hold. Making the other choice, makes the latter theorem true, but the former becomes false. We are still investigating more elegant ways to incorporate the empty DTS.

It seems to us, that the analysis of the three-person game should be carried out before addressing the issue of composition. Due to lack of time we have not done so, so far. In order to give the reader some feel for how composition can be defined we have given two definitions for the synchronous composition, one in the section on isomorphic representations and one in the section on composition. Other, alternative and maybe more elegant ways to define composition, can be given. For example, given an equivalence relation the composite of S and T can be defined is a representative of the DTS's that are equivalent to the collection $\{S,T\}$. In the case of synchronous composition each class has only one member, which represents that class. In the case of the asynchronous game we could choose for the one delay-insensitive DTS in the collection of DTS's equivalent to $\{S,T\}$, except when there is always interference in which case we choose the empty DTS as the result of the composition. More research is called for before we can make the most appropriate choice among these possibilities.

8 Acknowledgements

This research has been made possible by the financial support of the Computer Science Department at Washington University, St. Louis. We are most grateful to this department, especially to its chairman Jerome R. Cox Jr. We thank Wei Chen and Charles E. Molnar for many insightful comments upon earlier versions of this paper, which were made at numerous discussions while we were still shaping our ideas.

References

[0] M. Hennessy, *Algebraic Theory of Processes*, MIT Press Series in Found. of Comp., Cambridge Mass., 1988.

[1] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice-Hall Int., London, 1985.

[2] E.-R. Olderog and C.A.R. Hoare, "Specification-Oriented Semantics for Communicating Processes", *Acta Informatica*, vol. 23, pp. 9–66, 1986.

[3] J.T. Udding, *Classification and Composition of Delay-Insensitive Circuits*, Ph.D. Thesis, Eindhoven, 1984.

[4] T. Verhoeff, *Notes on Delay-Insensitivity*, Master's Thesis, Eindhoven, 1985.