Denotational semantics for thread algebra

Thuy Duong Vu

Sectie Software Engineering, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Received 11 May 2007; accepted 11 May 2007
Available online 26 May 2007

Abstract

This paper gives a denotational semantics for thread algebra (TA), an algebraic framework for the description and analysis of recent programming languages such as C# and Java [J.A. Bergstra, C.A. Middelburg, Thread algebra for strategic interleaving, Formal Aspects of Computing, in press. Preliminary version: Computing Science Report PGR0404: Sectie Software Engineering, University of Amsterdam]. We illustrate the technique taken from the metric topology of de Bakker and Zucker [J.W. Bakker, J.I. Zucker, Processes and the denotational semantics of concurrency, Inform. and Control 54 (1/2) (1982) 70–120] to turn the domains of TA into complete metric spaces. We show that the complete metric space consisting of projective sequences is an appropriate domain for TA. By using Banach’s fixed point theorem, we prove that the specification of a regular thread determines a unique solution. We also give a structural operational semantics for thread algebra and its relation to our denotational semantics. Finally, we present a particular interleaving strategy for TA that deals with abstraction in a natural way.

© 2007 Elsevier Inc. All rights reserved.

Keywords: Thread algebra; Metric denotational semantics; Strategic interleaving

1. Introduction

In 2002, Bergstra and Loots proposed a semantics for sequential programming languages called basic polarized process algebra (BPPA) [8]. Later, Bergstra and Middelburg renamed BPPA to basic thread algebra (BTA) and extended BTA to thread algebra (TA) with a collection of strategic interleaving operators [10]. It has been outlined in [8,9,10] that TA is a dominant form of concurrency provided by recent object-oriented programming languages such as C# and Java, where arbitrary interleaving is not an appropriate intuition when dealing with multi-threading.

This paper is an extension of Chapters 4 and 5 of [18] which focuses on denotational and operational semantics for TA. The difference between these two semantics is that the former constructs expressions of a programming language as elements of some suitable domain equation [17] while the latter generates them in a stepwise fashion. We employ the metric methodology of de Bakker and Zucker [4] to give a denotational semantics for TA. This method turns the domain of single threads into a complete metric space, in which the distance between two threads that do not differ in behavior until the n-th step is at most \(2^{-n}\). We show that the metric space consisting of projective sequences of threads is an appropriate domain for TA by comparing it to other domains of TA. In particular, the infinite threads of this domain are represented in a unique way. Furthermore, it deals naturally with abstraction [14,7]. Moreover, it is

E-mail address: tdvu@science.uva.nl

1567-8326/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jlap.2007.05.002
compatible with the domain based on complete partial orders (cpo’s) of [5]. Our domain can be extended with strategic interleaving operators of [10] in a natural way while the domain based on cpo’s cannot. By means of Banach’s fixed point theorem, one can show that the specification of a regular thread has a unique solution.

Next, we provide a structural operational semantics (SOS) for TA. Our SOS is less general than the SOS introduced in [10,9] since it does not deal with blocking actions. However, it is simpler and bisimulation induced by this SOS characterizes equality induced by the axioms of TA as shown in [18]. We will explain the relation between the two semantics of TA presented in the paper.

Finally, we propose a particular interleaving strategy for TA, called the cyclic internal persistence operator, with respect to abstraction of internal actions. In TA, concrete internal actions [10] may arise due to the interactions between clients and servers [6], threads and services, and threads and the execution environment. It is stated in [10] that the presence of concrete internal actions matters, and there is no abstraction made of it via equations that remove these actions. However, abstraction is still necessary in certain cases. For instance, in [6], abstraction is defined to emulate the interaction between clients and servers, assuming that clients and servers are threads in BTA. It would be natural if abstraction is compositional with respect to interleaving strategies of parallel threads. Unfortunately, this property does not hold for the existing interleaving operators of the thread algebra given in [10]. The cyclic internal persistence strategy is a variant of the cyclic interleaving operator of [10] which will not invoke the rotation of a thread sequence if the current action is internal. We will show that with the use of this strategy, abstraction can be made compositional, provided that threads cannot perform an infinite sequence of internal actions.

The structure of this paper is as follows. Section 2 recalls the basic concepts of complete metric spaces, complete partial orders, SOS, BTA and TA. Section 3 turns the domains of BTA into complete metric spaces, and shows that the complete metric space consisting of projective sequences is an appropriate domain for BTA. Section 4 extends the domain of BTA with the strategic interleaving operators of TA. Section 5 presents a SOS for TA and its relation with our denotational semantics. Section 6 defines the cyclic internal persistence operator dealing with abstraction for TA. The paper is ended with some concluding remarks in Section 7.

2. Preliminaries

In this section, we provide some basic concepts that will be needed for the rest of the paper.

2.1. Metric spaces and complete partial orders

Complete metric spaces and complete partial orders have major applications in denotational semantics. In this paper, we will use a few basic concepts of the metric topology and the domain theory taken from [12,17,3] to give a denotational semantics for TA.

2.1.1. Metric spaces

A metric space is a set where a notion of distance (or metric) between elements of the set is defined.

Definition 1. A metric space is a pair \((M, d)\) consisting of a set \(M\) and a metric \(d\) on \(M\). The metric \(d(x, y)\) defined for arbitrary \(x\) and \(y\) in \(M\) is a nonnegative, real valued function satisfying for all \(x, y, z \in M\) the conditions:

1. \(d(x, y) = 0\) if and only if \(x = y\),
2. \(d(x, y) = d(y, x)\),
3. \(d(x, z) \leq d(x, y) + d(y, z)\).

\((M, d)\) is said to be an ultra-metric space if \(d\) satisfies the strong triangle inequality: For all \(x, y, z \in M, d(x, z) \leq \max\{d(x, y), d(y, z)\}\).

We note that for all \(x, y, z \in M\),

\[d(x, z) \leq \max\{d(x, y), d(y, z)\} \Rightarrow d(x, y) + d(y, z) \geq d(x, z).\]

The notion of complete metric spaces is based on Cauchy sequences defined as follows.

Definition 2. \((x_n)\) is a Cauchy sequence in the space \((M, d)\) if

\[\forall \epsilon > 0 \exists N \forall n, m > N : d(x_n, x_m) < \epsilon.\]
Definition 3. If every Cauchy sequence in the metric space $M$ converges to an element in $M$, $M$ is said to be complete.

Note that the space containing $M$, together with all limits of its Cauchy sequences is a completion of $M$, where the distance between the limit points $x^* = \lim_{n \to \infty} x_n$ and $y^* = \lim_{n \to \infty} y_n$ of $M$ is defined as $d(x^*, y^*) = \lim_{n \to \infty} d(x_n, y_n)$.

Given a metric space $(M, d)$, we define the metric $d'$ on the set $M^n (n \geq 1)$ as follows.

**Definition 4.** Let $(M, d)$ be a metric space. Let $X, Y \in M^n$ for some $n \geq 1$, $X = [X_1, \ldots, X_n]$, $Y = [Y_1, \ldots, Y_n]$. Then

$$d'(X, Y) = \max_{i \leq n} d(X_i, Y_i).$$

Then the pair $(M^n, d')$ constitutes a complete metric space if $(M, d)$ does.

**Lemma 5.** If $(M, d)$ is complete then so is $(M^n, d')$ for all $n \geq 1$.

By using Banach’s fixed point theorem, one can guarantee the existence and uniqueness of fixed points of contraction mappings in complete metric spaces. These notions are given formally as in the following:

**Definition 6.** An element $x \in X$ is said to be a fixed point of a function $f : X \to X$ if $f(x) = x$.

**Definition 7.** Let $(X, d)$ be a metric space. A function $f : X \to X$ is a contraction mapping if there is a real number $c < 1$ such that $d(f(x), f(y)) < c \cdot d(x, y)$ for each $x, y \in X$.

**Theorem 8** (Banach’s fixed point theorem, see [13]). Every contraction mapping of a complete metric space has a unique fixed point.

2.1.2. Complete partial orders

Complete partial orders are special classes of partially ordered sets. These orders are characterized by a completeness property which essentially says that every monotone sequence has a supremum. Formally:

**Definition 9.** Let $\sqsubseteq$ be a partial order in a set $D$. A monotone sequence $(P_n)_n$ in $D$ is a sequence satisfying

$$P_0 \subseteq P_1 \subseteq \cdots \subseteq P_n \subseteq P_{n+1} \subseteq \cdots$$

**Definition 10.** A complete partial order (cpo) $D = (D, \sqsubseteq)$ is a partially ordered set with a least element such that every monotone sequence has a supremum in $D$.

2.1.3. Compatibility between metric spaces and cpo’s

A complete partial order and a metric space can be compared by the notion of compatibility [3]. More precisely, a complete partial order and a complete metric space of the same set are compatible if the supremum and the limit of every monotone Cauchy sequence are identified.

**Definition 11.** A cpo $(D, \sqsubseteq)$ and a complete metric space $(M, d)$ are said to be compatible if $D = M$ and $\bigcup_n x_n = \lim_{n \to \infty} x_n$ for each monotone Cauchy sequence $(x_n)_n$.

2.2. Structural operational semantics (SOS)

 Structural operational semantics (SOS) [16] (see [1] for an overview) is a formal semantics of programming and specification languages. It generates a labeled transition system (LTS) whose states are closed terms over an algebraic signature, and whose transitions between states are obtained inductively from a collection of transition rules, called transition system specification (TSS).
2.2.1. Labeled transition systems

Definition 12. A labeled transition system (LTS) is a quadruple \((\text{State}, A, \{ \rightarrow | a \in A \}, \text{Pred})\) satisfying:
- \text{State} is a set of states (or threads);
- \(A\) is a set of actions;
- \(\rightarrow \subseteq (\text{State} \times \text{State})\) for every \(a \in A\);
- \(P \subseteq \text{State}\) for every \(P \in \text{Pred}\). We write \(PP\) if state \(P\) satisfies predicate \(P\).

Binary relations \(P \overset{a}{\rightarrow} Q\) in an LTS are called transitions. We write \(P \overset{a}{\rightarrow} Q\) for \((P, Q) \in \overset{a}{\rightarrow}\).

Bisimulation [15] is an important equivalence in process algebras that classifies processes (or threads) behaving identically.

Definition 13. Given an LTS \((\text{State}, A, \{ \rightarrow | a \in A \}, \text{Pred})\), a symmetric relation \(B \subseteq \text{State} \times \text{State}\) is a bisimulation if it satisfies:

1. If \((P, Q) \in B\) and \(PP\) then \(QP\) for all \(P \in \text{Pred}\).
2. If \((P, Q) \in B\) and \(P \overset{a}{\rightarrow} P'\) then there exists \(Q'\) such that \(Q \overset{a}{\rightarrow} Q'\) and \((P', Q') \in B\).

Two threads \(P\) and \(Q\) are bisimilar, denoted by \((P \leftrightarrow Q)\), if there is a bisimulation relation \(B\) such that \((P, Q) \in B\).

2.2.2. Transition system specifications

The states of an LTS can be given as closed terms over some signature. Let \(\text{Var}\) be an infinite set of variables, with typical elements \(x, y, z\). A signature is a set \(\text{Sig}\) of function symbols \(f\) with arity \(\ar(f)\). The set \(\overline{\text{Sig}}\) of terms is defined as usual. A term is closed if it does not contain any variables. Let \(t, u\) denote terms and \(P, Q\) closed terms. A substitution is a mapping \(\sigma : \text{Var} \rightarrow \overline{\text{Sig}}\). A substitution is closed if it maps each variable to a closed term in \(\overline{\text{Sig}}\).

The transitions between states in an LTS can be generated inductively from a collection of transition rules called a transition system specification (TSS). Given a term algebra, we define a TSS as follows.

Definition 14 (Transition system specification). A literal is an expression \(t \overset{a}{\rightarrow} t'\) or \(tP\). A transition rule is of the form \(\frac{H}{\pi}\), where \(H\) is a set of literals called the premises, and \(\pi\) is a literal. A rule \(\frac{\emptyset}{\pi}\) is also written \(\pi\). A transition system specification (TSS) is a set of transition rules. A transition rule is closed if it contains only closed terms.

We note that in the premises in the previous definition are positive. We do not consider negative premises in this paper. To define the LTS generated by a TSS, we use the notion of a proof of a closed transition rule from a TSS.

Definition 15. A proof from a TSS \(T\) of a closed transition rule \(\frac{H}{\pi}\) consists of an upwardly branching tree in which all upward paths are finite, where the nodes of the tree are labeled by transitions such that:

- the root has label \(\pi\), and
- if some node has label \(l\), and \(K\) is the set of labels of nodes directly above this node then
  1. either \(K\) is the empty set and \(l \in H\),
  2. or \(\frac{K}{l}\) is a closed substitution instance of a transition rule in \(T\).

Definition 16 (Generated LTS). The LTS generated by a TSS \(T\) consists of the transitions \(\pi\) such that \(\pi\) can be proven from \(T\).

2.3. Basic thread algebra (BTA) and thread algebra (TA)

We recall from [8,10,9] the notions of basic thread algebra (BTA) and thread algebra (TA). We note that BTA was introduced as basic polarized process algebra (BPPA) in [8].

2.3.1. Basic thread algebra as a cpo

Let \(\Sigma\) be a set of actions. Each action returns a boolean value after its execution. Basic thread algebra (BTA) is defined by the following operators:
• **Termination:** $S \in BTA$ yields the terminating behavior.
• **Inactive behavior:** $D \in BTA$, represents the inactive behavior.
• **Postconditional composition:** $(-) \preceq a \succeq (-)$ with $a \in \Sigma$. The thread $P \preceq a \succeq Q \in BTA$ with $P, Q \in BTA$ first performs $a$ and then proceeds with $P$ if true was returned or with $Q$ otherwise. In case $P = Q$ we abbreviate this thread by the action prefix operator: $a \circ (-)$. In particular, $a \circ P = P \preceq a \succeq P$.

To provide domains for $BTA$, we consider the following domain equation:

$$P = \{S, D\} \bigcup (P \preceq X \succeq P),$$

(1)

where $X \preceq \Sigma \succeq Y = \{x \preceq a \succeq y | x \in X, y \in Y, a \in \Sigma\}$. We say that a solution of (1) is a domain of $BTA$. Let $BTA_\Sigma$ be the set of finite threads in $BTA$ defined as follows.

**Definition 17.** $BTA_\Sigma$ is a set consisting of all finite threads which are made from $S$ and $D$ by means of a finite number of applications of postconditional compositions.

**Lemma 18.** $BTA_\Sigma$ is a solution of (1), and therefore, it is a domain of $BTA$.

**Proof.** It is obvious that $S, D \in BTA_\Sigma$. If $P$ and $Q$ in $BTA_\Sigma$ and $a \in \Sigma$ then $P \preceq a \succeq Q$ is also in $BTA_\Sigma$. Vice versa, if $R = P \preceq a \succeq Q \in BTA_\Sigma$ then $P, Q \in BTA_\Sigma$. □

Threads can be infinite. Infinite threads are given by sequences of finite approximations. In [5], a technique based on cpo’s is described to give a domain for $BTA$. The main idea of this approach is to define a binary relation $\preceq$, a partial order, on threads. The expression $P \preceq Q$ means that $P$ is an approximation of $Q$. It is shown that the set of projective sequences for threads is a cpo. This implies that it is a domain for $BTA$. Thus, it serves as a semantics for $BTA$ in a natural way.

**Definition 19**

(1) The partial ordering $\preceq$ on $BTA_\Sigma$ is generated by the clauses

(a) for all $P \in BTA_\Sigma$, $D \preceq P$, and
(b) for all $P, Q, X, Y \in BTA_\Sigma$, $a \in \Sigma$,

$$P \preceq X \& Q \preceq Y \Rightarrow P \preceq a \succeq Q \preceq X \preceq a \succeq Y.$$  

(2) Let $(P_n)_n$ and $(Q_n)_n$ be two sequences in $BTA_\Sigma$, then

$$(P_n)_n \preceq (Q_n)_n \iff \forall n \in \mathbb{N} : P_n \preceq Q_n.$$  

In order to define a projective sequence in $BTA_\Sigma$, an operator called the approximation operator that finitely approximates every thread is provided.

**Definition 20.** For every $n \in \mathbb{N}$, the approximation operator $\pi_n : BTA_\Sigma \to BTA_\Sigma$ is defined inductively by

$$\pi_0(P) = D,$$

$$\pi_{n+1}(S) = S,$$

$$\pi_{n+1}(D) = D,$$

$$\pi_{n+1}(P \preceq a \succeq Q) = \pi_n(P) \preceq a \succeq \pi_n(Q).$$

A projective sequence is a sequence $(P_n)_n \in \mathbb{N}$ such that for each $n \in \mathbb{N}$,

$$\pi_n(P_{n+1}) = P_n.$$  

One can show that:

**Lemma 21.** Every projective sequence is monotone.
The following lemma gives an intuition for the projective approximations of a thread. Given \( k, n \in \mathbb{N} \) with \( k \leq n \), the \( k \)th and the \( n \)th projective approximations of a thread do not differ in behavior until the \( k \)th step.

**Lemma 22.** Let \((P_n)_{n \in \mathbb{N}}\) be a projective sequence. Then for all \( k \leq n \), \( P_k = \pi_k(P_n) \).

**Proof.** This can be proven by induction on \( n \).

Let \( BTA^\infty_\Sigma \) be the set of projective sequences. For a thread \( P \) represented by a projective sequence \((P_n)_{n \in \mathbb{N}}\) in \( BTA^\infty_\Sigma \), we denote \( \pi_n(P) = P_n \).

**Theorem 23** \([5]\). \( BTA^\infty_\Sigma \subset BTA^\infty_\Sigma \) and \((BTA^\infty_\Sigma, \sqsubseteq)\) is a complete partial order.

The theorem above indicates that the cpo \((BTA^\infty_\Sigma, \sqsubseteq)\) is a domain for BTA in which infinite threads can be represented as suprema of monotone sequences of their finite approximations.

### 2.3.2. Regular threads

When dealing with infinite threads in process algebra, besides the method of providing finite approximations of an (infinite) thread, there is another way to construct infinite threads by means of guarded recursive specifications \([14,7,11]\). The threads defined by these specifications are called regular threads.

**Definition 24.** A thread \( P \) is regular if \( P = E_1 \), where \( E_1 \) is defined by a finite system of the form \((n \geq 1)\):

\[
\{E_i = t_i | 1 \leq i \leq n, t_i = S \text{ or } t_i = D \text{ or } t_i = E_{il} \leq a_i \geq E_{ir}\}
\]

with \( E_{il}, E_{ir} \in \{E_1, \ldots, E_n\} \) and \( a_i \in \Sigma \).

The finite system in the previous definition is called a guarded recursive specification. If \( E \) is a guarded recursive specification and \( X \) a recursive variable in \( E \), then \( \langle X | E \rangle \) denotes the thread that has to be substituted for \( X \) in the solution for \( E \).

### 2.3.3. Abstraction in BTA

Abstraction \([14,7,2]\) plays an important role in process algebra. It allows a simpler view of a thread, ignoring internal details. In \([6]\) abstraction is used to emulate the interaction between clients and servers, assuming that clients and servers are threads in BTA. We assume the existence of a concrete internal action \( \tau \in \Sigma \) that does not have any side effects and always replies true after its execution. This action can be abstracted away by an operator called the abstraction operator defined as follows.

**Definition 25.** Let \( \tau_{\tau} : BTA_\Sigma \to BTA_\Sigma \) be defined by

\[
\tau_{\tau}(S) = S, \\
\tau_{\tau}(D) = D, \\
\tau_{\tau}(P \leq t \geq Q) = \tau_{\tau}(P), \\
\tau_{\tau}(P \leq a \geq Q) = \tau_{\tau}(P) \leq a \geq \tau_{\tau}(Q) \quad (a \neq \tau \in \Sigma).
\]

It is shown in \([6]\) that the abstraction operator is monotone, i.e.:

**Lemma 26.** For all \( P, Q \in BTA_\Sigma \), \( P \sqsubseteq Q \Rightarrow \tau_{\tau}(P) \sqsubseteq \tau_{\tau}(Q) \).

Lemma 26 suggests the definition of abstraction of an infinite thread \( P \) given as the supremum of a monotone sequence of threads below.

**Definition 27.** Let \((P_n)_{n \in \mathbb{N}}\) be a monotone sequence of finite approximations of a thread \( P \in BTA^\infty_\Sigma \). Then \( \tau_{\tau}(P) = \bigsqcup_n \tau_{\tau}(P_n) \).
2.3.4. Thread algebra

Thread algebra (TA) is a specific process algebra which is designed for strategic interleaving of parallel threads. A single thread is defined in BTA. A thread vector is a finite sequence of threads. Strategic interleaving operators turn a thread vector of arbitrary length into a single thread. This single thread obtained via a strategic interleaving operator is called a multi-thread. TA is meant to specify the collection of strategic interleaving operators, capturing essential aspects of multi-threading. For a simplification, in this paper, we only consider the simplest interleaving strategy called the cyclic interleaving operator [10].

Let $\langle \rangle$ denote the empty sequence, $\langle x \rangle$ stands for a sequence of length one, and $\alpha \triangleright \beta$ for the concatenation of two sequences. We assume that the following identity holds: $\alpha \triangleright \langle \rangle = \langle \rangle \triangleright \alpha = \alpha$.

**Definition 28.** The axioms for the cyclic interleaving operator on finite threads are given as follows:

\[
\|_{csi} (\langle \rangle) = S,
\|_{csi} ((S) \triangleright \alpha) = \|_{csi} (\alpha),
\|_{csi} ((D) \triangleright \alpha) = S_D(\|_{csi} (\alpha)),
\|_{csi} ((x \leq_a y) \triangleright \alpha) = \|_{csi} (\alpha \triangleright \langle y \rangle) \leq_a \|_{csi} (\alpha \triangleright \langle y \rangle),
\]

where the auxiliary deadlock at termination operator $S_D$ turns termination into deadlock and is defined by

\[
S_D(S) = D,
S_D(D) = D,
S_D(x \leq_a y) = S_D(x) \leq_a S_D(y).
\]

We note that for a thread vector of length one, the cyclic interleaving operator turns the thread vector into the single thread contained in it.

3. BTA as a complete ultra-metric space

In the previous section, we have seen that BTA can be modeled as a complete partial order. In this section, we follow [4] to give a metric denotational semantics for BTA. We prove that the domain of BTA can be turned into a complete metric space in which the distance $d$ between two threads that do not differ in behavior until the $n$th step is at most $2^{-n}$. We show that the complete metric space $(\text{BTA}_\Sigma^\infty, d)$ consisting of projective sequences is an appropriate domain for BTA by proving:

1. Infinite threads in $(\text{BTA}_\Sigma^\infty, d)$ are represented in a unique way.
2. $(\text{BTA}_\Sigma^\infty, d)$ is compatible with the domain $(\text{BTA}_\Sigma^\infty, \subseteq)$.
3. $(\text{BTA}_\Sigma^\infty, d)$ deals with abstraction in a natural way, in comparison with the domain of Cauchy sequences.
4. Finally, the specification of a regular thread in $(\text{BTA}_\Sigma^\infty, d)$ determines a unique thread by using Banach’s fixed point theorem.

3.1. The metric $d$ between threads

We formally define a metric (or distance) $d$ between two threads in $\text{BTA}_\Sigma$ as follows.

**Definition 29**

1. $d(S, S) = 0, d(D, D) = 0,$
   
   \[ d(P, P') = 1 \text{ if } P \in \{S, D\} \text{ and } P' \neq P \text{ with } P' \in \text{BTA}_\Sigma \text{ or vice versa}, \]

2. $d(P_1 \leq a_1 \geq P_2, Q_1 \leq a_2 \geq Q_2) = \begin{cases} 1 & \text{if } a_1 \neq a_2, \\ \frac{1}{2} \max\{d(P_1, Q_1), d(P_2, Q_2)\} & \text{otherwise} \end{cases}$

with $P_1, Q_1, P_2, Q_2 \in \text{BTA}_\Sigma$. 
According to Definition 29, the metric between two finite threads that do not differ in behavior until the $n$th step is at most $2^{-n}$.

**Lemma 30.** Let $P, Q \in \text{BTA}_\Sigma$. Then for all $n \in \mathbb{N}$,
\[
d(P, Q) \leq \frac{1}{2^n} \iff \pi_n(P) = \pi_n(Q).
\]

**Proof.** This can be proven by induction on $n$. □

One can show that the set of finite threads with the metric $d$ constitutes an ultra-metric space.

**Lemma 31.** $(\text{BTA}_\Sigma, d)$ is an ultra-metric space.

This lemma suggests the completion $(\text{BTA}_{\Sigma}^\omega, d)$ of the metric space $(\text{BTA}_\Sigma, d)$ whose elements are the limits of all Cauchy sequences in $(\text{BTA}_\Sigma, d)$.

### 3.2. The uniqueness of threads in $(\text{BTA}_\Sigma^\omega, d)$

In this section, we show that completion $(\text{BTA}_{\Sigma}^\omega, d)$ of all limits of Cauchy sequences of finite threads and the metric space $(\text{BTA}_\Sigma^\omega, d)$ of projective sequences achieve two equivalent domains for BTA. However, the domain $(\text{BTA}_\Sigma^\omega, d)$ represents (infinite) threads in a unique way.

First of all, we prove that the complete metric space $(\text{BTA}_{\Sigma}^\omega, d)$ is a solution of (1).

**Lemma 32.** $\text{BTA}_\Sigma^\omega = \{S, D\} \cup (\text{BTA}_\Sigma^\omega \subseteq \Sigma \supseteq \text{BTA}_\Sigma^\omega)$.

**Proof.**

1. $(\supseteq)$: Since $\{S, D\} \subseteq \text{BTA}_\Sigma$, $\{S, D\} \subseteq \text{BTA}_{\Sigma}^\omega$. We prove that if $P, Q \in \text{BTA}_{\Sigma}^\omega$ then $(P \preceq a \succeq Q) \in \text{BTA}_{\Sigma}^\omega$.

\[
P = \lim_{n \rightarrow \infty} P_n, \quad Q = \lim_{n \rightarrow \infty} Q_n
\]

for some Cauchy sequences $(P_n)_n$ and $(Q_n)_n$. It is not hard to see that $(P_n \preceq a \succeq Q_n)_n$ is also a Cauchy sequence and $(P \preceq a \succeq Q) = \lim_{n \rightarrow \infty} P_n \preceq a \succeq Q_n$. Thus, $P \preceq a \succeq Q \in \text{BTA}_{\Sigma}^\omega$.

2. $(\subseteq)$: If $P \notin \{S, D\}$ then $P = S$ or $P = D$ or $P = Q \preceq a \succeq R$, $Q, R \in \text{BTA}_{\Sigma}^\omega$. We only consider the case $P \notin \{S, D\}$. Since $P \notin \text{BTA}_\Sigma^\omega$, $P = \lim_{n \rightarrow \infty} P_n$ for some Cauchy sequence $(P_n)_n$. Without lack of generality we can assume that for all $n$, $P_n = Q_n \preceq a \succeq R_n$. Since $(P_n)_n$ is a Cauchy sequence and $d(P_n, P_m) = \frac{1}{2} \max(d(Q_n, Q_m), d(R_n, R_m))$, $(Q_n)_n$ and $(R_n)_n$ are also Cauchy sequences. Therefore, there exist $Q$ and $R$ in $\text{BTA}_{\Sigma}^\omega$ such that $Q = \lim_{n \rightarrow \infty} Q_n, R = \lim_{n \rightarrow \infty} R_n$. Hence $P = Q \preceq a \succeq R$. □

The previous lemma shows that the completion $(\text{BTA}_{\Sigma}^\omega, d)$ is a domain for BTA in which an infinite thread can be represented by a class of Cauchy sequences with the same limit. It can be seen that these representations are equivalent to projective sequences of threads. More precisely, we show that the metric space $(\text{BTA}_{\Sigma}^\omega, d)$ yields an equivalent domain in which all limits of Cauchy sequences are represented in a unique way. We will use some supporting results.

In the next lemma, we prove that every projective sequence is a Cauchy sequence.

**Lemma 33.** $(\text{BTA}_{\Sigma}^\omega, d) \subseteq (\text{BTA}_{\Sigma}^\omega, d)$.

**Proof.** Let $(P_n)_n$ be an element in $\text{BTA}_{\Sigma}^\omega$. By Lemma 22, for all $m, n \in \mathbb{N}$, $m > n > 0$, $P_{n-1} = \pi_{n-1}(P_n) = \pi_{n-1}(P_m)$. Therefore, by Lemma 30, $d(P_n, P_m) = \frac{1}{2^n}$. This implies that $(P_n)_n$ is a Cauchy sequence. □

We now show that for every Cauchy sequence, there always exists a projective sequence having the same limit.

**Lemma 34.** $(\text{BTA}_{\Sigma}^\omega, d) \subseteq (\text{BTA}_{\Sigma}^\omega, d)$.
Proof. Let \( Q \) be an element in \((\text{BTA}_\Sigma, d)\). We will show that there exists \( P = (P_k)_k \) in \((\text{BTA}_\Sigma^\infty, d)\) such that \( P = Q \). Since \( Q \) is an element in \((\text{BTA}_\Sigma^\omega, d)\), \( Q = \lim_{k \to \infty} Q_k \) for some Cauchy sequence \((Q_k)_k\). By Definition 2, we have that

\[
\forall \epsilon > 0 \exists N \in \mathbb{N} \forall m, n > N : d(Q_m, Q_n) < \epsilon.
\]

We choose a sequence \( N_0, N_1, \ldots \) of natural numbers satisfying \( \pi_k(\pi_{k+1}(Q_{N_k+1})) = \pi_k(Q_{N_k}) \) inductively as follows.

- Let \( \epsilon = \frac{1}{2^0} \). Then there exists \( N_0 \in \mathbb{N} \) such that for all \( m, n \geq N_0 \), \( d(Q_m, Q_n) < \frac{1}{2^0} \). It follows from Lemma 30 that for all \( m, n \geq N_0 \), \( \pi_0(Q_n) = \pi_0(Q_{N_0}) \).

- Assume that we have chosen the numbers \( N_0, \ldots, N_k \) such that for all \( m, n \geq N_k \), \( d(Q_m, Q_n) < \frac{1}{2^{k+1}} \). Then by Lemma 33, for all \( m, n \geq N_{k+1} \), \( d(Q_m, Q_n) < \frac{1}{2^{k+1}} \). Thus, by the induction hypothesis, \( \pi_k(Q_{N_{k+1}}) = \pi_k(Q_{N_k}) \).

Let \( P_k = \pi_k(Q_{N_k}) \) for all \( k \in \mathbb{N} \). Then \( P = (P_k)_k \) is an element of \( \text{BTA}_\Sigma^\infty \). To see that \( d(P, Q) = 0 \), consider \( m, n \in \mathbb{N} \) such that \( m > \max[N_k, n] \). Then \( \pi_m(Q_n) = \pi_n(Q_{N_n}) = P_n = \pi_n(P_m) \). Thus, \( d(P_m, Q_n) < \frac{1}{2^{k}} \). Hence \( \lim_{m \to \infty} d(P_m, Q_m) = 0 \) or \( d(P, Q) = 0 \). \( \Box \)

It follows from Lemma 33 and Lemma 34 that the metric spaces \((\text{BTA}_\Sigma^\omega, d)\) and \((\text{BTA}_\Sigma^\infty, d)\) are equivalent. Formally:

**Theorem 35.** \((\text{BTA}_\Sigma^\infty, d) = (\text{BTA}_\Sigma^\omega, d)\).

In addition, pointwise equal threads in \((\text{BTA}_\Sigma^\infty, d)\) are identified. That is, the approximations of two equivalent threads in \((\text{BTA}_\Sigma^\infty, d)\) are equal at every step. To prove this, we will use an auxiliary lemma which essentially says that the distance between the \( k \)th projective approximations \( P_k \) and \( Q_k \) of two infinite threads \( P \) and \( Q \) is monotone. Since these distances are less than or equal to 1, the distance of two (finite or infinite) threads is equal to the supremum of the distances between their \( k \)th projective approximations.

**Lemma 36.** For all \((P_n)_n, (Q_n)_n \in \text{BTA}_\Sigma^\infty, d(P_n, Q_n)\) is a non-decreasing sequence. Therefore,

\[
\lim_{n \to \infty} d(P_n, Q_n) = \bigcup_{n \in \mathbb{N}} d(P_n, Q_n).
\]

**Proof.** We show that for all \( n \in \mathbb{N} \), \( d(P_n, Q_n) \leq d(P_{n+1}, Q_{n+1}) \). It follows from Lemma 22 that

\[
d(P_n, Q_n) = d(\pi_n(P_{n+1}), \pi_n(Q_{n+1})) \leq d(P_{n+1}, Q_{n+1}).
\]

The previous lemma implies the uniqueness of representations of infinite threads by projective sequences, i.e.:

**Lemma 37.** Let \( P \) and \( Q \) be two threads in \((\text{BTA}_\Sigma^\infty, d)\) which are represented by two projective sequences \((P_n)_n\) and \((Q_n)_n\), respectively. Then

\[
P = Q \iff \forall n \in \mathbb{N} : P_n = Q_n.
\]

**Proof.** If \( P_n = Q_n \) for all \( n \in \mathbb{N} \) then \( d(P, Q) = \lim_{n \to \infty} d(P_n, Q_n) = 0 \). Therefore, \( P = Q \). We now show that if \( P = Q \) then \( P_n = Q_n \) for all \( n \in \mathbb{N} \). It follows from Lemma 36 that for all \( n \in \mathbb{N} \), \( d(P_n, Q_n) \leq d(P, Q) = 0 \). Hence \( d(P_n, Q_n) = 0 \) or \( P_n = Q_n \) for all \( n \in \mathbb{N} \). \( \Box \)

**3.3. Compatibility between \((\text{BTA}_\Sigma^\infty, \sqsubseteq)\) and \((\text{BTA}_\Sigma^\infty, d)\)**

This section shows the compatibility of the two domains \((\text{BTA}_\Sigma^\infty, \sqsubseteq)\) and \((\text{BTA}_\Sigma^\infty, d)\) based on complete partial orders and complete metric spaces for BTA. We will use the following lemma which states that given a monotone
sequence of finite threads and a number $n \in \mathbb{N}$, there always exists a subsequence with the property that the $n$th projective approximations of the threads in the subsequence do not differ.

**Lemma 38.** Let $(P_n)_n$ be a monotone sequence of finite threads. Then
\[
\forall n \exists N \forall m > N : \pi_n(P_m) = \pi_n(P_N).
\]

**Proof.** We distinguish two cases. If for all $m, P_m \in \{D, S\}$ then there exists a minimal $N$ such that for all $m > N$, $P_m = P_N$. Thus, for all $n, \pi_n(P_m) = \pi_n(P_N)$. The other case is that there exists a minimal $N_0$ such that for all $m \geq N_0$, $P_m = Q_m \preceq a \succeq R_m$. It is not hard to see that $(Q_m)_m$ and $(R_m)_m$ are also monotone sequences. We note that for all $m < N_0$, $Q_m = R_m = D$. We employ induction on $n$.

(1) If $n = 0$ then $N = 0$.

(2) If $n > 0$ then for all $m \geq N_0, \pi_n(P_m) = \pi_{n-1}(Q_m) \preceq a \succeq \pi_{n-1}(R_m)$. Applying the induction hypothesis, there exist $N_1$ and $N_2$ such that for all $m > N_1, \pi_{n-1}(Q_m) = \pi_{n-1}(Q_{N_1})$ and for all $m > N_2, \pi_{n-1}(R_m) = \pi_{n-1}(R_{N_2})$.

Let $N = \max\{N_0, N_1, N_2\}$. Then for all $m > N$, $\pi_n(P_m) = \pi_n(P_N)$. Therefore, for all $n \in \mathbb{N}$, there exists $N \in \mathbb{N}$ such that for all $m > N$, $\pi_n(P_m) = \pi_n(P_N)$. □

**Lemma 38** implies that a monotone sequence of finite threads is also a Cauchy sequence. Formally:

**Lemma 39.** Every monotone sequence $(P_n)_n$ of finite threads is a Cauchy sequence. As a consequence, $\bigcup_n P_n = \lim_{n \to \infty} P_n$.

**Proof.** Let $P = \bigcup_n P_n$. It follows from Lemma 38 and Lemma 30 that for all $n \in \mathbb{N}$, there exists $N \in \mathbb{N}$ such that for all $m > N$, $d(P_m, P) < \frac{1}{2^n}$. This implies that $\lim_{n \to \infty} P_n = P$. Hence, $(P_n)_n$ is a Cauchy sequence and $\bigcup_n P_n = \lim_{n \to \infty} P_n$. □

Hence, by Definition 11, the two domains of BTA based on complete partial orders and complete metric spaces are compatible.

**Theorem 40.** $(\text{BTA}_{\infty}, \subseteq)$ and $(\text{BTA}_{\infty}^\omega, d)$ are compatible.

3.4. Abstraction in $(\text{BTA}_{\infty}^\omega, d)$ and $(\text{BTA}_{\infty}, d)$

This section shows an advantage of the domain $(\text{BTA}_{\infty}^\omega, d)$ of projective sequences, in comparison with the domain $(\text{BTA}_{\infty}, d)$ of Cauchy sequences. More precisely, the former can deal with abstraction in a natural way, while the latter cannot.

As we have seen in the previous section, the two domains $(\text{BTA}_{\infty}, \subseteq)$ and $(\text{BTA}_{\infty}^\omega, d)$ are compatible. As a result, for a projective sequence $(P_n)_n$, the monotone sequence $(\tau_{\text{tau}}(P_n))_n$ has a limit. Hence, the abstraction of an (infinite) thread $P \in (\text{BTA}_{\infty}^\omega, d)$ represented by the projective sequence $(P_n)_n$ can be defined as the limit of the sequence $(\tau_{\text{tau}}(P_n))_n$. This definition coincides with the definition of abstraction of infinite threads in the domain $(\text{BTA}_{\infty}, \subseteq)$.

**Lemma 41.** Let $(P_n)_n$ be a projective sequence representing a thread $P \in \text{BTA}_{\infty}^\omega$. Then $\lim_{n \to \infty} \tau_{\text{tau}}(P_n)$ exists. In particular,
\[
\lim_{n \to \infty} \tau_{\text{tau}}(P_n) = \bigcup_{n \to \infty} \tau_{\text{tau}}(P_n) = \tau_{\text{tau}}(P).
\]

Abstraction, however, cannot be defined by means of Cauchy sequences. In particular, abstraction is not continuous in $(\text{BTA}_{\infty}^\omega, d)$.

**Lemma 42.** There exists $P = \lim_{n \to \infty}(P_n)_n$ for a Cauchy sequence $(P_n)_n$ such that $\lim_{n \to \infty} \tau_{\text{tau}}(P_n) \neq \tau_{\text{tau}}(P)$.

**Proof.** Let $(P_n)_n$ be defined as follows:
\[(P_n)_n = D, \tau \circ S, \tau^2 \circ D, \ldots, \tau^{2n} \circ D, \tau^{2n+1} \circ S, \ldots\]
One can see that \((P_n)_n\) is a Cauchy sequence. Let \(P = \lim_{n \to \infty} P_n\). Then \(P \in \text{BTA}_\Sigma^\omega\). Thus, there exists \(\tau_{\text{tau}}(P) \in \text{BTA}_\Sigma^\omega\). However, the sequence 
\[
(\tau_{\text{tau}}(P_n))_n = D, S, D, \ldots, D, S, \ldots
\]
is not a Cauchy sequence. Thus, it does not have a limit in \(\text{BTA}_\Sigma^\omega\). Therefore, \(\lim_{n \to \infty} \tau_{\text{tau}}(P_n) \neq \tau_{\text{tau}}(P)\). □

### 3.5. The uniqueness of regular threads in \((\text{BTA}_\Sigma^\omega, d)\)

We now show that a guarded recursive specification determines a unique thread by using Banach’s fixed point theorem. We consider the thread represented by this specification as a component of the solution of the equation \(X = T(X)\), where the definition of \(T\) is given as follows.

**Definition 43.** Let \(T : (\text{BTA}_\Sigma^\omega)^n \to (\text{BTA}_\Sigma^\omega)^n\) be defined such that

\[
T = \lambda X. [t_1(X), \ldots, t_n(X)],
\]

where

- \(t_i = \lambda X_1, \ldots, X_n. S\) or
- \(t_i = \lambda X_1, \ldots, X_n. D\) or
- \(t_i = \lambda X_1, \ldots, X_n. X_{il} \leq a_i \geq X_{ir}\)

with \(X_{il}, X_{ir} \in \{X_1, \ldots, X_n\}\).

Given a complete metric space \((\text{BTA}_\Sigma^\omega, d)\), we define the metric \(d'\) on \((\text{BTA}_\Sigma^\omega)^n\) as in Definition 4, assuming that \(\text{BTA}_\Sigma^\omega = M\). Thus, by Lemma 5, the metric space \(((\text{BTA}_\Sigma^\omega)^n, d')\) is also complete.

**Theorem 44.** \(T\) has a unique fixed point.

**Proof.** Let \(I\) be the set of all indexes \(i\) such that \(t_i = X_{il} \leq a_i \geq X_{ir}\). Then \(d(t_i(X), t_i(Y)) = 0\) if \(i \notin I\), since \(t_i(X)\) is a constant, and \(d(t_i(X), t_i(Y)) = \frac{1}{2} \max\{d(X_{il}, Y_{il}), d(X_{ir}, Y_{ir})\}\) otherwise. Let \(X, Y\) be elements of \((\text{BTA}_\Sigma^\omega)^n\). By Definition 4 we have

\[
d'(T(X), T(Y)) = \max_{i \leq n} d(t_i(X), t_i(Y))
\]

\[
= \max_{i \in I} \left(\frac{1}{2} \max\{d(X_{il}, Y_{il}), d(X_{ir}, Y_{ir})\}\right)
\]

\[
\leq \frac{1}{2} \max_{i \leq n} d(X_i, Y_i) = \frac{1}{2} d'(X, Y).
\]

It follows from Definition 7 that \(T\) is a contraction mapping. Since \(((\text{BTA}_\Sigma^\omega)^n, d')\) is complete and by Banach’s fixed point theorem, \(T\) has a unique solution. □

The previous theorem implies the uniqueness of regular threads. This suggests the set \(\text{BTA}_\Sigma^\omega\) consisting of regular threads in \((\text{BTA}_\Sigma^\omega, d)\).

**Lemma 45.** \(\text{BTA}_\Sigma^\omega\) is a domain of \(\text{BTA}\), and \(\text{BTA}_\Sigma \subset \text{BTA}_\Sigma^\omega\subset \text{BTA}_\Sigma^\omega\).

**Proof.** It is straightforward that \(\text{BTA}_\Sigma^\omega\) is a domain of \(\text{BTA}\), and \(\text{BTA}_\Sigma \subset \text{BTA}_\Sigma^\omega\subset \text{BTA}_\Sigma^\omega\). In the following, we give two examples to show the strictness of the inclusions.

1. \(\text{BTA}_\Sigma \subset \text{BTA}_\Sigma^\omega\): Let \(R = a \circ R\). Then \(R \in \text{BTA}_\Sigma^\omega\) but \(R \notin \text{BTA}_\Sigma\).
2. \(\text{BTA}_\Sigma^\omega \subset \text{BTA}_\Sigma^\omega\): Let \(P = (P_n)_n\) be the thread taken from [11] which performs \(a \circ b \circ a \circ b^2 \circ a \circ b^3 \circ \ldots\):
4. Extending BTA with strategic interleaving operators to TA

In this section, we show that the domain \((\mathbb{TA}_\infty^\Sigma, d)\) of BTA can be extended with the strategic interleaving operators of thread algebra. For simplicity, we will consider only the basic strategy, the cyclic interleaving operator in [10]. Other operators as TA\(\infty^\Sigma\) with strategic interleaving operators as TA\(\Sigma\) and TA\(\infty^\Sigma\), respectively. It will be shown that multi-threads in TA\(\infty^\Sigma\) can be defined by means of Cauchy sequences in the domain \((\mathbb{TA}_\infty^\Sigma, d)\), but not by means of monotone sequences in the domain \((\mathbb{TA}_\infty^\Sigma, \sqsubseteq)\). We will provide the projective sequence for a multi-thread by the projective sequences of its components.

4.1. \((\mathbb{TA}_\infty^\Sigma, d)\) as an appropriate domain for TA

As followed from the previous section, the complete metric space \((\mathbb{TA}_\infty^\Sigma, d)\) contains all limits of Cauchy sequences of finite threads. Given a thread vector of some limits in \(\mathbb{TA}_\infty^\Sigma\), we will show that the limit of the sequence, whose elements are the multi-threads obtained by vectors of the approximations of those limits via the cyclic interleaving operator, exists. We will use some supporting results.

Our results show that \((\mathbb{BTA}_\infty^\Sigma, d)\) is an appropriate domain for BTA, called the projective limit domain of BTA.

\[ P = a \circ Q_{1,0}, \]
\[ Q_{i+1,j} = b \circ Q_{1,j+1}, \]
\[ Q_{0,j} = a \circ Q_{j+1,0}. \]

Then \(P \in \mathbb{BTA}_\infty^\Sigma\) but \(P \notin \mathbb{BTA}_\Sigma^\infty\). We note that

\[ P_{n+1} = \begin{cases} 
\alpha \circ a \circ D & \text{if } P_n = \alpha \circ D \text{ and } n + 1 \leq \frac{1}{2}k(k + 1) \text{ for some } k, \\
\alpha \circ b \circ D & \text{if } P_n = \alpha \circ D \text{ and } n + 1 \neq \frac{1}{2}k(k + 1) \text{ for all } k.
\end{cases} \]

\[ \square \]

Our results show that \((\mathbb{BTA}_\infty^\Sigma, d)\) is an appropriate domain for BTA, called the projective limit domain of BTA.

4.1. \((\mathbb{TA}_\infty^\Sigma, d)\) as an appropriate domain for TA

As followed from the previous section, the complete metric space \((\mathbb{TA}_\infty^\Sigma, d)\) contains all limits of Cauchy sequences of finite threads. Given a thread vector of some limits in \(\mathbb{TA}_\infty^\Sigma\), we will show that the limit of the sequence, whose elements are the multi-threads obtained by vectors of the approximations of those limits via the cyclic interleaving operator, exists. We will use some supporting results.

The first auxiliary lemma shows the compositionality property of the deadlock at termination operator \(S_D\).

**Lemma 46.** Let \(P_i \in \mathbb{TA}_\Sigma\) \((1 \leq i \leq m)\) be finite threads. Then

\[ S_D(\|_{\text{csi}} ((P_1) \circ \cdots \circ (P_m))) = \|_{\text{csi}} ((S_D(P_1)) \circ \cdots \circ (S_D(P_m))). \]

**Proof.** This can be proven by induction on the lengths of threads. \(\square\)

Next, we prove that the distance of two finite threads after turning termination into deadlock decreases.

**Lemma 47.** Let \(P\) and \(Q\) be finite threads. Then \(d(S_D(P), S_D(Q)) \leq d(P, Q).\)

**Proof.** Straightforward. \(\square\)

The following lemma shows that the distance of two multi-threads with the same length obtained via the cyclic interleaving operator is always less than or equal to the maximum distance of their corresponding components.

**Lemma 48.** Let \(P_i\) and \(Q_i\) \((1 \leq i \leq m)\) be finite threads. Then

\[ d(\|_{\text{csi}} ((P_1) \circ \cdots \circ (P_m)), \|_{\text{csi}} ((Q_1) \circ \cdots \circ (Q_m))) \leq \max_{1 \leq i \leq m} \{d(P_i, Q_i)\}. \]

**Proof.** We prove this lemma by induction on the length and the number of threads. Let \(d = d(\|_{\text{csi}} ((P_1) \circ \cdots \circ (P_m)), \|_{\text{csi}} ((Q_1) \circ \cdots \circ (Q_m)))\). We distinguish the following cases:

1. \(P_1 \neq Q_1\) and \((P_1 \in \{S, D\} \text{ or } Q_1 \in \{S, D\})\). Then \(d(P_1, Q_1) = 1\). Thus, \(d \leq \max_{1 \leq i \leq m} \{d(P_i, Q_i)\} = 1\).
2. \(P_1 = Q_1 = S\) or \(P_1 = Q_1 = D\). Then \(d = d(\|_{\text{csi}} ((P_2) \circ \cdots \circ (P_m)), \|_{\text{csi}} ((Q_2) \circ \cdots \circ (Q_m)))\) or \(d = d(\|_{\text{csi}} ((S_D(P_2)) \circ \cdots \circ (S_D(P_m))), \|_{\text{csi}} ((S_D(Q_2)) \circ \cdots \circ (S_D(Q_m))))\). By the induction hypothesis and Lemma 47, \(d \leq \max_{1 \leq i \leq m} \{d(P_i, Q_i)\}\.\)
(3) \( P_1 = P' \leq a \geq P'' \) and \( Q_1 = Q' \leq a \geq Q'' \). Then
\[
d = \frac{1}{2} \max \left\{ d(\|_{csi} ((P_2) \bowtie \cdots \bowtie (P_m) \bowtie (P')), \|_{csi} ((Q_2) \bowtie \cdots \bowtie (Q_m) \bowtie (Q'))), \\
d(\|_{csi} ((P_2) \bowtie \cdots \bowtie (P_m) \bowtie (P'')), \|_{csi} ((Q_2) \bowtie \cdots \bowtie (Q_m) \bowtie (Q''))) \right\}.
\]
By the induction hypothesis and \( d(P_1, Q_1) = \frac{1}{2} \max\{d(P', Q'), d(P'', Q'')\} \),
\[
d \leq \max_{1 \leq i \leq m} \{d(P_i, Q_i)\}. \quad \Box
\]

**Lemma 49.** Let \((P_n^k)_{n}\) be Cauchy sequences for \(1 \leq k \leq m\). Then \((\|_{csi} ((P_n^1) \bowtie \cdots \bowtie (P_n^m)))_{n}\) is also a Cauchy sequence.

**Proof.** By Definition 2, we have
\[
\forall 1 \leq k \leq m \forall \epsilon > 0 \exists N_k \in \mathbb{N} \forall i, j > N_k : d(P_n^k, P_n^j) < \epsilon.
\]
Let \(Q_n = \|_{csi} ((P_n^1) \bowtie \cdots \bowtie (P_n^m))\) for all \(n \in \mathbb{N}\) and \(N = \max_{1 \leq k \leq m} \{N_k\}\). It follows from Lemma 48 that
\[
\forall \epsilon > 0 \exists N \in \mathbb{N} \forall i, j > N : d(Q_i, Q_j) < \epsilon.
\]
Therefore, \((Q_n)_{n}\) is a Cauchy sequence. \( \Box \)

Lemma 49 suggests a definition for multi-threads obtained by thread vectors in \((\text{TA}_\infty^\infty, d)\) via the cyclic interleaving operator \(\|_{csi} (-)\) as follows.

**Definition 50.** Let \(P_j = \lim_{n \to \infty} P_n^j\) (\(1 \leq j \leq m\)) be threads in \(\text{TA}_\infty^\infty\), where \((P_n^j)_{n}\) (\(1 \leq j \leq m\)) are Cauchy sequences. Then
\[
\|_{csi} ((P_1) \bowtie \cdots \bowtie (P_m)) = \lim_{n \to \infty} \|_{csi} ((P_n^1) \bowtie \cdots \bowtie (P_n^m))
\]
We note that these multi-threads cannot be defined by means of monotone sequences as can be seen in the following example.

**Example 51.** Let \((P_n)_{n}\) and \((Q_n)_{n}\) be monotone sequences of finite threads defined as follows: \(P_0 = D, P_n = a \circ D\) for all \(n > 0\) and \(Q_n = b \circ D\) for all \(n \geq 0\). Let \(R_n = \|_{csi} ((P_n) \bowtie (Q_n))\) for all \(n \geq 0\). Then the supremum of the sequence \((R_n)_{n}\) does not exist, since it is not monotone as \(R_0 = b \circ D\) and \(R_1 = a \circ b \circ D\).

### 4.2. Projective sequences of multi-threads

This section shows that the projective sequence of a multi-thread in \(\text{TA}_\infty^\infty\) can be computed by the projective sequences of its components.

First of all, we prove that two multi-threads are the same in behavior until the \(n\)th step if their corresponding components also are.

**Lemma 52.** Let \(P_i\) be single threads in \(\text{TA}_\infty^\infty\) for all \(1 \leq i \leq m\). Then
\[
\pi_n(\|_{csi} ((\pi_n(P_1)) \bowtie \cdots \bowtie (\pi_n(P_m)))) = \pi_n(\|_{csi} ((\pi_{i_1}(P_1)) \bowtie \cdots \bowtie (\pi_{i_m}(P_m))))
\]
with \(i_j \geq n\) for all \(1 \leq j \leq m\).

**Proof.** This can be proven by induction on \(n\) and \(m\). \( \Box \)
We now show that a multi-thread and the multi-thread obtained by the nth projective approximations of its components do not differ until the nth step. This property allows us to compute the projective sequence of a multi-thread by the projective sequences of its components.

**Theorem 53.** Let \( P_i \) be single threads in \( \text{TA}^\Sigma \) for all \( 1 \leq i \leq m \). Then
\[
\pi_n(\|e\| ((P_1) \circ \cdots \circ (P_m))) = \pi_n(\|e\| ((\pi_n(P_1)) \circ \cdots \circ (\pi_n(P_m)))).
\]

**Proof.** Let \( Q = \|e\| ((P_1) \circ \cdots \circ (P_m)) \) and \( Q_n = \|e\| ((\pi_n(P_1)) \circ \cdots \circ (\pi_n(P_m))) \). We show that \( \pi_n(Q) = \pi_n(Q_n) \) for all \( n \in \mathbb{N} \). It follows from Lemma 52 that \( (\pi_n(Q_n))_n \) is a projective sequence. Since \( d(\pi_n(Q_n), Q_n) \leq \frac{1}{n!} \), \( \lim_{n \to \infty} \pi_n(Q_n) = \lim_{n \to \infty} Q_n = Q \). Thus, \( \pi_n(Q_n) \) is a projective sequence of \( Q \). Hence, \( \pi_n(Q) = \pi_n(Q_n) \) for all \( n \in \mathbb{N} \). \( \Box \)

5. SOS for thread algebra

This section presents a SOS for TA. Our SOS is less general than the SOS of \([10,9]\), but it is simpler.

5.1. Labeled transition systems and transition rules for TA

We use the following transition relations on threads.

- The action step \( P \xrightarrow{a,k} P' \) which essentially says that a thread \( P \) is capable of first performing a basic action \( a \), and proceeding with thread \( P' \), where \( \kappa \in \{T, F\} \) denotes the returned boolean value after the execution of \( a \) (\( \kappa = T \) if \( \texttt{true} \) is returned after the execution of \( a \) and \( \kappa = F \) otherwise). This transition can also be written as \( P \xrightarrow{a} P' \) if \( P \xrightarrow{a,k} P' \) for both \( \kappa = T \) and \( \kappa = F \), or \( \kappa \) is always \( T \);
- The concrete internal action step \( P \xrightarrow{\tau} P' \) which essentially says that a thread \( P \) is capable of first performing an internal action \( \tau \), and proceeding with thread \( P' \);
- The termination \( P \downarrow \) means that thread \( P \) is capable of successful termination;
- The deadlock \( Q \uparrow \) means that thread \( Q \) is neither capable of performing an action nor capable of successful termination.

Let \( A = (\Sigma \setminus \{\tau\}) \times \{T, F\} \cup \{\tau\} \). A labeled transition system for TA is an LTS whose states are threads, whose actions are from the set \( A \), whose transitions are described as above, and whose predicates are \( \uparrow \) and \( \downarrow \). For a thread \( P \), we write \( P \uparrow \) if \( P \uparrow \) or \( Q \downarrow \). The transition rules for BTA, approximation operators, regular threads and the cyclic interleaving operators are given in Table 1. The transition rules for other strategic interleaving operators of TA can be given similarly. It is shown in \([18]\) that in the case of regular threads, bisimulation equivalence induced by our SOS characterizes equality induced by the axioms of TA.

5.2. Relation between SOS and denotational semantics of thread algebra

This section proves that bisimulation coincides with equality between regular threads.

**Theorem 54.** Let \( P \) and \( Q \) be two regular threads in \( \text{BTA}_\Sigma \). Then \( P \Leftrightarrow Q \Leftrightarrow P = Q \).

**Proof**

(1) \( \Rightarrow \): We show that \( \pi_n(P) \Leftrightarrow \pi_n(Q) \) for all \( n \in \mathbb{N} \). Let \( B \) be a relation defined by \( (\pi_n(P'), \pi_n(Q')) \in B \) for all \( n \in \mathbb{N} \) if \( P' \Leftrightarrow Q' \). One can see that \( B \) is a bisimulation. Since \( \pi_n(P) \) and \( \pi_n(Q) \) are finite, one can prove by induction on the length of \( \pi_n(P) \) and \( \pi_n(Q) \) that \( \pi_n(P) = \pi_n(Q) \) for all \( n \in \mathbb{N} \). Therefore, \( P = Q \).

(2) \( \Leftarrow \): Since \( P = Q \), \( \pi_n(P) = \pi_n(Q) \) for all \( n \in \mathbb{N} \). We define a binary relation \( B \) between threads \( P' \) and \( Q' \) as follows: \( (P', Q') \in B \) if \( \pi_n(P') = \pi_n(Q') \) for all \( n \in \mathbb{N} \). We show that \( B \) is a bisimulation. If \( P' \in \{S, D\} \) then this is trivial. If \( P' \xrightarrow{a} P'' \) then \( \pi_{n+1}(P') \xrightarrow{a} \pi_n(P'') \) for all \( n \in \mathbb{N} \). Since \( \pi_{n+1}(P') = \pi_{n+1}(Q') \), \( Q' \xrightarrow{a} Q'' \) and \( \pi_{n+1}(Q') \xrightarrow{a} \pi_n(Q'') \). Furthermore, \( \pi_n(P'') = \pi_n(Q'') \). This implies that \( (P'', Q'') \in B \). Therefore, \( B \) is a bisimulation. Hence, \( P \Leftrightarrow Q \). \( \Box \)
6. An interleaving strategy with respect to abstraction

We have introduced and discussed abstraction of single threads in Section 2.3.3 and Section 3.4. It would be natural if abstraction is compositional with respect to the interleaving strategies in [10]. Unfortunately, this property does not hold for the cyclic interleaving operator, the basic strategy of the interleaving strategies in [10], as can be seen in the following example.

**Example 55.** Let $P = \tau \circ a \circ S$ and $Q = b \circ S$ be two single threads. Then $\llbracket \text{cst} \rrbracket (P \concat (Q)) = \tau \circ b \circ a \circ S$. One can see that $\tau_{\text{tau}}(\llbracket \text{cst} \rrbracket (P \concat (Q))) = (b \circ a \circ S)$ and $\llbracket \text{cst} \rrbracket (\llbracket \tau_{\text{tau}}(P) \rrbracket \concat \llbracket \tau_{\text{tau}}(Q) \rrbracket) = (a \circ b \circ S)$ are not equal.

In this section, we propose a variant of the cyclic interleaving operator called the **cyclic internal persistence** operator for TA. We will show that this interleaving strategy deals with abstraction in a natural way.

### 6.1. The cyclic internal persistence operator

The phrase **cyclic internal persistence** means that upon the execution of a thread vector, the internal action $\tau$ is persistent. That is, its execution will not invoke the rotation of the thread vector. The definition of the cyclic internal persistence strategy is given below.

**Definition 56.** The axioms for the **cyclic internal persistence** operator $\llbracket \text{cip} \rrbracket$ on finite threads are given by

\[
\begin{align*}
\llbracket \text{cip} \rrbracket (\emptyset) &= S, \\
\llbracket \text{cip} \rrbracket (S \concat \alpha) &= \llbracket \text{cip} \rrbracket (\alpha), \\
\llbracket \text{cip} \rrbracket (D \concat \alpha) &= S_D (\llbracket \text{cip} \rrbracket (\alpha)), \\
\llbracket \text{cip} \rrbracket (\tau \circ x \concat \alpha) &= \tau \circ \llbracket \text{cip} \rrbracket (\{x\} \concat \alpha), \\
\llbracket \text{cip} \rrbracket (x \preceq a \succeq y \concat \alpha) &= \llbracket \text{cip} \rrbracket (\alpha \concat \{x\}) \preceq a \succeq \llbracket \text{cst} \rrbracket (\alpha \concat \{y\}).
\end{align*}
\]

Like the cyclic interleaving operator, we define the cyclic internal operator on infinite threads as follows.

**Definition 57.** Let $P_j = \lim_{n \to \infty} P_n^j$ ($1 \leq j \leq m$) be threads in $\text{TA}_\infty^\Sigma$, where $(P_n^j)_n$ ($1 \leq j \leq m$) are Cauchy sequences. Then

\[
\llbracket \text{cip} \rrbracket (P_1 \concat \cdots \concat P_m) = \lim_{n \to \infty} \llbracket \text{cip} \rrbracket ((P_n^1) \concat \cdots \concat (P_n^m))
\]
Furthermore, one can approximate the multi-threads obtained via the cyclic internal persistence operator by the projective approximations of its components.

**Theorem 58.** Let $P_i$ be single threads in $TA_{\Sigma}^\infty$ for all $1 \leq i \leq m$. Then

$$\pi_n(\|cip\((\langle P_1 \rangle \leadsto \cdots \leadsto \langle P_m \rangle)\)) = \pi_n(\|cip\((\langle \pi_n\(P_1) \rangle \leadsto \cdots \leadsto \langle \pi_n\(P_m) \rangle)\)).$$

**Proof.** Similar to the proof of Theorem 53. □

### 6.2. Compositionality of abstraction with respect to the cyclic internal persistence strategy

This section shows that abstraction satisfies compositionality with respect to the cyclic internal persistence operator, provided that threads cannot perform an infinite sequence of internal actions. The condition suggests an approximation operator $\pi_{\tau n}(-)$ which respects concrete internal actions. This operator only takes the performance of non-internal actions into account.

**Definition 59.** The approximation operator with respect to $\tau\pi_{\tau n}: TA_{\Sigma} \rightarrow TA_{\Sigma}$ is defined on finite threads by

$$\begin{align*}
\pi_{\tau 0}(P) &= D, \\
\pi_{\tau n+1}(S) &= S, \\
\pi_{\tau n+1}(D) &= D, \\
\pi_{\tau n+1}(\tau \circ P) &= \tau \circ \pi_{\tau n+1}(P), \\
\pi_{\tau n+1}(P \leq a \geq Q) &= \pi_{\tau n}(P) \leq a \geq \pi_{\tau n}(Q).
\end{align*}$$

A *projective sequence with respect to $\tau$* is a sequence $(P_n)_{n \in \mathbb{N}}$ such that for each $n \in \mathbb{N}$,

$$\pi_{\tau n}(P_{n+1}) = P_n.$$

One can prove that every projective sequence with respect to $\tau$ is monotone, and therefore, its supremum is in $(TA_{\Sigma}^\infty, \sqsubseteq)$. Let $TA_{\Sigma}^{\tau n}$ be the set of the threads represented by these projective sequences. For a thread $P \in TA_{\Sigma}^{\tau n}$ represented by a projective sequence $(P_n)_{n}$ with respect to $\tau$, we denote $\pi_{\tau n}^\tau(P) = P_n$.

In the following lemma, we will see that the abstraction of the $n$th projective approximation with respect to $\tau$ of a finite thread coincides with the $n$th projective approximation of its abstraction.

**Lemma 60.** Let $P$ be a finite thread. Then for all $n \in \mathbb{N}$,

$$\tau_{\tau n}(\pi_{\tau n}^\tau(P)) = \pi_{n}(\tau_{\tau}(P)).$$

**Proof.** This can be proven by induction on $n$. □

By using the approximation operator $\pi_{\tau n}^\tau(-)$, one can approximate a multi-thread in $TA_{\Sigma}^{\tau n}$ obtained via the cyclic internal persistence strategy, by the approximations of its components as follows.

**Theorem 61.** Let $P_i$ $(1 \leq i \leq m)$ be threads in $TA_{\Sigma}^{\tau n}$. Then

$$\|cip\((\langle P_1 \rangle \leadsto \cdots \leadsto \langle P_m \rangle)\) = \bigsqcup_n \pi_{\tau n}^\tau(\|cip\((\langle \pi_{\tau n}^\tau(P_1) \rangle \leadsto \cdots \leadsto \langle \pi_{\tau n}^\tau(P_m) \rangle)\)).$$

**Proof.** Let $Q = \|csi\((\langle P_1 \rangle \leadsto \cdots \leadsto \langle P_m \rangle)\)$ and $Q_n = \|csi\((\langle \pi_{\tau n}^\tau(P_1) \rangle \leadsto \cdots \leadsto \langle \pi_{\tau n}^\tau(P_m) \rangle)\)$. Similar to the proof of Theorem 53, one can show that the sequence $(\pi_{\tau n}^\tau(Q_n))_n$ is a projective sequence with respect to $\tau$. Therefore, $\bigsqcup_n(\pi_{\tau n}^\tau(Q_n)) = \lim_n(\pi_{\tau n}^\tau(Q_n)) = \lim_n Q_n = Q$. □
Finally, abstraction is compositional with respect to the cyclic internal persistence operator, provided that threads cannot perform an infinite sequence of internal actions.

**Theorem 62.** Let $P_i (1 \leq i \leq m)$ be threads in $\mathcal{T}_{\Sigma}$. Then

$$\tau^{\text{tau}}(\parallel \text{cip} (\langle P_1 \rangle \bowtie \cdots \bowtie \langle P_m \rangle )) = \parallel \text{cip} (\langle \tau^{\text{tau}}(P_1) \rangle \bowtie \cdots \bowtie \langle \tau^{\text{tau}}(P_m) \rangle ).$$

**Proof.** We consider two possibilities:

1. The threads $P_i (1 \leq i \leq m)$ are finite. The theorem can be proven by induction on the length of threads.
2. The threads $P_i (1 \leq i \leq m)$ are infinite. Let $P = (\parallel \text{csi} (\langle P_1 \rangle \bowtie \cdots \bowtie \langle P_m \rangle ))$. It follows from Theorem 58, Theorem 61, Lemma 60 and the previous case that

$$\tau^{\text{tau}}(P) = \bigcup_n \tau^{\text{tau}}(\pi_n (\parallel \text{cip} (\langle \pi_n(\tau^{\text{tau}}(P_1)) \rangle \bowtie \cdots \bowtie \langle \pi_n(\tau^{\text{tau}}(P_m)) \rangle )))$$

(by Theorem 61)

$$= \bigcup_n \pi_n (\tau^{\text{tau}}(\parallel \text{cip} (\langle \pi_n(\tau^{\text{tau}}(P_1)) \rangle \bowtie \cdots \bowtie \langle \pi_n(\tau^{\text{tau}}(P_m)) \rangle )))$$

(by Lemma 60)

$$= \bigcup_n \pi_n (\parallel \text{csi} (\langle \tau^{\text{tau}}(\pi_n(\tau^{\text{tau}}(P_1))) \rangle \bowtie \cdots \bowtie \langle \tau^{\text{tau}}(\pi_n(\tau^{\text{tau}}(P_m))) \rangle ))$$

(by 1)

$$= \bigcup_n \pi_n (\parallel \text{cip} (\langle \pi_n(\tau^{\text{tau}}(P_1)) \rangle \bowtie \cdots \bowtie \langle \pi_n(\tau^{\text{tau}}(P_m)) \rangle ))$$

(by Lemma 60)

$$= \parallel \text{cip} (\langle \tau^{\text{tau}}(P_1) \rangle \bowtie \cdots \bowtie \langle \tau^{\text{tau}}(P_m) \rangle )$$

(by Theorem 58).

□

7. Concluding remarks

We have studied a metric denotational semantics for TA. We have shown that the projective limit domain $(BTA_{\Sigma}^\infty, d)$ is an appropriate domain for BTA. In particular, this domain represents infinite threads in a unique way. Furthermore, it is compatible with the domain based on cpo’s in [5]. Moreover, it deals naturally with abstraction. As a consequence of Banach’s fixed point theorem, the specification of a regular thread yields a unique thread. In the setting of multi-threads, we have shown that $(BTA_{\Sigma}^\infty, d)$ can be extended with the cyclic interleaving operator. The extension of $(BTA_{\Sigma}^\infty, d)$ with the other strategic interleaving operators in [10] can be dealt with in the same way. In the paper, we have also presented a SOS for TA. We have shown that bisimulation induced by this SOS coincides with equality between regular threads. Finally, we have proposed an interleaving strategy with respect to abstraction, namely the cyclic internal persistence operator, for TA.

References