Report Number: WUCS-88-17

1988-05-01

Using a Partial Order and a Metric to Analyze a Recursive Trace Set Equation

Jan Tijmen Udding and Tom Verhoeff

In Trace Theory the notion of a process is defined in terms of a set of finite-length traces over an alphabet. These processes are used as the semantics for a program notation. The program text for a recursive component naturally gives rise to an equation over trace sets. This paper takes two approaches at the analysis of that equation. The first approach is based on a partial order and it concentrates on the projection operator for processes. This yields a condition under which the greatest solution of that equation can be approximated by iteration. The second approach introduces a... **Read complete abstract on page 2.**

Using a Partial Order and a Metric to Analyze a Recursive Trace Set Equation

Jan Tijmen Udding and Tom Verhoeff

Complete Abstract:

In Trace Theory the notion of a process is defined in terms of a set of finite-length traces over an alphabet. These processes are used as the semantics for a program notation. The program text for a recursive component naturally gives rise to an equation over trace sets. This paper takes two approaches at the analysis of that equation. The first approach is based on a partial order and it concentrates on the projection operator for processes. This yields a condition under which the greatest solution of that equation can be approximated by iteration. The second approach introduces a metric on the process domain. Application of Banach's Contraction Theorem results in a condition under which there exists a unique solution that can be approximated by iteration starting anywhere.

USING A PARTIAL ORDER AND A METRIC TO ANALYZE A RECURSIVE TRACE SET EQUATION

Jan Tijmen Udding and Tom Verhoeff

WUCS-88-17

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

Using a Partial Order and a Metric to Analyze a Recursive Trace Set Equation

Jan Tijmen Udding

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

*Tom Verhoeff**

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

May 1988

Abstract

In Trace Theory the notion of a process is defined in terms of a set of finite-length traces over an alphabet. These processes are used as the semantics for a program notation. The program text for a recursive component naturally gives rise to an equation over trace sets. This paper takes two approaches at the analysis of that equation.

The first approach is based on a partial order and it concentrates on the projection operator for processes. This yields a condition under which the greatest solution of that equation can be approximated by iteration. The second approach introduces a metric on the process domain. Application of Banach's Contraction Theorem results in a condition under which there exists a unique solution that can be approximated by iteration starting anywhere.

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

Contents

0 The Problem and Its Context

Throughout this paper the reader is assumed to be familiar with the basics of Trace Theory [8,5]. To make the paper self-contained we give a brief summary of the fundamental concepts and notations of Trace Theory in Subsection 0.1. For more details the reader is referred to [5, §1.1–1.3]. General notational conventions are explained in Subsection 0.0. Subsection 0.2 states the recursive equation and the questions about it that inspired the rest of the paper.

In Section 1 we present an overview of two general approaches to recursive equations, viz. based on a partial order and on a metric. A number of statements concerning finiteness of trace sets is collected in Section 2. This section also contains some Trace Theoretic concepts and notations not covered in [5]. It is only needed for the proofs in the succeeding section. Section 3 begins with the definition of the partial order and a summary of earlier results. The major part consists of a detailed investigation of the projection operator. Subsequently the results are applied to the recursive equation. In Section 4, which can be read independently of the preceding two sections, the metric is introduced and analyzed. As an application we strengthen an old theorem about the recursive equation and obtain a new proof for it. Finally, we summarize the results and make suggestions for further research in Section 5.

0.0 General Notational Conventions

We use a slightly unconventional notation for variable-binding constructs. For example, universal quantification is denoted by

$$(\forall\, l : D : E),$$

where $\forall$ is the quantifier, l is a list of bound variables, D is a predicate, and E is the quantified expression. Both D and E will, in general, contain variables from l. Predicate D determines the domain over which the bound variables range. Expression E need only be defined for values that satisfy D.

Existential quantification is denoted likewise using the quantifier $\exists$. The quantifier for set construction is implicit in the braces. For instance,

$$\{i : i \geq 0 : i^3\}$$

is the set of cubes. Numbers are implicitly restricted to the set of natural numbers unless stated otherwise.

A sequence in set D is a mapping from the natural numbers into D. Function application is in this case often denoted by subscripting. If F is, for example, the sequence of Fibonacci numbers then

$$F_{n+2} = F_{n+1} + F_n$$

for $n \geq 0$.

For expressions E and G, an expression of the form $E \Rightarrow G$ will often be proved in a number of steps by the introduction of intermediate expressions. For instance, we can prove $E \Rightarrow G$ by proving $E \equiv F$ and $F \Rightarrow G$ for some expression F. In order not to be forced to write down expressions like F twice, we record proofs like these as follows.

$$\begin{array}{ll} & E \\ = & \quad \{ \text{ hint why } E \equiv F \ \} \\ & F \\ \Rightarrow & \quad \{ \text{ hint why } F \Rightarrow G \ \} \\ & G \end{array}$$

0.1 Trace Theory

Let Ω be a (possibly infinite) set; its elements are called *symbols*. An *alphabet* is a subset of Ω. For alphabet A the set of finite-length sequences over A is denoted by A^*. Elements of Ω^* are called *traces*. A *trace set* is a set of traces. The *length* of trace t is denoted by $\ell(t)$. The empty trace is denoted by ϵ, and concatenation of traces is denoted by juxtaposition.

Trace t is *prefix* of trace u, denoted by $t \leq u$, when $(\exists\, v : v \in \Omega^* : tv = u)$. The *prefix-closure* of trace set V, denoted by $pref(V)$, is the trace set defined by

$$pref(V) = \{t, v : t \leq v \wedge v \in V : t\}.$$

Trace set V is *prefix-closed* when $pref(V) = V$. Trace t *projected* on alphabet A is the trace denoted by $t\lceil A$ obtained from t by deleting all occurrences of symbols not in A. That is, for trace t, symbol a, and alphabet A we have

$$\begin{array}{rcl} \epsilon\lceil A & = & \epsilon \\ ta\lceil A & = & (t\lceil A)a \quad \text{if } a \in A \\ ta\lceil A & = & t\lceil A \quad \text{if } a \notin A \end{array}$$

A *process* is a pair $\langle A, V \rangle$ where A is some alphabet and V is a non-empty prefix-closed trace set with $V \subseteq A^*$. For process T its alphabet is also denoted by $\mathrm{a}T$ and its trace set by $\mathrm{t}T$. As a slight abuse of notation we shall sometimes write $t \in T$ for $t \in \mathrm{t}T$. The collection of processes with alphabet A is denoted by $\mathcal{T}(A)$.

The *weave* of processes T and U is the process denoted by $T \,\mathrm{w}\, U$ and defined by

$$T \,\mathrm{w}\, U = \langle \mathrm{a}T \cup \mathrm{a}U, \{t : t \in (\mathrm{a}T \cup \mathrm{a}U)^* \wedge t\lceil \mathrm{a}T \in T \wedge t\lceil \mathrm{a}U \in U : t\}\rangle.$$

Weaving models parallel composition where interaction by means of the common symbols is symmetric and synchronous. The *projection* of process T on alphabet A is the process denoted by $T\lceil A$ and defined by

$$T\lceil A = \langle \mathrm{a}T \cap A, \{t : t \in T : t\lceil A\}\rangle.$$

Projection models abstraction of (internal) communications.

A *renaming* p of alphabet A is a one-to-one mapping from A into Ω such that

$$(\forall\, a : a \in A : p{\cdot}a \notin A).$$

A renaming is extended (where its domain permits) to operate on alphabets, traces, trace sets, and processes in the obvious way. Function application for renamings is written with a dot, i.e. $p{\cdot}a$ for $p(a)$. For example, for renaming p of alphabet A we have

$$p{\cdot}A = \{a : a \in A : p{\cdot}a\}$$

and, hence, A and $p{\cdot}A$ are disjoint alphabets.

0.2 The Equation

The processes of Trace Theory serve as the semantics of a program notation [5, Ch. 2]. The semantic function that maps a program text onto its associated process will be denoted by PR ([5] uses TR). Instead of giving a complete definition of the program notation's syntax and the semantic function PR we present just enough to describe the problem studied in this paper.

The simplest form of program text is a regular expression over Ω (excluding $\emptyset$), which is also called a *command*. The meaning of command E is defined by

$$PR(E) = \langle A, pref(\mathcal{L}(E)) \rangle,$$

where A is the set of symbols that occur in E and $\mathcal{L}(E)$ is the language (trace set) generated by E. We use the following operators in regular expressions (in order of increasing binding power): bar (|) for union, semicolon (;) for concatenation, and asterisk (*) for Kleene closure. For example, we have

$$PR(a \,|\, b\,; c^*) = \langle \{a, b, c\}, \{\varepsilon, a\} \cup \{i : i \geq 0 : b(c^i)\} \rangle.$$

Process R is *regular* when there exists a command E such that $R = PR(E)$.

A more complex program text, also called *component*, combines: a process defined previously by some other program text, a renaming, and a command. The exact syntax of components does not concern us here. We are interested in the way that the meaning of such a component is defined in terms of its three parts. Consider component c consisting of: previously defined component d, renaming p of $\mathrm{a}PR(d)$, and command E with $p{\cdot}\mathrm{a}PR(d) \subseteq \mathrm{a}PR(E)$. The meaning of c is defined by

$$PR(c) = (PR(E) \,\mathrm{w}\, p{\cdot}PR(d)) \lceil A,$$

where alphabet A equals $\mathrm{a}PR(E) \setminus p{\cdot}\mathrm{a}PR(d)$. The idea is that in component c processes $PR(E)$ and $PR(d)$ operate in parallel, being connected according to p. Alphabet A gives the external communications of c ($\mathrm{a}PR(c) = A$) and all communications with $PR(d)$ are internal. The latter means that $PR(d)$ is a local process of c; d is, therefore, called the *subcomponent* of c. The way the meaning of c is obtained from its constituents resembles the master/slave operator of [7].

Now imagine that we attempt to define a *recursive* component c that has itself as subcomponent, by requiring for subcomponent d that it has the same meaning as c, that is, $PR(d) = PR(c)$. The equation defining the meaning of c then becomes

$$PR(c) = (PR(E) \,\mathrm{w}\, p{\cdot}PR(c)) \lceil A. \tag{0}$$

There are a number of questions to be asked about this equation if it is to serve as a definition for the meaning of recursive components. Does the equation have a solution for $PR(c)$? Is there, possibly, more than one solution? If the solution is not unique, can we easily single out a specific (canonical) solution? Can we prove things about the canonical solution? Can the canonical solution be "implemented"? Can the canonical solution, for instance, be captured explicitly in a closed formula, maybe as some limit of "manageable" (finite, regular) approximations?

These are the questions that inspired this paper. We shall deal with a generalization of recursive equation (0). We do not restrict ourselves to regular processes in the role of $PR(E)$. Hence, we get the following problem formulation. Given alphabet A, renaming p of A, and process R in $\mathcal{T}(A \cup p{\cdot}A)$ analyze the recursive equation

$$S = (R \,\mathrm{w}\, p{\cdot}S)\lceil A, \tag{1}$$

for processes S in $\mathcal{T}(A)$. Since (1) is trivially satisfied as far as the alphabets of the processes are concerned, the problem can also be cast in a trace set form: Given alphabet A, renaming p of A, and trace set R with $R \subseteq (A \cup p{\cdot}A)^*$ analyze the recursive trace set equation

$$S = \{t : t \in R \wedge t\lceil p{\cdot}A \in p{\cdot}S : t\lceil A\}$$

for trace sets S with $S \subseteq A^*$.

1 Two Approaches to Recursive Equations

Let us generalize equation (1) even further. Define the function f from $\mathcal{T}(A)$ into $\mathcal{T}(A)$ by

$$f(S) = (R \,\mathrm{w}\, p{\cdot}S)\lceil A. \tag{2}$$

Equation (1) can now be rephrased as

$$f(S) = S,$$

that is, we are interested in fixed points of this function f.

Even more abstractly, suppose we have some set $\mathcal{D}$ and a function f from $\mathcal{D}$ into $\mathcal{D}$. We are interested in the equation $x{:}\, f(x) = x$ over $\mathcal{D}$, i.e. in fixed points of f. We give an overview of two ways in which further knowledge about $\mathcal{D}$ and f can be helpful to analyze this equation.

1.0 Complete Partial Orders (CPOs)

For the first approach it is necessary to find a *partial order* $\sqsubseteq$ on $\mathcal{D}$, that is, a reflexive, antisymmetric, and transitive relation $\sqsubseteq$ on $\mathcal{D}$. The structure $\langle \mathcal{D}, \sqsubseteq \rangle$ is then called a *poset*.

Let us assume we have found such a relation $\sqsubseteq$ on $\mathcal{D}$. This gives rise to the following notions for a subset X of $\mathcal{D}$:

- *least* element of X,
- *upper bound* of X,
- *least upper bound* of X, and
- X is a *chain*.

We assume that the reader is familiar with the first three notions (cf. [1]). If X, $X \subseteq \mathcal{D}$, has a least upper bound in $\mathcal{D}$ then this least upper bound is denoted by $\bigsqcup X$. A chain is a subset of $\mathcal{D}$ on which $\sqsubseteq$ induces a linear order. We call a poset *complete*, or a *CPO*, when every chain has a least upper bound in $\mathcal{D}$. Notice that $\emptyset$ is a chain and that $\bigsqcup \emptyset$ is the least element of the CPO; it is called *bottom* and denoted by $\bot$.

The above notions for subsets are extended to sequences by applying them to the subset $\{n : n \geq 0 : S_n\}$ for sequence S in $\mathcal{D}$. Sequence S in $\mathcal{D}$ is called *ascending* when $S_n \sqsubseteq S_{n+1}$ for all $n \geq 0$. Notice that in this case the set $\{n : n \geq 0 : S_n\}$ is a chain in $\mathcal{D}$.

Function $f{:}\mathcal{D} \to \mathcal{D}$ is *monotonic* on the poset when $x \sqsubseteq y$ implies $f(x) \sqsubseteq f(y)$ for every x and y in $\mathcal{D}$. Moreover, we call function f *upward continuous* when for every ascending sequence S in $\mathcal{D}$ that has a least upper bound, the least upper bound of $\{n : n \geq 0 : f(S_n)\}$ exists and

$$f(\bigsqcup\{n : n \geq 0 : S_n\}) = \bigsqcup\{n : n \geq 0 : f(S_n)\}$$

Abian–Brown's Fixed Point Theorem [0, Thm. 2] implies that a monotonic function on a CPO has a (unique) least fixed point. Furthermore, if f is upward continuous, then its least fixed point is

$$\bigsqcup\{n : n \geq 0 : f^n(\bot)\},$$

where f^n denotes the function f iterated n times.

Thus, if we can find a relation $\sqsubseteq$ on $\mathcal{D}$ such that $\langle \mathcal{D}, \sqsubseteq \rangle$ is a CPO and f is monotonic on this CPO, then the equation $x{:}\, f(x) = x$ has a (unique) least solution in $\mathcal{D}$. If f is also upward continuous, then this least solution can be approximated by iteration of f starting in $\bot$.

In the literature there is little agreement on the terminology regarding posets. Especially, completeness and continuity have been defined in many ways, often yielding subtly different concepts. When choosing between alternative definitions one faces three conflicting interests: the complexity of the concepts, their range of application, and the complexity of proofs about fixed points. Our choice yields fairly simple definitions for completeness and

continuity with wide applicability, but the proof for the existence of a least fixed point is complicated (it needs to be done only once, of course). The reader may wish to consult [6] for some pointers into the literature.

The *dual* of relation $\sqsubseteq$ on $\mathcal{D}$ is the relation $\sqsupseteq$ defined by

$$x \sqsupseteq y \equiv y \sqsubseteq x$$

for all x and y in $\mathcal{D}$. Notice that the dual of $\sqsupseteq$ is again $\sqsubseteq$. The dual of a notion defined in the context of the structure $\langle \mathcal{D}, \sqsubseteq \rangle$ is that same notion but now placed in the context of the dual structure $\langle \mathcal{D}, \sqsupseteq \rangle$. Table 0 lists pairs of notions that are dual. Notice that poset, chain, and monotonic are self-dual, but CPO is not. We may thus go back and forth between dual structures or we may stick with one and consider dual notions instead. We have preferred the latter, which explains why Subsection 3.1 is about *descending* sequences even though they have not been mentioned so far.

poset	–	poset
least	–	greatest
upper bound	–	lower bound
bottom	–	top
chain	–	chain
ascending	–	descending
monotonic	–	monotonic
upward continuous	–	downward continuous
CPO	–	every chain has greatest lower bound

Table 0: Dual notions

Even if one is only interested in functions from a poset into itself, then it may still be useful to generalize the notions of monotonicity and continuity to functions from one poset into another. The reason is that function composition maintains monotonicity and continuity. Thus monotonicity of a function from a poset into itself can be proven by decomposing it into a number of simpler functions mapping into (possibly) different intermediate posets and proving their monotonicity separately.

For example, if $\langle \mathcal{D}_0, \sqsubseteq_0 \rangle$ and $\langle \mathcal{D}_1, \sqsubseteq_1 \rangle$ are posets, then $f: \mathcal{D}_0 \to \mathcal{D}_1$ is called monotonic when

$$x \sqsubseteq_0 y \;\Rightarrow\; f(x) \sqsubseteq_1 f(y)$$

for all x and y in $\mathcal{D}_0$. If $g: \mathcal{D}_1 \to \mathcal{D}_0$ is also a monotonic function, then $g \circ f$ maps $\mathcal{D}_0$ monotonically into itself. The definitions of upward (and downward) continuity are extended in a similar way.

1.1 Complete Metric Spaces (CMSs)

In the second approach one needs to find a *metric* d on $\mathcal{D}$, that is, a function $d: \mathcal{D} \times \mathcal{D} \to \mathcal{R}$, where $\mathcal{R}$ is the set of real numbers, such that for all x, y, and z in $\mathcal{D}$

0. $d(x,y) \geq 0$,
1. $d(x,y) = 0 \;\equiv\; x = y$,
2. $d(x,y) = d(y,x)$, and
3. $d(x,y) \leq d(x,z) + d(z,y)$.

The structure $\langle \mathcal{D}, d \rangle$ is then called a *metric space*.

Let us assume that d is a metric on $\mathcal{D}$. This gives rise to the following notions for a sequence S in $\mathcal{D}$:

- *limit* of S, and
- S is a *Cauchy* sequence.

We assume that the reader is familiar with the first notion (cf. [4]). Sequence S is a Cauchy sequence when

$$(\forall \epsilon : \epsilon > 0 : (\exists N :: (\forall m, n : m \geq N \wedge n \geq N : d(S_m, S_n) < \epsilon))),$$

where ϵ ranges over the real numbers. A metric space is called *complete*, or a *CMS*, when every Cauchy sequence has a limit in $\mathcal{D}$.

Function $f: \mathcal{D} \to \mathcal{D}$ is a *contraction* on the metric space when there exists a real number c, $0 \leq c < 1$, such that

$$d(f(x), f(y)) \leq c * d(x,y)$$

for all x and y in $\mathcal{D}$. Banach's Contraction Theorem states that a contraction on a CMS has a unique fixed point and that for every x in $\mathcal{D}$ this fixed point equals the limit of the sequence S defined by $S_n = f^n(x)$.

Thus, if we can find a metric d on $\mathcal{D}$ such that $\langle \mathcal{D}, d \rangle$ is a CMS and f is a contraction on this CMS, then the equation $x: f(x) = x$ has a unique solution in $\mathcal{D}$ and it can be approximated by iteration of f starting anywhere in $\mathcal{D}$.

2 Finiteness of Trace Sets

In this section we have collected a number of statements concerning the finiteness of trace sets.

We write $\#V$ for the number of elements in set V. "Set V is finite" will also be expressed by the formula $\#V < \infty$. The *length* of trace set V is denoted by $\ell(V)$ and is defined by

$$\begin{aligned} \ell(\emptyset) &= 0, \\ \ell(V) &= (\mathbf{MAX}\, t : t \in V : \ell(t) + 1), \text{ for } V \neq \emptyset. \end{aligned}$$

Trace set V is called *bounded* when $\ell(V) < \infty$. Because traces have finite length, we obviously have

Property 0 For trace set V

$$\#V < \infty \;\Rightarrow\; \ell(V) < \infty,$$

that is, a finite trace set is bounded.

Furthermore, $\ell(t) < \ell(V)$ for every trace $t \in V$. Since $s \leq t$ implies $\ell(s) \leq \ell(t)$, we also see that $\ell(\mathit{pref}(V)) = \ell(V)$.

Property 1 For trace set V

$$\#V < \infty \;\equiv\; \#\mathit{pref}(V) < \infty,$$

that is, a trace set is finite if and only if its prefix-closure is finite.

Proof Observe that $V \subseteq \mathit{pref}(V)$ and that the number of prefixes of trace t equals $\ell(t) + 1$. Therefore, we have

$$\#V \;\leq\; \#\mathit{pref}(V) \;\leq\; \#V * \ell(V).$$

Finally, recall Property 0.
(End of Proof)

For trace set V and integer k, $k \geq 0$, the *k-th level* of V is the trace set denoted by $V^{[k]}$ and defined by

$$V^{[k]} = \{t : t \in V \wedge \ell(t) = k : t\}.$$

Trace set V is called *level-finite* when all its levels are finite.

Property 2 If trace set V is bounded, then

$$\#V < \infty \;\equiv\; (\forall k : k \geq 0 : \#V^{[k]} < \infty),$$

that is, a bounded trace set is finite if and only if it is level-finite.

Proof Observe that

$$V = (\bigcup k : 0 \leq k < \ell(V) : V^{[k]}).$$

(End of Proof)

Property 3 Prefix-closed trace set V is bounded if and only if

$$(\exists k : k \geq 0 : V^{[k]} = \emptyset).$$

Proof Let V be a prefix-closed trace set. If V is bounded, then taking $k = \ell(V)$ will do. Now assume that $V^{[k]}$ is empty for some $k \geq 0$. Since V is prefix-closed, $V^{[n]}$ is empty for all $n \geq k$ as well. Hence, V is bounded.
(End of Proof)

Trace set V has *finite fanout*, denoted by $\mathit{finfan}(V)$, when

$$(\forall t : t \in V : \#\{a : a \in \Omega \wedge ta \in V : ta\} < \infty).$$

Obviously, a finite trace set has finite fanout.

Property 4 Prefix-closed trace set V has finite fanout if and only if it is level-finite.

Proof For the if-part observe that for trace $t \in V$

$$\#\{a : a \in \Omega \wedge ta \in V : ta\} \leq \#V^{[\ell(t)+1]}.$$

The only-if-part follows by mathematical induction on k: $\#V^{[0]} \leq 1$, and for $k \geq 0$

$$V^{[k+1]} = (\bigcup t : t \in V^{[k]} : \{a : a \in \Omega \wedge ta \in V : ta\}),$$

because of the prefix-closedness of V.
(End of Proof)

Lemma 0 For prefix-closed trace set V

$$\ell(V) < \infty \;\wedge\; \mathit{finfan}(V) \;\Rightarrow\; \#V < \infty,$$

that is, a bounded prefix-closed trace set with finite fanout is finite.

Proof Use Properties 2 and 4.
(End of Proof)

Property 5 For trace set V and alphabets A and B such that $V \subseteq A^*$ and $A \setminus B$ is finite

$$\mathit{finfan}(V\lceil B) \;\Rightarrow\; \mathit{finfan}(V).$$

Proof Observe that on account of $V \subseteq A^*$ we have for trace $t \in V$

$$\{a : a \in \Omega \wedge ta \in V : a\} \subseteq \{a : a \in \Omega \wedge (t\lceil B)a \in (V\lceil B) : a\} \cup (A \setminus B).$$

(End of Proof)

It may be interesting to observe that, in general, the reverse implication does not hold in the above property.

Lemma 1 For trace set V and alphabets A and B such that $V \subseteq A^*$ and $A \setminus B$ is finite

$$\ell(V) < \infty \;\wedge\; \#(V\lceil B) < \infty \;\Rightarrow\; \#V < \infty.$$

Proof Let A and B be alphabets with $A \setminus B$ finite, and let V be a trace set with $V \subseteq A^*$. We derive

$$\begin{array}{ll}
& \ell(V) < \infty \;\wedge\; \#(V\lceil B) < \infty \\
= & \quad \{\ \ell(V) = \ell(\mathit{pref}(V)) \text{ and Property 1 }\} \\
& \ell(\mathit{pref}(V)) < \infty \;\wedge\; \#\mathit{pref}(V\lceil B) < \infty \\
= & \quad \{\ \mathit{pref}(V\lceil B) = \mathit{pref}(V)\lceil B\ \} \\
& \ell(\mathit{pref}(V)) < \infty \;\wedge\; \#(\mathit{pref}(V)\lceil B) < \infty \\
\Rightarrow & \quad \{\text{ a finite trace set has finite fanout }\} \\
& \ell(\mathit{pref}(V)) < \infty \;\wedge\; \mathit{finfan}(\mathit{pref}(V)\lceil B) \\
\Rightarrow & \quad \{\text{ Property 5, using finiteness of } A \setminus B\ \} \\
& \ell(\mathit{pref}(V)) < \infty \;\wedge\; \mathit{finfan}(\mathit{pref}(V)) \\
\Rightarrow & \quad \{\text{ Lemma 0 applied to } \mathit{pref}(V)\ \} \\
& \#\mathit{pref}(V) < \infty \\
= & \quad \{\text{ Property 1 }\} \\
& \#V < \infty
\end{array}$$

(End of Proof)

The following two lemmas deal with descending sequences of trace sets. Sequence V is *descending* when $V_{n+1} \subseteq V_n$ for all $n \geq 0$. Since the poset $\langle \mathcal{P}(X), \subseteq \rangle$ is well-founded—that is, every non-empty chain in it has a least element—if (and only if) X is a finite set, we have

Lemma 2 If V is a descending sequence of finite trace sets, then

$$(\exists\, i : i \geq 0 : (\bigcap n : n \geq 0 : V_n) = V_i),$$

and, thus

$$(\bigcap n : n \geq 0 : V_n) = \emptyset \;\equiv\; (\exists\, n : n \geq 0 : V_n = \emptyset).$$

Lemma 3 If V is a descending sequence of prefix-closed trace sets such that all V_n are level-finite, then

$$\#(\bigcap n : n \geq 0 : V_n) < \infty \;\equiv\; (\exists\, n : n \geq 0 : \#V_n < \infty).$$

Proof Define V_ω as $(\bigcap n : n \geq 0 : V_n)$. Observe that V_ω is prefix-closed and that

$$V_\omega^{[k]} = (\bigcap n : n \geq 0 : V_n^{[k]}).$$

Hence, all $V_\omega^{[k]}$ are finite. We now derive

$$\begin{array}{ll}
 & \#V_\omega < \infty \\
= & \quad \{ \text{ Properties 0 and 2, using the finiteness of } V_\omega^{[k]} \} \\
 & \ell(V_\omega) < \infty \\
= & \quad \{ \text{ Property 3, using the prefix-closedness of } V_\omega \} \\
 & (\exists\, k : k \geq 0 : V_\omega^{[k]} = \emptyset) \\
= & \quad \{ \text{ above observation } \} \\
 & (\exists\, k : k \geq 0 : (\bigcap n : n \geq 0 : V_n^{[k]}) = \emptyset) \\
= & \quad \{ \text{ Lemma 2, using the finiteness of } V_n^{[k]} \} \\
 & (\exists\, k : k \geq 0 : (\exists\, n : n \geq 0 : V_n^{[k]} = \emptyset)) \\
= & \quad \{ \text{ predicate calculus } \} \\
 & (\exists\, n : n \geq 0 : (\exists\, k : k \geq 0 : V_n^{[k]} = \emptyset)) \\
= & \quad \{ \text{ Property 3, using the prefix-closedness of } V_n \} \\
 & (\exists\, n : n \geq 0 : \ell(V_n) < \infty) \\
= & \quad \{ \text{ Properties 0 and 2, using the finiteness of } V_n^{[k]} \} \\
 & (\exists\, n : n \geq 0 : \#V_n < \infty)
\end{array}$$

(End of Proof)

3 The CPO Approach

In Subsection 3.0 a partial order on processes is defined, viz. based on the subset ordering of their trace sets. We briefly summarize earlier results for this partial order; for example, it is complete. Subsection 3.1 analyzes the projection operator with regard to descending sequences of processes. We characterize a subset of the process domain on which projection is downward continuous. Finally, Subsection 3.2 derives an approximation theorem for the greatest solution of equation (1).

3.0 The Partial Order and Some Earlier Results

For more details on the material of this subsection the reader is referred to [5, §1.6].

We define the relation $\subseteq$ on the process domain by

$$T \subseteq U \;\equiv\; \mathrm{a}T = \mathrm{a}U \;\wedge\; \mathrm{t}T = \mathrm{t}U$$

for all processes T and U. The relation $\subseteq$ is a partial order.

Let A be an alphabet. We denote the relation that $\subseteq$ induces in $\mathcal{T}(A)$ by the same name. The structure $\langle \mathcal{T}(A), \subseteq \rangle$ is a poset. It has bottom and top, viz. the processes $STOP(A)$ and $RUN(A)$ respectively, defined by

$$\begin{aligned} STOP(A) &= \langle A, \{\varepsilon\} \rangle, \\ RUN(A) &= \langle A, A^* \rangle. \end{aligned}$$

Least upper bound and greatest lower bound exist for every subset X of $\mathcal{T}(A)$, viz.

$$(\bigcup T : T \in X : T) = \begin{cases} STOP(A) & \text{if } X = \emptyset \\ \langle A, (\bigcup T : T \in X : \mathrm{t}T) \rangle & \text{if } X \neq \emptyset \end{cases}$$

and

$$(\bigcap T : T \in X : T) = \begin{cases} RUN(A) & \text{if } X = \emptyset \\ \langle A, (\bigcap T : T \in X : \mathrm{t}T) \rangle & \text{if } X \neq \emptyset \end{cases}$$

respectively. This makes $\langle \mathcal{T}(A), \subseteq \rangle$ and its dual CPOs. (In fact, they are even complete lattices.)

Weaving with a fixed process and renaming are upward and downward continuous, and hence monotonic. Projection is upward continuous, and

hence also monotonic. Since the function f defined by (2) is the composition of upward continuous functions it is itself upward continuous, and thus monotonic. Application of the Fixed Point Theorem then tells us that f has a least and a greatest fixed points. Furthermore, from the upward continuity of f we infer that the least fixed point of f is, in fact,

$$(\bigcup n : n \geq 0 : f^n(STOP(A))). \tag{3}$$

The following "standard" counterexample shows that, in general, projection is not downward continuous.

Example 0 Define sequence T by

$$T_n = \langle \{a,b\}, pref\{i : i \geq n : a^i b\} \rangle$$

and take $B = \{b\}$. Then we have:

$$\begin{aligned} T_\omega &= \langle \{a,b\}, \{i : i \geq 0 : a^i\} \rangle \\ T_\omega \lceil B &= \langle \{b\}, \{\varepsilon\} \rangle \\ T_n \lceil B &= \langle \{b\}, \{\varepsilon, b\} \rangle \quad \text{for } n \geq 0 \\ (\bigcap n : n \geq 0 : T_n \lceil B) &= \langle \{b\}, \{\varepsilon, b\} \rangle \end{aligned}$$

Hence, $T_\omega \lceil B \neq (\bigcap n : n \geq 0 : T_n \lceil B)$.

(End of Example)

Because projection is not downward continuous, we cannot infer an approximation result for the greatest fixed point of f. That, in general, the greatest fixed point of function f is *not*

$$(\bigcap n : n \geq 0 : f^n(RUN(A))), \tag{4}$$

is shown by the next example (cf. the example following Thm. 3.0 in [9]).

Example 1 Define alphabet A as $\{a,b\}$ and define process U by

$$U = \langle A \cup p{\cdot}A, pref\{i : i \geq 0 : (p{\cdot}a)^i (p{\cdot}b) a^{i+1} b\} \rangle.$$

For $n \geq 1$ we now have

$$f^n(RUN(A)) = \langle A, pref\{i : i \geq n : a^i b\} \rangle$$

and, thus,

$$(\bigcap n : n \geq 0 : f^n(RUN(A))) = \langle A, \{i : i \geq 0 : a^i\} \rangle.$$

This is not a fixed point since applying f once more yields $STOP(A)$, which turns out to be the unique fixed point of f.

Notice that the sequence of approximations $f^n(RUN(A))$ is the same as the sequence T from Example 0.

(End of Example)

In the preceding example U is not a regular process. The following example is based on a regular process U.

Example 2 Consider alphabet $A = \{a, b, c\}$ and process U defined by

$$U = PR((a\,;p{\cdot}a)^*;a\,;p{\cdot}b\,;b\,|(p{\cdot}a)^*;p{\cdot}b\,;c).$$

For $n \geq 1$ we have

$$f^n(RUN(A)) = PR(a^*;a^n;b\,|\,c).$$

Thus

$$(\bigcap n : n \geq 0 : f^n(RUN(A))) = \langle A, \{i : i \geq 0 : a^i\} \cup \{c\}\rangle,$$

which is not a fixed point. Applying f once more yields the (unique) fixed point $\langle A, \{i : i \geq 0 : a^i\}\rangle$.

The process can be slightly modified to reduce the size of the alphabet by taking alphabet $A = \{a, b\}$ and

$$U = PR((a\,;p{\cdot}a)^*;a\,;p{\cdot}b\,;b\,|(p{\cdot}a)^*;p{\cdot}a\,;p{\cdot}b\,;b).$$

(End of Example)

3.1 Projection and Descending Sequences of Processes

In this subsection we investigate distribution of projection over the intersection (greatest lower bound) of descending sequences, because that will give a clearer picture of why projection is not downward continuous.

In this subsection A is an alphabet, T is a descending sequence in $\mathcal{T}(A)$, and B, $B \subseteq A$, is an alphabet such that $A \setminus B$ is finite. The process T_ω is defined as $(\bigcap n : n \geq 0 : T_n)$. From

$$\begin{array}{ll} & \text{true} \\ = & \quad \{\ T_\omega \text{ is a lower bound of the sequence } T\ \} \\ & (\forall\, n : n \geq 0 : T_\omega \subseteq T_n) \end{array}$$

$\Rightarrow$ { projection is monotonic }
$(\forall n : n \geq 0 : T_\omega \lceil B \subseteq T_n \lceil B)$
$=$ { property of greatest lower bound }
$T_\omega \lceil B \subseteq (\bigcap n : n \geq 0 : T_n \lceil B)$

we infer

$$(\bigcap n : n \geq 0 : T_n) \lceil B \ \subseteq \ (\bigcap n : n \geq 0 : T_n \lceil B). \tag{5}$$

We are interested in conditions under which equality holds in (5). We shall formulate one necessary and sufficient condition and three sufficient conditions for equality in (5). This will also enable us to find a condition under which (4) does yield the greatest fixed point.

We now introduce some auxiliary concepts. For trace u, process T, and alphabet C, the trace set $orig(u, T, C)$ is defined by

$$orig(u, T, C) = \{t : t \in T \wedge t \lceil C = u : t\}.$$

Obvious properties of *orig* are:

$$orig(u, T, C) \lceil C \subseteq \{u\},$$

$$orig(u, T, C) \neq \emptyset \ \equiv \ u \in T \lceil C, \tag{6}$$

$$T \subseteq T' \ \Rightarrow \ orig(u, T, C) \subseteq orig(u, T', C). \tag{7}$$

In the sequel we shall write $orig(u, T)$ as short for $orig(u, T, B)$.

Lemma 4 For all traces u we have

$$orig(u, T_\omega) = (\bigcap n : n \geq 0 : orig(u, T_n)).$$

Proof We derive

$orig(u, T_\omega)$
$=$ { definition of T_ω }
$orig(u, (\bigcap n : n \geq 0 : T_n))$
$=$ { definition of *orig* }
$\{t : t \in \mathrm{t}(\bigcap n : n \geq 0 : T_n) \wedge t \lceil B = u : t\}$
$=$ { definition of $\bigcap$ }
$\{t : (\forall n : n \geq 0 : t \in T_n) \wedge t \lceil B = u : t\}$

$$
\begin{array}{ll}
= & \{ \text{ predicate calculus, using } (\exists\, n :: n \geq 0) \,\} \\
& \{t : (\forall\, n : n \geq 0 : t \in T_n \wedge t\lceil B = u) : t\} \\
= & \{ \text{ definition of } \bigcap \,\} \\
& (\bigcap n : n \geq 0 : \{t : t \in T_n \wedge t\lceil B = u : t\}) \\
= & \{ \text{ definition of } \mathit{orig} \,\} \\
& (\bigcap n : n \geq 0 : \mathit{orig}(u, T_n))
\end{array}
$$

(End of Proof)

For trace u, process T, and alphabet C, the number $\mathit{minorig}(u, T, C)$ is defined by

$$\mathit{minorig}(u, T, C) = (\text{MIN}\, t : t \in \mathit{orig}(u, T, C) : \ell(t)).$$

Notice that $\mathit{minorig}(u, T, C) = \infty$ if and only if $\mathit{orig}(u, T, C) = \emptyset$, that is, iff $u \notin T\lceil C$. Again, we write $\mathit{minorig}(u, T)$ for $\mathit{minorig}(u, T, B)$. For instance, in Example 0 we have $\mathit{minorig}(b, T_n) = \ell(a^n b) = n + 1$.

For convenience, we define the sequence U and process U_ω by $U_n = T_n\lceil B$ and $U_\omega = (\bigcap n : n \geq 0 : U_n)$. Thus, (5) expresses that $T_\omega\lceil B \subseteq U_\omega$. We also adopt the following convention for naming traces. Traces t and t_i belong to trace structures with alphabet A (like T_n and T_ω), and traces u, u_i, and v belong to trace structures with alphabet B (like U_n and U_ω).

Theorem 0 Equality holds in (5) if and only if

$$(\forall\, u : u \in U_\omega : (\exists\, K :: (\forall\, n : n \geq 0 : \mathit{minorig}(u, T_n) \leq K))). \tag{8}$$

Proof We show the two implications separately.

Only if) Assume equality holds in (5), that is, $T_\omega\lceil B = U_\omega$. Let $u \in U_\omega$. Hence, $u \in T_\omega\lceil B$. On account of (6), take some $t \in \mathit{orig}(u, T_\omega)$. From Lemma 4 we then infer $(\forall\, n : n \geq 0 : t \in \mathit{orig}(u, T_n))$. Therefore, taking $K = \ell(t)$ suffices as upper bound. This proves (8).

If) Assume condition (8) holds. Let $u \in U_\omega$. On account of assumption (8) let K be an upper bound of $\{n : n \geq 0 : \mathit{minorig}(u, T_n)\}$. Consider the sequence of trace sets S defined by

$$S_n = \{t : t \in \mathit{orig}(u, T_n) \wedge \ell(t) \leq K : t\}.$$

Since $A \setminus B$ is finite, $\ell(S_n) \leq K + 1$, and $S_n\lceil B = \{u\}$ is finite, Lemma 1 tells us that each S_n is finite. Furthermore, this sequence is descending by (7). We now derive

$$
\begin{array}{ll}
 & \text{true} \\
= & \quad \{ \text{ choice of } K \ \} \\
 & (\forall n : n \geq 0 : S_n \neq \emptyset) \\
= & \quad \{ \text{ Lemma 2 using the finiteness of all } S_n \text{ and that } S \text{ is descending } \} \\
 & (\bigcap n : n \geq 0 : S_n) \neq \emptyset \\
\Rightarrow & \quad \{ \ S_n \subseteq orig(u, T_n) \text{ for all } n \ \} \\
 & (\bigcap n : n \geq 0 : orig(u, T_n)) \neq \emptyset \\
= & \quad \{ \text{ Lemma 4 } \} \\
 & orig(u, T_\omega) \neq \emptyset \\
= & \quad \{ \ (6) \ \} \\
 & u \in T_\omega \lceil B
\end{array}
$$

This proves equality in (5).
(End of Proof)

The following example shows that (8) is not a sufficient condition for equality in (5) if $A \setminus B$ is allowed to be infinite.

Example 3 Take $A = \{i : i \geq 0 : a_i\} \cup \{b\}$, define the sequence T by

$$T_n = \langle A, pref\{i : i \geq n : a_i b\}\rangle$$

and take $B = \{b\}$. Then we have:

$$
\begin{array}{rcl}
T_\omega & = & \langle A, \{\varepsilon\}\rangle \\
T_\omega \lceil B & = & \langle B, \{\varepsilon\}\rangle \\
T_n \lceil B & = & \langle B, \{\varepsilon, b\}\rangle \quad \text{for } n \geq 0 \\
U_\omega & = & \langle B, \{\varepsilon, b\}\rangle
\end{array}
$$

Thus, $T_\omega \lceil B \neq U_\omega$, but (8) is satisfied, because we also have:

$$
\begin{array}{rcl}
minorig(\varepsilon, T_n) & = & 0, \\
minorig(b, T_n) & = & \ell(a_n b) \ = \ 2.
\end{array}
$$

Notice that $orig(b, T_n)$ is infinite for every $n \geq 0$.
(End of Example)

Theorem 1 Equality holds in (5) if

$$(\forall u :: (\exists n : n \geq 0 : \#orig(u, T_n) < \infty)). \tag{9}$$

Proof We show that (9) implies (8), and apply Theorem 8. Assume (9). Let u be a trace in U_ω. On account of (9) let i, $i \geq 0$, be such that $\#orig(u,T_i) < \infty$. Now define K as $(\mathrm{MAX}t : t \in orig(u,T_i) : \ell(t))$. Because the sequence S defined by $S_n = orig(u,T_n)$ is descending, this K suffices as upper bound in (8).
(End of Proof)

Example 4 That (9) is not a necessary condition for equality in (5), follows from the constant sequence T defined by

$$T_n = \langle \{a,b\}, \{i : i \geq 0 : a^i\} \rangle$$

with $B = \{b\}$. This sequence does not satisfy (9), but, of course, we do have equality in (5).
(End of Example)

Before we strengthen (9) we introduce another auxiliary concept. We say "process T diverges when projected on alphabet C", denoted by $div(T,C)$, when

$$(\exists u :: \#orig(u,T,C) = \infty).$$

If $\mathrm{a}T \setminus C$ is finite, then $div(T,C)$ is equivalent to $livelockfree(\mathrm{a}T \setminus C, T)$ as defined in [5, Ch. 5]. In case $\mathrm{a}T \setminus C$ is infinite, this equivalence does not hold as is shown by projection of process T_0 on alphabet B from Example 3.

From (7) we infer

$$T \subseteq T' \wedge div(T,C) \;\Rightarrow\; div(T',C). \tag{10}$$

We write $div(T)$ as short for $div(T,B)$.

Theorem 2 Equality holds in (5) if

$$(\exists n : n \geq 0 : \neg div(T_n)). \tag{11}$$

Proof We show that (11) implies (9), and apply Theorem 9:

$$\begin{array}{ll} & (\exists n : n \geq 0 : \neg div(T_n)) \\ = & \quad \{ \text{ definition of } div \ \} \\ & (\exists n : n \geq 0 : (\forall u :: \#orig(u,T_n) < \infty)) \end{array}$$

$\Rightarrow$ { predicate calculus }
$(\forall u :: (\exists n : n \geq 0 : \#orig(u, T_n) < \infty))$

(End of Proof)

That (11) is strictly stronger than (9) follows from

Example 5 Define sequence T by

$$T_n = \langle \{a, b\}, pref\{i, j : i \geq n \wedge j \geq 0 : b^i a^j\} \rangle$$

and take $B = \{b\}$. This sequence satisfies (9), because for $n = \ell(u) + 1$ we have $\#orig(u, T_n) \leq 1$. But it does not satisfy (11), since every T_n diverges when projected on B. Notice however that

$$T_\omega = \langle \{a, b\}, \{i : i \geq 0 : b^i\} \rangle,$$

hence, T_ω does not diverge when projected on B. This inspired Theorem 3 below.

(End of Example)

Property 6 For trace u, process T, and alphabet C, the trace set

$$(\bigcup v : v \leq u : orig(v, T, C))$$

equals the trace set

$$\{t : t \in T \wedge (\exists v : v \leq u : t\lceil C = v) : t\}$$

and it is prefix-closed.

Proof We derive

$t \in (\bigcup v : v \leq u : orig(v, T, C))$
$=$ { definition of $\bigcup$ and $orig$ }
$(\exists v : v \leq u : t \in T \wedge t\lceil C = v)$
$=$ { predicate calculus }
$t \in T \wedge (\exists v : v \leq u : t\lceil C = v)$

and

$st \in T \wedge (\exists v : v \leq u : st\lceil C = v)$
$=$ { property of projection }

$$
\begin{array}{ll}
& st \in T \ \wedge \ (\exists\, v_0, v_1 : v_0 v_1 \leq u : s\lceil C = v_0 \wedge t\lceil C = v_1) \\
\Rightarrow & \quad \{\ \mathrm{t}T \text{ is prefix-closed, transitivity of } \leq, \text{ and pred. calc. } \} \\
& s \in T \ \wedge \ (\exists\, v_0 : v_0 \leq u : s\lceil C = v_0)
\end{array}
$$

(End of Proof)

Theorem 3 Equality holds in (5) if $\neg div(T_\omega)$.

Proof We show that $\neg div(T_\omega)$ implies (9) in order to apply Theorem 1. Given a trace u, define the sequence of trace sets S and process S_ω by

$$
\begin{array}{lcl}
S_n & = & (\bigcup v : v \leq u : orig(v, T_n)) \\
S_\omega & = & (\bigcup v : v \leq u : orig(v, T_\omega))
\end{array}
$$

From Property 6 we know that these trace sets are prefix-closed. Analogous to Lemma 4, using Property 6, one can prove

$$S_\omega = (\bigcap n : n \geq 0 : S_n). \tag{12}$$

Furthermore, $S_n^{[k]}$ is bounded and $S_n^{[k]}\lceil B = pref\{u\}$ is finite. Hence, by Lemma 1 all $S_n^{[k]}$ are finite. We now derive

$$
\begin{array}{ll}
& \neg div(T_\omega) \\
= & \quad \{\ \text{definition of } div\ \} \\
& (\forall\, v :: \#orig(v, T_\omega) < \infty) \\
= & \quad \{\ \text{a trace has a finite number of prefixes, and set theory}\ \} \\
& (\forall\, u :: \#(\bigcup v : v \leq u : orig(v, T_\omega)) < \infty) \\
= & \quad \{\ \text{definition of } S_\omega\ \} \\
& (\forall\, u :: \#S_\omega < \infty) \\
= & \quad \{\ (12)\ \} \\
& (\forall\, u :: \#(\bigcap n : n \geq 0 : S_n) < \infty) \\
= & \quad \{\ \text{Lemma 3, using that } S \text{ is a descending sequence of level-finite} \\
& \quad\ \ \text{prefix-closed trace sets}\ \} \\
& (\forall\, u :: (\exists\, n : n \geq 0 : \#S_n < \infty)) \\
= & \quad \{\ \text{definition of } S_n\ \} \\
& (\forall\, u :: (\exists\, n : n \geq 0 : \#(\bigcup v : v \leq u : orig(v, T_n)) < \infty)) \\
= & \quad \{\ \text{set theory, (6), and } T \text{ is descending}\ \} \\
& (\forall\, u :: (\exists\, n : n \geq 0 : \#orig(u, T_n) < \infty))
\end{array}
$$

(End of Proof)

By the way, notice that we proved equivalence of (9) and $\neg div(T_\omega)$ instead of just the desired implication. Let us summarize the relationships between the various conditions for equality in (5) as expressed by the theorems of this section. Keep in mind that finiteness of $A \setminus B$ is a global assumption.

$$
\begin{array}{ll}
 & (\exists\, n : n \geq 0 : \neg div(T_n)) \\
\Rightarrow & \quad \{\ (10), \text{ using } T_\omega \subseteq T_n \text{ for all } n\ \} \\
 & \neg div(T_\omega) \\
= & \quad \{\ \text{proof of Theorem 3}\ \} \\
 & (\forall\, u : u \in U_\omega : (\exists\, n : n \geq 0 : \#orig(u, T_n) < \infty)) \\
\Rightarrow & \quad \{\ \text{proof of Theorem 1}\ \} \\
 & (\forall\, u : u \in U_\omega : \#\{n : n \geq 0 : minorig(u, T_n)\} < \infty) \\
= & \quad \{\ \text{Theorem 0}\ \} \\
 & T_\omega \lceil B = U_\omega
\end{array}
$$

Theorem 3 can also be interpreted as expressing that projection on alphabet B is downward continuous "in" processes that do not diverge when projected on B. An alternative proof of this can be based on a slightly different version of projection as used in the mathematical model for CSP [2, Section 2.8]. For process T and alphabet B the process $T \lfloor B$ is defined by

$$
\begin{array}{lcl}
\mathrm{a}(T \lfloor B) & = & \mathrm{a}T \cap B \\
\mathrm{t}(T \lfloor B) & = & \{t : t \in T : t \lceil B\} \cup \\
 & & \{u, v : \#orig(u, T) = \infty \wedge v \in B^* : uv\}
\end{array}
$$

Obviously, $T \lceil B \subseteq T \lfloor B$ and if $\neg div(T)$ then $T \lceil B = T \lfloor B$. Furthermore, it is known that $\lfloor$-projection is downward continuous. Hence, we can derive

$$
\begin{array}{ll}
 & (\bigcap n : n \geq 0 : T_n \lceil B) \\
\subseteq & \quad \{\ T_n \lceil B \subseteq T_n \lfloor B\ \} \\
 & (\bigcap n : n \geq 0 : T_n \lfloor B) \\
= & \quad \{\ \lfloor\text{-projection is downward continuous}\ \} \\
 & (\bigcap n : n \geq 0 : T_n) \lfloor B \\
= & \quad \{\ \text{definition of } T_\omega\ \} \\
 & T_\omega \lfloor B
\end{array}
$$

$$= \quad \{ \neg div(T_\omega) \ \} \\ T_\omega \lceil B$$

proving Theorem 3.

Finally, we note that some of the conditions can be weakened. For instance, finiteness of $A \setminus B$ can be replaced by "finite fanout on $A \setminus B$". The condition of Theorem 3, i.e. $\neg div(T_\omega)$, can be weakened to

$$(\forall u :: \#(orig(u, T_\omega) \cap \{t, a : a \in \Omega \wedge ta \in T_\omega : ta\}) < \infty).$$

3.2 Application

In this subsection we give a condition under which the greatest fixed point of equation (1) is obtained as the limit of the approximations starting from the top element.

Property 7 Let T and U be processes such that $\mathrm{a}T \subseteq \mathrm{a}U$, and let C be an alphabet such that $\neg div(U, C)$. Then we have $\neg div(U \,\mathrm{w}\, T, C)$.

Proof Observe that $U \,\mathrm{w}\, T \subseteq U$, and apply (10).
(End of Proof)

Let A be a finite alphabet, p a renaming of A, and R a process in $\mathcal{T}(A \cup p{\cdot}A)$. The function $f{:}\mathcal{T}(A) \to \mathcal{T}(A)$ is defined by

$$f(S) = (R \,\mathrm{w}\, p{\cdot}S) \lceil A.$$

Property 8 If $S \in \mathcal{T}(A)$ is a fixed point of the function f, then S is at most

$$(\bigcap n : n \geq 0 : f^n(RUN(A))). \tag{13}$$

Proof Since $RUN(A)$ is the top of $\mathcal{T}(A)$ we have for any fixed point S of f that $S \subseteq RUN(A)$. From the monotonicity of f then follows that for all $n \geq 0$ we have

$$S = f^n(S) \subseteq f^n(RUN(A)).$$

Thus, $S \subseteq (\bigcap n : n \geq 0 : f^n(RUN(A)))$.
(End of Proof)

Theorem 4 If $\neg div(R, A)$ holds, then the greatest fixed point of the function f is given by (13).

Proof On account of Property 8 it is sufficient to show that (13) is a fixed point of f. We derive

$$
\begin{array}{ll}
& f(\bigcap n : n \geq 0 : f^n(RUN(A))) \\
= & \quad \{ \text{ definition of } f \ \} \\
& (U \,\mathrm{w}\, p{\cdot}(\bigcap n : n \geq 0 : f^n(RUN(A))))\lceil A \\
= & \quad \{ \text{ weaving and renaming are downward continuous } \} \\
& (\bigcap n : n \geq 0 : U \,\mathrm{w}\, p{\cdot}f^n(RUN(A)))\lceil A \\
\doteq & \quad \{ \text{ Theorem 2 using Property 7 and } \neg div(R,A) \ \} \\
& (\bigcap n : n \geq 0 : (U \,\mathrm{w}\, p{\cdot}f^n(RUN(A)))\lceil A) \\
= & \quad \{ \text{ definition of } f \ \} \\
& (\bigcap n : n \geq 0 : f^{n+1}(RUN(A))) \\
= & \quad \{ \ f \text{ is monotonic } \} \\
& (\bigcap n : n \geq 0 : f^n(RUN(A)))
\end{array}
$$

(End of Proof)

Example 6 The condition $\neg div(R, A)$ is not a necessary condition in Theorem 4. That is, (13) can be the greatest fixed point even if the condition is not satisfied. Take, for instance, $A = \{a\}$ and

$$R = PR((p{\cdot}a)^*; a^*).$$

In this case we have

$$f(STOP(A)) = RUN(A) = f(RUN(A)).$$

Thus, (13) equals $RUN(A)$, which is the unique fixed point of f.
(End of Example)

The preceding theorem can be generalized to

Theorem 5 If $\neg div(R, A)$ holds, then for any process T in $\mathcal{T}(A)$

$$(\bigcap n : n \geq 0 : (\bigcup m : m \geq n : f^m(T))) \tag{14}$$

is a fixed point of the function f.

Proof Define the sequence U by

$$U_n = R \,\mathrm{w}\, p{\cdot}(\bigcup m : m \geq n : f^m(T)).$$

This sequence is descending and from Property 7 using $\neg div(R, A)$ we infer $\neg div(U_n, A)$. We now derive

$$
\begin{array}{ll}
 & f(\bigcap n : n \geq 0 : (\bigcup m : m \geq n : f^m(T))) \\
= & \quad \{ \text{ definition of } f \ \} \\
 & (R \,\mathrm{w}\, p \cdot (\bigcap n : n \geq 0 : (\bigcup m : m \geq n : f^m(T))))\lceil A \\
= & \quad \{ \text{ renaming and weaving are downward continuous } \} \\
 & (\bigcap n : n \geq 0 : R \,\mathrm{w}\, p \cdot (\bigcup m : m \geq n : f^m(T)))\lceil A \\
= & \quad \{ \text{ definition of } U \ \} \\
 & (\bigcap n : n \geq 0 : U_n)\lceil A \\
= & \quad \{ \text{ Theorem 2 using that } U \text{ is descending and } \neg div(U_n, A) \ \} \\
 & (\bigcap n : n \geq 0 : U_n\lceil A) \\
= & \quad \{ \text{ definition of } U \ \} \\
 & (\bigcap n : n \geq 0 : (R \,\mathrm{w}\, p \cdot (\bigcup m : m \geq n : f^m(T)))\lceil A) \\
= & \quad \{ \text{ renaming, weaving, and projection are upward continuous } \} \\
 & (\bigcap n : n \geq 0 : (\bigcup m : m \geq n : (R \,\mathrm{w}\, p \cdot f^m(T))\lceil A)) \\
= & \quad \{ \text{ definition of } f \ \} \\
 & (\bigcap n : n \geq 0 : (\bigcup m : m \geq n : f^{m+1}(T))) \\
= & \quad \{ \text{ renaming dummies } \} \\
 & (\bigcap n : n \geq 1 : (\bigcup m : m \geq n : f^m(T))) \\
= & \quad \{ \text{ monotonicity } \} \\
 & (\bigcap n : n \geq 0 : (\bigcup m : m \geq n : f^m(T)))
\end{array}
$$

(End of Proof)

4 The CMS Approach

In Subsection 4.0 we define a distance function for processes and show that it is a metric. The resulting metric space is proven to be complete and compact. The relationship with the partial order of the preceding section is briefly discussed. Subsection 4.1 applies this to equation (1). We give a condition under which the associated function is a contraction on the metric space. This yields an approximation theorem for solutions of equation (1).

In this section A is an alphabet.

4.0 The Metric

For $T \in \mathcal{T}(A)$ and $U \in \mathcal{T}(A)$ we define the distance $d(T,U)$ between T and U by

$$d(T,U) = 2^{-(\mathrm{MIN}\, t : t \in \mathrm{t}T \div \mathrm{t}U : \ell(t))},$$

where $X \div Y$ denotes the symmetric difference of sets X and Y, that is, $X \setminus Y \cup Y \setminus X$. This notion of distance is similar to the Golson distance defined in [3] for synchronization trees.

Before we show that d is a metric on $\mathcal{T}(A)$, it is useful to have another characterization of the distance between two processes. For process T and natural number k, the *k-th approximation* of T is the process denoted by $T^{(k)}$ and defined by

$$T^{(k)} = \langle \mathrm{a}T, \{t : t \in T \wedge \ell(t) \leq k : t\} \rangle.$$

The process $T^{(k)}$ is also called a finite approximation of T. For example, we have $T^{(0)} = STOP(\mathrm{a}T)$. Furthermore, process T is the least upper bound (in the sense of the preceding section) of its finite approximations, that is,

$$T = (\bigcup k : k \geq 0 : T^{(k)}).$$

Property 9 For processes T and U in $\mathcal{T}(A)$ we have

$$(\mathrm{MIN}\, t : t \in \mathrm{t}T \div \mathrm{t}U : \ell(t)) \;=\; (\mathrm{MIN}\, k : T^{(k)} \neq U^{(k)} : k).$$

From the prefix-closedness of k-th approximations then follows that for any natural number k

$$d(T,U) < 2^{-k} \;\equiv\; T^{(k)} = U^{(k)}.$$

Theorem 6 $\langle \mathcal{T}(A), d \rangle$ is a metric space.

Proof Axioms 0, 1, and 2 for a metric (cf. Section 1.1) are trivially satisfied. We now prove Axiom 3 (triangle inequality). Let T, U, and V be processes in $\mathcal{T}(A)$. We have to show $d(T,U) \leq d(T,V) + d(V,U)$. Without loss of generality we assume $T \neq U$. On account of this let k be the smallest natural number such that $T^{(k)} \neq U^{(k)}$; hence, we have $d(T,U) = 2^{-k}$. We now derive

$$\begin{array}{ll} & \mathit{true} \\ = & \quad \{ \text{ calculus } \} \end{array}$$

$$
\begin{array}{ll}
& T^{(k)} = V^{(k)} \ \vee \ T^{(k)} \neq V^{(k)} \\
\Rightarrow & \quad \{\ T^{(k)} \neq U^{(k)}\ \} \\
& V^{(k)} \neq U^{(k)} \ \vee \ T^{(k)} \neq V^{(k)} \\
\Rightarrow & \quad \{\ \text{Property 9}\ \} \\
& d(V,U) \geq 2^{-k} \ \vee \ d(T,V) \geq 2^{-k} \\
= & \quad \{\ d(T,U) = 2^{-k} \text{ and Axiom 0 for a metric}\ \} \\
& d(T,U) \leq d(T,V) + d(V,U)
\end{array}
$$

(End of Proof)

Theorem 7 $\langle \mathcal{T}(A), d \rangle$ is a CMS.

Proof We show that every Cauchy sequence has a limit in $\mathcal{T}(A)$. Let T be a Cauchy sequence in $\mathcal{T}(A)$. Define the sequence U and process L by

$$
\begin{array}{rcl}
U_i & = & (\bigcup j : j \geq i : T_j) \quad \text{for } i \geq 0, \text{ and} \\
L & = & (\bigcap i : i \geq 0 : U_i).
\end{array}
$$

Obviously, the processes U_i are in $\mathcal{T}(A)$, and hence so is L. We show that L is the limit of sequence T by showing that for any natural number k there exists a natural number N such that for all natural n, $n \geq N$, we have $d(L, T_n) < 2^{-k}$.

Let k be a natural number and let natural number N be such that $d(T_m, T_n) < 2^{-k}$ for all m and n at least N. Such an N exists, since T is a Cauchy sequence. We show for $n \geq N$ that $L^{(k)} = T_n^{(k)}$, which then implies $d(L, T_n) < 2^{-k}$ on account of Property 9. From the choice of N together with Property 9 we infer

$$(\forall\, m, n : m \geq N \wedge n \geq N : T_m^{(k)} = T_n^{(k)}). \tag{15}$$

Now we derive for any $n \geq N$

$$
\begin{array}{ll}
& L^{(k)} \\
= & \quad \{\ \text{definition of } L \text{ and property of } k\text{-th approximation}\ \} \\
& (\bigcap i : i \geq 0 : U_i^{(k)}) \\
= & \quad \{\ U \text{ is a descending sequence of processes}\ \} \\
& (\bigcap i : i \geq N : U_i^{(k)}) \\
= & \quad \{\ \text{definition of } U_i \text{ and property of } k\text{-th approximation}\ \}
\end{array}
$$

$$
\begin{array}{ll}
 & (\bigcap i : i \geq N : (\bigcup j : j \geq i : T_j^{(k)})) \\
= & \quad \{ \ (15) \ \} \\
 & (\bigcap i : i \geq N : (\bigcup j : j \geq i : T_n^{(k)})) \\
= & \quad \{ \text{ intersection and union are idempotent } \} \\
 & T_n^{(k)}
\end{array}
$$

(End of Proof)

Although not strictly necessary for our current analysis, we also know something about the compactness of our metric space. A metric space is compact when every sequence in it has a converging subsequence. Sequence S is a subsequence of sequence T when there exists a monotonic one-to-one mapping g of the set of natural numbers into itself such that $T \circ g = S$.

Theorem 8 If alphabet A is finite then $\langle \mathcal{T}(A), d \rangle$ is compact.

Proof Assume A is finite and let T be a sequence in $\mathcal{T}(A)$. We first define a sequence U of subsequences of T. Sequence $U(0)$ is just T, and for $k \geq 1$ we obtain $U(k)$ as a subsequence of $U(k-1)$ for which the k-th approximations of the processes are all equal. This is possible because there exist only a finite number of sets of traces over A of length at most k (A being finite). Now define the sequence V by $V_n = U(n)_n$. Then V is a subsequence of T and it is a Cauchy sequence; hence, it converges by Theorem 7.

(End of Proof)

Example 7 When A is infinite, $\langle \mathcal{T}(A), d \rangle$ is not compact. Consider the infinite alphabet $A = \{i : i \geq 0 : a_i\}$ and the sequence T in $\mathcal{T}(A)$ defined by

$$T_n = \langle A, \{i : i \leq n : a_i\} \rangle.$$

This sequence has no converging subsequence since

$$m \neq n \ \Rightarrow \ d(T_m, T_n) = \tfrac{1}{2},$$

for all natural numbers m and n.

(End of Example)

For finite alphabet A we can also express the following relationship between the partial order as defined in Section 3 and the metric. If T is an

ascending or descending sequence in $\langle \mathcal{T}(A), \subseteq \rangle$ then it is a converging sequence in $\langle \mathcal{T}(A), d \rangle$ and its least upper bound in $\langle \mathcal{T}(A), \subseteq \rangle$ equals its limit in $\langle \mathcal{T}(A), d \rangle$. The ascending sequence defined in Example 7 shows that this is not necessarily the case when A is infinite. By the way, notice the similarity of (14) and the definition of process L in terms of the sequence T in the proof of Theorem 7.

4.1 Application

In [9, Prop. 3.3 combined with Thm. 3.8] a condition is given under which equation (1) has a unique solution. We now strengthen this result in Theorem 10, yielding an alternative proof for the earlier result.

Let A be an alphabet, p a renaming of A, and R a process in $\mathcal{T}(A \cup p{\cdot}A)$. The function $f: \mathcal{T}(A) \to \mathcal{T}(A)$ is defined by

$$f(S) = (R \,\mathrm{w}\, p{\cdot}S)\lceil A.$$

Theorem 9 The function f is a contraction on $\langle \mathcal{T}(A), d \rangle$ if

$$(\forall r : r \in R : \ell(r\lceil p{\cdot}A) \leq \ell(r\lceil A)). \tag{16}$$

Proof Let T and U be processes in $\mathcal{T}(A)$. We show that f is a contraction by proving $d(f(T), f(U)) \leq \frac{1}{2} d(T, U)$. Without loss of generality we may assume $T \neq U$. On account of this, let k be such that $d(T, U) = 2^{-k}$.

For any $S \in \mathcal{T}(A)$ we have

$$f(S) = \{r : r \in R \wedge r\lceil p{\cdot}A \in p{\cdot}S : r\lceil A\}.$$

Hence, for any trace s of length l in $f(S)$ there exists a trace $r \in R$ with $r\lceil A = s$, $\ell(r\lceil A) = l$, and, due to (16), $\ell(r\lceil p{\cdot}A) \leq l$. If r is of the form $r' p{\cdot}a$, for some $a \in A$, then $r' \in R$ since R is prefix-closed, $r'\lceil p{\cdot}A \in p{\cdot}S$ since S is prefix-closed, and $r'\lceil A = s$. Moreover, $\ell(r'\lceil p{\cdot}A) \leq l - 1$. If on the other hand r ends in a symbol of A then $\ell(r\lceil p{\cdot}A) \leq l - 1$ on account of (16) and the fact that R is prefix-closed. In any case, we conclude for any trace s

$$s \in f(S) \;\equiv\; s = \varepsilon \vee (\exists r : r \in R \wedge r\lceil A = s : r\lceil p{\cdot}A \in p{\cdot}S^{(\ell(s)-1)}) \tag{17}$$

Now we show that $f(T)^{(k)} = f(U)^{(k)}$, which then implies

$$d(f(T), f(U)) \leq 2^{-k-1} = \tfrac{1}{2} d(T, U).$$

For $s \in A^*$ such that $\ell(s) \leq k$ we derive

$$
\begin{array}{ll}
 & s \in f(T) \\
= & \quad \{\ (17)\ \} \\
 & s = \varepsilon \ \vee\ (\exists r : r \in R \wedge r \lceil A = s : r \lceil p{\cdot}A \in p{\cdot}T^{(\ell(s)-1)}) \\
= & \quad \{\ \ell(s) \leq k \text{ and } T^{(k-1)} = U^{(k-1)} \text{ because } d(T,U) = 2^{-k}\ \} \\
 & s = \varepsilon \ \vee\ (\exists r : r \in R \wedge r \lceil A = s : r \lceil p{\cdot}A \in p{\cdot}U^{(\ell(s)-1)}) \\
= & \quad \{\ (17)\ \} \\
 & s \in f(U)
\end{array}
$$

(End of Proof)

Example 8 Condition (16) is not necessary for f to be a contraction. Consider the case $A = \{a\}$ and $R = PR(a \,|\, p{\cdot}a)$. Then R does not satisfy (16), since $p{\cdot}a \in R$. Nevertheless, for any $T \in \mathcal{T}(A)$ we have $f(T) = PR(a)$, and therefore f is a contraction.

(End of Example)

Theorem 10 If the function f is a contraction then it has a unique fixed point and this fixed point U is for every $T \in \mathcal{T}(A)$ the limit of the sequence S defined by

$$S_n = f^n(T).$$

Furthermore, we have for all $n \geq 0$

$$S_n^{(n)} = U^{(n)},$$

that is, the rate of convergence of S is at least "linear".

Proof The first two claims follow from Banach's Contraction Theorem, using the completeness of our metric space. Furthermore, we observe that

$$
\begin{array}{ll}
 & d(S_n, U) \\
= & \quad \{\ \text{definition of } S_n \text{ and } U \text{ is fixed point of } f\ \} \\
 & d(f^n(T), f^n(U)) \\
\leq & \quad \{\ f \text{ contracts by at least a factor } \tfrac{1}{2}\ \} \\
 & 2^{-n} d(T,U) \\
< & \quad \{\ d(T,U) < 1\ \} \\
 & 2^{-n}
\end{array}
$$

and, hence, $S_n^{(n)} = U^{(n)}$ follows from Property 9.
(End of Proof)

By the way, it is possible that f has a unique fixed point but that f is not a contraction for our metric on $\mathcal{T}(A)$. One need only look back to Examples 1 and 2. The functions exhibited there have a unique fixed point, but that fixed point is not the limit of the approximations obtained by iterating the function on $RUN(A)$ (recall the relationship between the metric and the partial order mentioned at the end of the preceding subsection). Hence, these functions cannot be contractions. This does not necessarily mean that they cannot be contractions for some other metric on $\mathcal{T}(A)$.

A slightly different approach to proving Theorem 9 can be taken by extending the notion of a contraction to functions from one metric space into another. Let $\langle \mathcal{D}_0, d_0 \rangle$ and $\langle \mathcal{D}_1, d_1 \rangle$ be metric spaces and let X be a subset of $\mathcal{D}_0$. The function $f: \mathcal{D}_0 \rightarrow \mathcal{D}_1$ is called a c-contraction on X, where c is a real number, when

$$d_1(f(x), f(y)) \leq c * d_0(x, y)$$

for all x and y in X. For instance, renaming is a 1-contraction on the whole space, and so is weaving with a fixed process. If $f_0: \mathcal{D}_0 \rightarrow \mathcal{D}_1$ is a c_0-contraction on X, $X \subseteq \mathcal{D}_0$, and $f_1: \mathcal{D}_1 \rightarrow \mathcal{D}_2$ is a c_1-contraction on $f_0(X)$, then $f_1 \circ f_0$ is a $c_0 * c_1$-contraction on X. The problem in Theorem 9 now is to show that the involved projection is a $\frac{1}{2}$-contraction on the image space created by the composition of renaming and weaving. So, again, projection is the problematic operator.

5 Concluding Remarks

The introduction of recursive components in the program notation of Trace Theory gives rise to a recursive equation involving non-empty prefix-closed trace sets. It becomes, therefore, interesting to know more about the solutions of this equation. For instance, when does the equation have a solution, and if there is more than one solution how can a canonical solution be selected? From the use of a partial order on the process domain (based on trace set inclusion) it is known that the equation has a least and a greatest solution (with respect to this order). Furthermore, the least solution can be approximated by iteration.

In the context of this partial order we have carefully analyzed the projection operator and we have given a number of conditions under which projection distributes over the greatest lower bound of descending sequences of processes (Theorems 0 through 3). Application of this has provided a condition under which the greatest solution of our equation can be approximated by iteration (Theorem 4). It has also been shown that iteration starting anywhere converges to a solution under this condition (Theorem 5).

We have introduced a metric on the process domain and we have shown that it yields a compact complete metric space (Theorems 6 through 8). Under a condition formulated previously the function derived from our equation turned out to be a contraction (Theorem 9). By applying Banach's Contraction Theorem we then showed that under this condition our equation has a unique solution that can be approximated by iteration starting anywhere, and that the rate of convergence is at least linear (Theorem 10).

The conditions that we have formulated are sufficient for their respective purposes but not necessary, as we have shown by appropriate examples. We are still interested in conditions that are sufficient and necessary. That is, exactly when can the greatest solution be approximated, exactly when is the function a contraction, and exactly when has the equation a unique solution?

We have used only one particular partial order and metric in our analysis of the equation. There is no reason to restrict oneself to these choices. It is very well possible that other partial orders and metrics bring new insights. For example, one could think of a metric that not only indicates in which smallest finite approximation two processes differ, but also by how much they differ. A careful analysis of the contracting properties of projection seems a promising area for further investigation.

Acknowledgment This work was supported by the Department of Computer Science, Washington University, St. Louis, Missouri.

References

[0] S. Abian and A.B. Brown, "A theorem on partially ordered sets, with application to fixed point theorems", *Canadian Journal of Mathematics*, Vol. 13, No. 1, pp. 78–82 (1961).

[1] G. Birkhoff, *Lattice Theory*, Colloquium Publications, Vol. 25, Providence RI: American Mathematical Society, 1967 (Third edition).

[2] S.D. Brookes, *A Model for Communicating Sequential Processes*, Report CMU-CS-83-149, Carnegie-Mellon University, 1983.

[3] W.G. Golson and W.C. Rounds, "Connections between two theories of concurrency: metric spaces and synchronization trees", *Information and Control*, Vol. 57, pp. 102–124 (1983).

[4] G.J.O. Jameson, *Topology and Normed Spaces*, New York: Wiley and Sons, 1974.

[5] A. Kaldewaij, *A Formalism for Concurrent Processes*, Dissertation, Eindhoven University of Technology, 1986.

[6] J.-L. Lasses, V.L. Nguyen, and E.A. Sonenberg, "Fixed point theorems and semantics: a folk tale", *Information Processing Letters*, Vol. 14, No. 3, pp. 112–116 (May 1982).

[7] A.W. Roscoe, *A Mathematical Theory of Communicating Processes*, Ph.D. thesis, University of Oxford, 1982.

[8] J.L.A. van de Snepscheut, *Trace Theory and VLSI Design*, Lecture Notes in Computer Science, Vol. 200, New York: Springer-Verlag, 1985.

[9] J.T. Udding, *On Recursively Defined Sets of Traces*, Memorandum JTU0a, Eindhoven University of Technology, 1983.