Seeing beyond divergence

A.W. Roscoe*

June 22, 2004

Abstract

A long-standing complaint about the theory of CSP has been that all theories which encompass divergence are *divergence-strict*, meaning that nothing beyond the first divergence can be seen. In this paper we show that a congruence previously identified as the weakest one to predict divergence over LTS's can be given a new fixed point theory, which we term *reflected fixed points* and thereby turned into a full CSP model which is congruent to the operational semantics.

1 Introduction

This paper was presented at the BCS FACS meeting held in July 2004 in London to commemorate "25 years of CSP"

The author has long (actually 26 years!) worked on mathematical models for concurrent systems, in particular Hoare's CSP [5]. The models he has used – based on observable behaviours – bear an obvious similarity to the congruences studied, for example, by Valmari and his co-workers (for example [13, 14]). The main different has been that those in the CSP "school" have sought complete semantic theories in which the semantics of every term – including recursive ones – could be calculated denotationally, whereas Valmari has concentrated on congruences for sets of operators not including recursion[1]

The models themselves have been broadly similar, particularly after one factors out the differences (more or less irrelevant to this paper) caused by the difference choice operators used: either the CSP $\sqcap$ and $\Box$ (the latter not

*Oxford University Computing Laboratory

[1]The semantic value arises here from applying the observations to the operational (LTS) semantics of the process under examination.

being resolved by a τ) and the CCS $+$ (which is resolved by τ, necessitating knowledge of initial stability). All inhabit the world of finite and infinite traces, divergence traces, and failures/acceptances. The only difference has been that the CSP models have been unable to determine what the non-minimal traces are that a process can diverge on, and what the infinite traces are beyond potential divergence. (The failures/divergences/infinite traces model $\mathcal{U}$ gives all relevant information up to a minimal divergence, and the stable failures/traces model $\mathcal{F}$ gives all information on finite traces and stable failures whether beyond potential divergence or not. See [9] for details of these.)

The main reason for this difficulty is that none of the straightforward ways of finding fixed points give the correct (i.e. operationally congruent) answer. We show here how this problem can be solved using the model of [7] and a more exotic method of calculating fixed points.

For simplicity in this paper we present a model which predicts only finite and infinite traces and divergences – ignoring failures. The latter can be calculated independently through $\mathcal{F}$, or alternatively an extra component can straightforwardly be added to the model we present.

We adopt the full language of CSP from [9], including the interrupt operator $\triangle$ (which has some interesting properties). We note that since a model without failures identifies $\sqcap$, $\Box$ and $+$, this model will also work for languages in the style of CCS. The semantics of CSP over $\mathcal{U}$ and the other standard models can be found in [9], as can the operational semantics of the language. Some of the simpler forms of operational/denotational congruence result are proved in that book, but not the much harder result for $\mathcal{U}$, which is proved in [11]. This paper rests heavily both on this latter result and adaptations of the techniques used in proving it, for example the idea of an approximating sequence of abstraction functions.

The rest of this paper is organised as follows. In the next section we recall two congruences without divergence strictness which would, if a fixed point theory could be developed, solve our problem. However we see that one of these cannot have a conventional denotational fixed-point theory, which leads us to concentrate on a single model. In the following section we see why neither the least nor greatest fixed point gives a sensible semantics for recursion over it. In Section 4 we demonstrate a fixed-point that does apparently give the correct answer, and we show that it is the operationally correct one.

In appendices we give a summary of notation which follows [9], and the new semantics for CSP.

2 Congruences and fixed points

The CFFD congruence (see, for example, [12]) records a process's finite and infinite traces, its divergence traces and its stable failures (pairs (s, X) where X is a set of actions that the process can refuse in some stable, namely τ-free, state after trace s, where neither s nor X includes the invisible event τ). Since we are not seeking to model failures in this paper, it makes sense to simplify CFFD to CFFD$_{\mathcal{T}}$, in which the component of stable failures is removed. This remains a congruence since in CSP and similar languages refusal information does not affect traces or divergences through any operator. (For it to do so would correspond to the operational semantics having a form of negation in the antecedents to some transitions.)

Since CFFD and CFFD$_{\mathcal{T}}$ are attractive congruences (seemingly containing just the information we are looking for) it is natural to ask whether we can find a denotational fixed point theory for them. Unfortunately the answer appears to be no, at least in any conventional sense, if we want that theory to be operationally congruent.

THEOREM 2.1 *There are pairs of CSP recursions whose operational semantics yield different values in CFFD and CFFD$_{\mathcal{T}}$, but which generate identical functions from each of these two models to itself. Therefore there can be no operationally congruent definition of recursion derived from the function a recursion represents.*

PROOF Let $\Sigma = \{a\}$. Consider the process

$$FA = STOP \sqcap (FA \bigtriangleup a \rightarrow (\mathbf{div} \sqcap STOP))$$

(Here, **div** denotes a process that does nothing but diverge.) This may diverge immediately since the nondeterministic choice may always resolve to the right and interrupt may never occur. However it may perform any finite number of a's thanks to layers of interrupts occurring, and plainly may diverge after any of them. For CFFD, which records failures, it can refuse any set after any trace. However, crucially, it cannot perform an infinite trace of as since whenever it performs its first a the number of subsequent ones has some finite bound. (The bound is the number of recursive unfoldings that have occurred up to the point that the first a occurs.)

Notice that FA has every possible trace, divergence and failure except for the infinite trace a^ω. The same value can be created without interrupt by using, for example, infinite nondeterministic choice.[2]

[2]The particular version given here in terms of $\bigtriangleup$ is due to Valmari.

Now consider the two CSP contexts:

$$F_1(P) \quad = \quad FA \sqcap P$$

$$F_2(P) \quad = \quad FA \sqcap a \to P$$

The functions F_1 and F_2 that these contexts generate are identical: the nondeterministic choice of *FA* means that all behaviours other than a^ω belong to both $F_i(P)$ independent of the value of P. In each case it is clear that this infinite trace is present if and only if P has it. It follows that F_1 and F_2 are, extensionally, the same function.

However, operationally, the recursions $P_i = F_i(P_i)$ yield different values. P_1 can perform an arbitrarily large finite number of τs and act like *FA*, or may simply diverge without reaching *FA*. The important thing is that it has no way of performing the trace a^ω. On the other hand P_2 can obviously perform this trace by always picking the right-hand of its two nondeterministic options.

It follows that the extensional value of a function over CFFD or $\mathrm{CFFD}_{\mathcal{T}}$ does not determine the value of the recursion produced by that function. ∎

The above example works by using the *FA* process to shroud the difference between what F_1 and F_2 do to P. The example would not work if we removed *FA*'s ability to diverge after any trace since it is clear that P_1 would in any case diverge on $\langle\rangle$, whereas P_2 would not diverge, and if *FA* could (for example) diverge on the empty trace then P_2 could diverge on all traces.

It turns out that the type of difficulty we have experienced with F_1 and F_2 only occurs in cases like this, where the difference between the two recursions is restricted to infinite traces which belong to $\overline{\mathit{divergences}(P)}$, the closure of the set of divergent traces (namely, the infinite traces that have an infinite chain of divergent prefixes).

The clear lesson to draw from this example is that we either need to add detail to, or remove detail from, our model in order to get a working fixed point model of recursion. One possibility, which co-incides with the abstraction from CFFD made by Puhakka and Valmari for a related reason[3] in [8] (see also [7]), is not to care about whether an infinite trace *which has an infinite chain of divergent prefixes* is present or not: we may choose either to omit all such traces or to include them all. The point of interest here is that the operationally determined values of processes P_1 and P_2 above are

[3] The most abstract congruence capable of detecting all divergence traces

identified by this new equivalence: the trace a^ω is put into the don't-care category by the divergences P_1 and P_2 share.

If, as is typically the case in CSP, we want to have a simple theory of refinement, the correct choice is to include, rather than exclude, all traces that are the limits of chains of divergences. For obviously the process $P = a \to P$ refines $P \sqcap FA$, but this would not appear from the usual reverse containment relation if we were forced to exclude a^ω from the representation of the right-hand process.

We therefore use the model $\mathcal{SBD}$ (standing for seeing beyond divergence) in which each process is represented by a triple (T, I, D), where T and D are the finite traces and divergence traces, and I is the union of the infinite traces and the infinite traces in $\overline{D}$. (This model's elements are obviously in 1-1 correspondence with the members of the $Tr - DivTr - Enditr$ congruence from [8].) The healthiness conditions on this model (in the style usually used for CSP models) are

T1 T is nonempty and prefix-closed.

I1 $s \in \Sigma^* \wedge s\hat{\ }u \in I \Rightarrow s \in T$

D1 $D \subseteq T$

D2 $u \in \Sigma^\omega \wedge (\{s \in \Sigma^* \mid s < u\} \cap D)\text{infinite} \Rightarrow u \in I$

We term the infinite traces covered by D2 *ω-divergent traces*. Following [8] we term the set of these $DivCl(P)$ for process P.

The above model differs from $Tr - DivTr - Enditr$ in that it specifically includes rather than excludes the members of $DivCl(P)$. That does not affect which processes are considered equivalent, but it gives a smoother definition of refinement, as discussed above.

By extension from the result of [8], this is the weakest congruence which predicts all divergences of a CSP process. We will show in the next section that it is possible to find a working, if complex, fixed point theory that calculates the values of recursive processes. The reasons why it *is* a congruence for all CSP operators are, in essence

- If we have two processes that differ only in ω-divergent traces, then the result of applying any CSP operator to them could only differ in either a divergence (which can arise because an infinite trace is hidden in one but not the other) or an infinite trace.

- However if the ω-divergent trace u of P generates, in $C[P]$, the divergence trace s, then certainly there is a finite prefix t of u such that all

$t \leq t' \leq u$ yield the trace s in $C[P]$. However we know that infinitely many of these t' are divergence traces of P, meaning that $C[P]$ gets the divergence from u's divergent prefixes as well as u itself. (For once $C[P]$ has generated the trace s via t, and has carried on generating internal actions through subsequent actions of u, then as soon as P's trace reaches divergent t' then $C[P]$ can diverge without any more actions of u occurring.) Therefore the divergence would still be in $C[P]$ even if u were removed from P.

- If an ω-divergent trace u of P yields the infinite trace v in $C[P]$ then *either* a finite prefix of u also generates v *or* none do. In the former case the presence of v does not depend on u, so u may be removed without affecting it. In the latter an infinite chain of divergent prefixes of u will yield an infinite chain of divergent prefixes of v: this means that v is ω-divergent in $C[P]$ and so in the category of traces whose presence is immaterial thanks to D2.

In any case removing an ω-divergent trace from P will never affect the presence of any behaviour of $C[P]$ that is recorded in $\mathcal{SBD}$.

3 Greatest and least fixed points

It is clear that $\mathcal{SBD}$ provides a less abstract view of CSP processes than the finite/infinite traces and divergences model $\mathcal{I}$ which is strict after divergence, in the sense that the value of a process in $\mathcal{SBD}$ trivially yields the value in $\mathcal{I}$. (More details of $\mathcal{I}$ and the projection Π which performs the translation are given below.)

This immediately tells us quite a lot about the value of any recursive term in $\mathcal{SBD}$: it must be one that projects to the value calculated in $\mathcal{I}$. The existing theory of CSP models tells us how to compute this as a fixed point. For details on this and why the result is operationally correct, see [11, 9]. In order to understand how we might solve the fixed point problem for $\mathcal{SBD}$, it is a good idea to review how other CSP models solve this problem.

The first model for CSP [4] was the finite traces model $\mathcal{T}$ in which each process is represented by its (nonempty and prefix-closed) set of finite traces (sequences of visible events). It is straightforward to give a semantics to each non-recursive CSP operator over $\mathcal{T}$ which is operationally congruent. The fixed-point theory which has always been quoted as standard for that model is based on least fixed points under subset (corresponding to greatest fixed points under refinement). There is a clear operational intuition for this: if we run the recursive process $\mu\, p.F(p)$ then any actual trace will have occurred after a finite time, so that the recursion can only have been unwound a finite number of times. Since $\mu\, p.F(p)$ can only actually do anything once it has been unwound, it follows that (on the assumption that the operational and semantic representations of F are congruent), every trace comes from $F^n(STOP)$ for some n. The reverse is also true for similar reasons.

Given that every CSP operator is continuous under the subset order (see [9], Chapter 8), this informally justifies the use of this fixed point. For a formal proof that it is correct, see [9], Chapter 9.

The second model for CSP was the failures model of [2] which adopted the opposite fixed point strategy: it takes the refinement-least. The argument used then[4] was based on a clear rationale: assume the worst and you will not be disappointed. But of course there was a good pragmatic reason too: there is no least element of that model under subset. A consequence of that decision was that ill-defined recursions such as $\mu\, p.p$ were given lots of traces even though there was no operational reason for doing so. This was the origin of the intuition that a divergent process should be identified with the least refined one: however that model did not in fact stand careful examination when it came to divergent terms. This led to the introduction of the divergences component in [3] leading to the now-standard failures/divergences model $\mathcal{N}$.

Brief consideration of how the mathematics of fixed points works reveals that the refinement-least fixed point is necessary in considering divergence. For example the recursion $\mu\, p.p$ plainly is divergent operationally, if it means anything at all, so within $\mathcal{N}$ we *want* it to denote the least refined process, since that is the only one which is immediately divergent. If we were to throw away the refusal components of failures (giving a finite traces/divergences model) so that we regained a $\subseteq$-least element ($STOP$ again), the $\subseteq$-least fixed point would not be operationally correct. The essential point here is that a divergence is not something that can ever arise in a finite number of iterations of $F(\cdot)$ from $STOP$ except where $F(P)$ may diverge even though P does not. So, with this class of exceptions, $\bigcup\{F^n(STOP) \mid n \in \mathbb{N}\}$ cannot diverge either: this is not true of operational fixed points. Rather each particular divergence should be proved absent in some number of iterations of $F(\cdot)$, which is the essence of the $\sqsubseteq$-least fixed point calculation.[5] Unless

[4] Originally before there was a proper operational semantics for CSP.

[5] With respect to $\sqsubseteq$, not all operators are continuous, though they are monotone (see [9] Chapter 8), meaning that fixed point calculations may need to go to higher ordinals that ω. However this model only turns out to be operationally congruent for finitely nondeterministic CSP, in which operators are continuous. The essence of the proof that

it can be proved absent, it is deemed to be there; and indeed this is accurate since we can again prove congruence with the operational semantics.

An interesting observation that can be made here is that we now know that both $\subseteq$ greatest *and* least fixed points are accurate for computing operationally congruent fixed points for any non-divergent process in (finitely nondeterministic) CSP. It follows that these two are the same, so the fixed point is actually unique. Another way of demonstrating the uniqueness of fixed points for divergence-free processes is via a more rigid divergence-based order as shown in [10]. (The only way $P \leq Q$ is if Q's divergences are a subset of P's and P and Q agree precisely on all traces not in $divergences(P)$: this is called the *strong* order, but only makes sense for divergence-strict models.)

Since $\mathcal{T}$ and $\mathcal{N}$, a number of other CSP models on similar lines have been developed. Notable amongst these are the *stable failures model* $\mathcal{F}$ and infinite traces/failures/divergences model $\mathcal{U}$. The former, representing a process as (F, T) (its sets of stable failures – ones generated in a τ-free state – and finite traces) contains only finitely observable behaviours and, like $\mathcal{T}$, uses subset-least fixed points. The latter represents a process as (F, D, I), its sets of failures, divergences and infinite traces, each closed under divergences so as to make the divergent process bottom under refinement. This has a particularly interesting fixed point theory, since the healthiness conditions relating failures and infinite traces cause the refinement partial order to be incomplete. Nevertheless, as can be shown by various methods [11, 1], refinement-least fixed points do exist for all CSP-definable functions and are operationally congruent. The proof of the congruence between operational and denotational semantics, which is both difficult and crucial to the present paper, may be found in [11]. $\mathcal{U}$ (introduced in that paper) is in essence the minimal extension of $\mathcal{N}$ that can cope accurately with unboundedly nondeterministic operators, and treats divergence in the same way.

As mentioned earlier in relation to CFFD, $\mathcal{U}$ can be simplified to $\mathcal{I}$ in which failures are replaced by finite traces, so a process is represented as (T, D, I). This at least makes the incompleteness problem go away[6], though it does not significantly affect the congruence proof. Once again the model has a refinement top ($STOP$) and since the greatest fixed point of $\mu\, p.a \rightarrow p$ has no infinite trace (it can simply perform every finite trace of a's) we see

that in the finitely nondeterministic case this method predicts the correct divergences is an application of König's Lemma.

[6]Though this is really superficial, since there is still an imperative to demonstrate that the set of infinite traces calculated is consistent with the set of failures calculated via the stable failures model.

that the use of the $\sqsubseteq$-least fixed point is vital not only for divergences (as discussed above) but also for infinite traces: operationally, this process obviously does have an infinite trace, which is correctly predicted by the $\sqsubseteq$-least fixed point.

The conclusion to this survey is that the $\sqsubseteq$-least fixed point is necessary to handle infinite behaviours (namely infinite traces and divergences) properly, but that $\subseteq$ is the correct order to use for finitary ones (namely finite traces and stable failures). The latter is emphasised by the fact that $\mathcal{T}$ and $\mathcal{F}$ are accurate for the full language (including unbounded nondeterminism) and predict the correct behaviours even beyond potential divergence (which is just as well since the models don't know when this occurs). We get away with using $\sqsubseteq$-least fixed points for finite traces and failures prior to divergence simply because these parts of a process are uniquely determined so any fixed point theory would work. And of course in the failures divergences model we do not have to worry about what happens after divergence because everything is mapped to bottom.

In this section we have switched between component-wise subset ($\subseteq$) and refinement ($\sqsubseteq$) – opposites – in discussing fixed points thanks to historical conventions. However from now on, in discussing the calculation of fixed points over $\mathcal{SBD}$, we will use only $\subseteq$.

4 Reflected fixed points

Greatest fixed points do not produce the operationally correct values in $\mathcal{SBD}$: for example the recursion $\mu\, p.p$ is given all finite and infinite traces rather than just the empty trace, which is the right answer. (It is also given all divergences rather than the correct $\{\langle\rangle\}$.) Nor do least fixed points, since the same (divergent) recursion is given the non-divergent value $STOP$, and the process $\mu\, p.a \to p$ is given no infinite trace.

We therefore have to seek a new way of producing a fixed point. In this section we will do this, as part of an overall exercise in demonstrating that the CSP semantics over $\mathcal{SBD}$ including this fixed point theory is congruent to the operational semantics. These two things go hand in hand since the operational justification of the fixed point theory only makes sense in the context of congruence proof, which we now develop.

As with any congruence proof, we need to be careful in distinguishing operational terms and semantic values. The proof itself will be by structural induction over all CSP terms in which all free process identifiers are instantiated by nodes of an arbitrary LTS (perhaps the closed CSP terms themselves), *for all such substitutions simultaneously.*

Since we need to distinguish between different sorts of semantics, we use the notation $\mathbf{SBD}[\![P]\!]\rho$ to mean the denotational semantics of the term P in $\mathcal{SBD}$ where the *environment* $\rho : PV \to \mathcal{SBD}$ gives the binding of all process variables that might appear free in P. Similarly $\mathbf{I}[\![P]\!]\eta$ is the denotational semantics over $\mathcal{I}$.

Specifically, we will aim for the following result.

THEOREM 4.1 *For each CSP term P, and each operational environment $\sigma : PV \to \hat{C}$ (C being an arbitrary LTS and $\hat{C}$ its set of nodes), we have*

$$\Psi(P[\sigma]) = \mathbf{SBD}[\![P]\!](\Psi(\sigma))$$

where Ψ is the natural map from the nodes of any LTS to $\mathcal{SBD}$, $P[\sigma]$ is P with the substitution σ carried out, and $\Psi(\sigma)$ is the function from PV to $\mathcal{SBD}$ (an environment) produced by applying Ψ to $\sigma(v)$ for each variable v[7]*.*

Additionally, the function $\mathbf{SBD}[\![P]\!]$ of environments is monotone. ∎

The second part of this result is necessary to justify some of the constructions and arguments we make below. It is straightforward aside from the case of a recursive term, which requires a more complex inductive argument which we will discuss later. (Note that $\mathbf{SBD}[\![\mu\, p.P]\!]$ is not even defined at present since we don't yet know how to calculate the value of a recursive term.)

To prove this result we need to consider every possible top-level structure for P. These split up into three categories:

- The case where the term P is just a process variable is trivial.

- There are many cases in which P is either a process constant (like $STOP$) or a non-recursive operator. The semantics of all non-recursive operators over $\mathcal{SBD}$ are given in Appendix B. In every case they are identical to the semantics over $\mathcal{I}$ implied[8] in [11] except that the clauses used to enforce divergence strictness are omitted. In each case it is reasonably straightforward to prove that the denotational and operational semantics of the operator are congruent using the same techniques as in [11].

  Of course these results simply parallel the result of [8] that $\mathcal{SBD}$ is a congruence.

[7] This lifting $\Psi(\sigma)$ of Ψ mapping substitutions to environments actually equals the functional composition $\Psi \circ \sigma$.

[8] The semantics in that paper is given over the more complex model $\mathcal{U}$.

- P might be a recursion $\mu\, p.Q$. As in [11], we do not explicitly cover mutual recursion in this proof because of the extra book-keeping, but there is no doubt that the same techniques used below would work for any finite mutual recursion; in infinite mutual recursions similar techniques should work.

  The rest of this section is devoted to developing a fixed-point method which makes the result true for $\mu\, p.Q$ on the assumption that it is true for Q.

Fix a substitution σ, and let $\Theta = \Psi((\mu\, p.Q)[\sigma])$ be the operationally correct value in $\mathcal{SBD}$ relating to the recursion evaluated over σ.

The recursion yields functions $F_{\mathcal{SBD}} : \mathcal{SBD} \to \mathcal{SBD}$ and $F_{\mathcal{U}} : \mathcal{I} \to \mathcal{I}$ defined by

$$\begin{array}{rcl} F_{\mathcal{I}}(X) & = & \mathbf{I}[\![Q]\!]\eta[X/p] \\ F_{\mathcal{SBD}}(X) & = & \mathbf{SBD}[\![Q]\!]\rho[X/p] \end{array}$$

where $\eta = \Phi(\sigma)$ and $\rho = \Psi(\sigma)$ are the environments corresponding to σ over the respective models.

Now let Π be the projection from $\mathcal{SBD}$ to $\mathcal{I}$ obtained by closing up under divergence-strictness . (All extensions of divergences are added to all three components.) And let Φ be the abstraction map (analogous to Ψ) which maps the nodes of any LTS to $\mathcal{I}$ (this is essentially the same as the function of the same name defined in [11]). By construction we have $\Phi = \Pi \circ \Psi$ and since Q is (by our overall inductive assumption in proving Theorem 4.1) operationally accurate over $\mathcal{SBD}$ we get

$$F_{\mathcal{I}}(\Pi(\Psi(X))) = \Phi(Q[\sigma[X/p]]) = \Pi(\Psi(Q[\sigma[X/p]])) = \Pi(F_{\mathcal{SBD}}(\Psi(X)))$$

Since we can without loss of generality assume that our transition system maps *onto* $\mathcal{SBD}$ under Ψ (analogously to the arguments used in [11]) this tells that for each $x \in \mathcal{SBD}$ we have

$$F_{\mathcal{I}}(\Pi(x)) = \Pi(F_{\mathcal{SBD}}(x))$$

or in other words we have a commuting diagram.

Let Ω be the greatest (i.e. $\sqsubseteq$-least) fixed point of $F_{\mathcal{I}}$. Thanks to the known congruence of that semantics with the operational one, we have

$$\Omega = \Phi((\mu\, p.Q)[\sigma]) = \Pi(\Theta)$$

Furthermore, if K is any member of $\Pi^{-1}(\Omega)$ we get

$$\Pi(F_{\mathcal{SBD}}(\mathrm{K})) = F_{\mathcal{I}}(\Pi(\mathrm{K})) = F_{\mathcal{I}}(\Omega) = \Omega$$

In other words $F_{\mathcal{SBD}}$ maps $\Pi^{-1}(\Omega)$ to itself.

If $\Lambda = (T, D, I)$ is any member of $\mathcal{I}$, then $\Pi^{-1}(\Lambda)$ is the interval in $\mathcal{SBD}$ between its $\subseteq$-greatest element, Λ itself, and its least, which is $\tilde{\Lambda} = (T', D', I')$ where

$$\begin{aligned} T' &= T \setminus \{t\hat{\ }s \mid t \in D \wedge s \neq \langle\rangle\} \\ D' &= D \setminus \{t\hat{\ }s \mid t \in D \wedge s \neq \langle\rangle\} \\ I' &= I \setminus \{t\hat{\ }u \mid t \in D \wedge u \in \Sigma^\omega\} \end{aligned}$$

In other words, all behaviour following a potential divergence has been removed. So

$$\Pi^{-1}(\Lambda) = \{K \in \mathcal{SBD} \mid \tilde{\Lambda} \subseteq K \subseteq \Lambda\}$$

The preceding two paragraphs show (together with an easy demonstration of the existence of least upper bounds) that $\Pi^{-1}(\Omega)$ is a complete lattice preserved by the function $F_{\mathcal{SBD}}$. The monotonicity of $F_{\mathcal{SBD}}$ then tells us that it has both greatest and least fixed points within $\Pi^{-1}(\Omega)$.

We know that $\Theta \in \Pi^{-1}(\Omega)$. It is also a fixed point of $F_{\mathcal{SBD}}$ since

$$\begin{aligned} \Theta &= \Psi(\mu\, p.Q[\sigma]) & \\ &= \Psi(Q[\sigma[\mu\, p.Q[\sigma]/p]]) & (1) \\ &= \mathbf{SBD}[\![Q]\!](\Psi(\sigma[\mu\, p.Q[\sigma]/p])) & (2) \\ &= F_{\mathcal{SBD}}(\Psi(\mu\, p.Q[\sigma])) & (3) \\ &= F_{\mathcal{SBD}}(\Theta) & \end{aligned}$$

Here (1) follows by the operational semantics of recursion (which is to execute a τ action and unfold to the term on the right), (2) by our inductive assumption that the denotational and operational semantics of Q are congruent, and (3) by our definition of $F_{\mathcal{SBD}}$.

It follows that Θ lies between the least and greatest fixed points of $F_{\mathcal{SBD}}$ within $\Pi^{-1}(\Omega)$. If these are the same, we have no more to do, but unfortunately this is not always the case, for example in the recursion $\mu\, p.p$.

We will prove, however, that the least fixed point in $\Pi^{-1}(\Omega)$ is always the operationally correct one, and furthermore that it is always given by the standard formula

$$\bigcup_{n=0}^{\infty} F^n_{\mathcal{SBD}}(\tilde{\Omega})$$

(bearing in mind that $\tilde{\Omega}$ is the bottom element of the complete lattice we are concentrating on). As is common, the proof that the above term, which we will name Ξ, equals Θ, comes in two parts.

That $\Xi \subseteq \Theta$ is easy, since $\tilde{\Omega} \subseteq \Theta$ and

$$F^n_{\mathcal{SBD}}(\tilde{\Omega}) \subseteq \Theta \Rightarrow F^{n+1}_{\mathcal{SBD}}(\tilde{\Omega}) \subseteq F_{\mathcal{SBD}}(\Theta) = \Theta$$

giving the result for $\Xi = \bigcup_{n=0}^{\infty} F^n_{\mathcal{SBD}}(\tilde{\Omega})$ by induction.

The intuition behind the argument up to this point can be explained as follows. We know that Ω *is the accurate model of our recursion in* $\mathcal{I}$*. Therefore all the behaviours recorded in* $\tilde{\Omega}$ *must actually be present in* Θ *(for we know that none of them have been inserted by divergence strictness). Since all of these behaviours are present and* Θ *is a fixed point of* $F_{\mathcal{SBD}}$ *it follows that all the behaviours of* $F^n_{\mathcal{SBD}}(\tilde{\Omega})$ *are present for each* n*, and hence those of* Ξ*.*

It is unfortunately more difficult to prove $\Theta \subseteq \Xi$. If this were false, then thanks to the definition of $\mathcal{SBD}$ (especially $D2$), Θ has a behaviour of one of the following sorts not present in Ξ:

- A finite trace.
- A divergence.
- An infinite trace u such that there is some maximal divergence d with $d < u$.

Necessarily the finite trace or divergence s would have to be a *proper* extension of some longest divergence, namely there is a divergence $t < s$ which is maximal subject to this. (Differences cannot appear up to the first divergence since we are operating entirely within $\Pi^{-1}(\Omega)$, a region in which all processes agree up to that point.) We can assume the infinite trace is also the extension of a longest divergence t because if there was no maximal divergence $< u$ then *either* Θ would have a divergence not in Ξ *or* Ξ would have u by axiom $D2$.

This suggests that, in proving $\Theta \subseteq \Xi$, we might try to work by induction based on this maximal divergence. One thought is to count how many divergences there are along a trace, or use the length of trace of the maximal divergence, but these do not work, at least easily, thanks to the complications of hiding. What we actually do is to count how many computation steps (both visible and τ) in the operational semantics of $\mu\, p.F(p)$ are needed to reach this maximal divergence (though not to execute it): clearly this is always a finite number.

We use the technique of an increasing sequence of abstraction functions which approximate Ψ, similar to that used in the main congruence result of [11]. However we only need a countable sequence of them here rather than the arbitrary ordinals in the Φ_α of that paper. The construction we use is exactly that of the previous paper except for the Φ_0 case, in which we map a process to the subset-least member of $\mathcal{SBD}$ consistent with its value in $\mathcal{I}$. (In the previous paper it was mapped to bottom.)

Suppose we are given an LTS L and that $P \in \hat{L}$. Define

$$\begin{array}{rcl}
\Psi_0(P) & = & \widetilde{\Phi(P)} \\
\Psi_{n+1}(P) & = & ?x : P^0 \rightarrow \bigsqcap\{\Psi_n(Q) \mid P \xrightarrow{x} Q\} \\
 & & \qquad \text{if } P \text{ is stable} \\
\Psi_{n+1}(P) & = & ?x : P^0 \rightarrow \bigsqcap\{\Psi_n(Q) \mid P \xrightarrow{x} Q\} \\
 & & \rhd\ \bigsqcap\{\Phi_n(Q) \mid P \xrightarrow{\tau} Q\} \\
 & & \qquad \text{if } P \text{ is not stable}
\end{array}$$

We can view Ψ_n as $\mathcal{G}^n(\Psi_0)$ where $\mathcal{G} : (\hat{L} \rightarrow \mathcal{SBD}) \rightarrow (\hat{L} \rightarrow \mathcal{SBD})$ is implied by the $n+1$ case above. (Syntactically it is precisely the same as the $\mathcal{G}$ operator in [11], though the model is different.) Since Ψ_0 is the $\subseteq$-minimal abstraction consistent with Φ, and we know that (over $\mathcal{I}$) $\mathcal{G}(\Phi) = \Phi$ it follows that $\Psi_1(P) \supseteq \Psi_0(P)$ and hence inductively that the Ψ_i are a $\subseteq$-increasing sequence.

Let $\Psi^*(P) = \bigcup\{\Psi_n(P) \mid n \in \mathbb{N}\}$, where again $\bigcup$ is component-wise union followed by closure under $D2$.

$\mathcal{G}(\Psi) = \Psi$ by construction, as the natural abstractions of the members of an LTS plainly satisfy the defining equations of $\mathcal{G}$. It therefore follows from induction on the n in Ψ_n that $\Psi^*(P) \subseteq \Psi(P)$ for all P. In fact, $\Psi^* = \Psi$, as demonstrated by the following argument.

If b is any behaviour of $\Psi(P) \setminus \Psi^*(P)$, then (as argued for $\Theta \setminus \Xi$ above) by $D2$ we can assume that b is not an infinite trace with an infinite number of divergent prefixes in $\Psi(P)$. Therefore any sequence $\underline{Q}$ of states in P's operational behaviour which witnesses b must have a final state Q with the properties that the trace of b is not complete and which is itself divergent. (There must be some non-final divergence of this form in the sequence of states because $\Pi(\Psi(P)) = \Pi(\Psi^*(P)$ by construction.) Let Q' be the state which appears in $\underline{Q}$ immediately after Q. Let t be the trace up to Q' and let b' be the residue of b, so that $b = t\hat{\ }b'$.

By construction b' is in $\Psi_0(Q')$, and inducting backwards along the sequence of states on our path from P to Q' would easily show that $b \in \Psi_n(P)$ (for n the length of the path). This contradicts our assumption that b is not in $\Psi^*(P)$, and we can conclude that indeed $\Psi = \Psi^*$.

The claim that $\Theta \subseteq \Xi$ will be proved if we can show that, for every n,

$$F^n_{\mathcal{SBD}}(\tilde{\Omega})) \supseteq \Psi_n(\mu\, p.Q[\sigma]) \qquad (\dagger)$$

since in the limit this proves $\Xi \supseteq \Theta$, which is what remained to be proved.

This claim is proved by induction: the $n = 0$ case is trivial since both sides equal $\tilde{\Omega}$.

The $n + 1$ case is a corollary to the following result.

Lemma 4.2 *For all terms Q' and substitutions σ,*

$$\Psi_n(Q'[\sigma]) \subseteq \mathbf{SBD}[\![Q']\!](\Psi_n(\sigma))$$

Note that if $Q' = Q$ (Q as in the term $\mu\, p.Q$ we are addressing in main inductive step) and X is an arbitrary member of the underlying transition system, this result implies

$$\Psi_n(Q[\sigma[X/p]]) \subseteq \mathbf{SBD}[\![Q]\!](\Psi_n(\sigma[X/p]))$$

and the right hand side of this inequality is trivially a subset of $F_{\mathcal{SBD}}(\Psi_n(X))$. (It may be a proper subset since Ψ_n is applied to all components of $\sigma[X/p]$, not just the p one.) ▮

That the step case of the proof of (†) is a corollary follows thanks to the τ which is generated on unfolding $\mu\, p.Q$, which implies:

$$\Psi_{n+1}((\mu\, p.Q)[\sigma]) = \Psi_n(Q[\sigma[(\mu\, .p.Q)[\sigma]/p]]) \subseteq F_{\mathcal{SBD}}(\Psi_n((\mu\, p.Q)[\sigma]))$$

The proof of Lemma 4.2 is by adding it to the structural induction of the main theorem: they are actually proved together. Note that this is the approach used in [11] for a very similar result involving the functions Φ_α.

Before we look at any technical details of the proof it is helpful to realise that this seemingly very technical result actually has a simple intuition. If we accept that Theorem 4.1 holds for our given Q, then what this lemma says is that the behaviours of $\Psi(Q[\sigma])$ for which any non-final divergence occurs before the nth state cannot depend on any behaviour of any $\sigma[q]$

which can only happen beyond a non-final divergence later than n. This can be argued at least semi formally thanks to of two properties of the operational semantics:

1. When any context $C[P]$ runs, the first n steps of its behaviour depend on at most n steps of P. This is because no operational semantic clause ever lets an operand perform an action (visible or invisible) without the overall process doing so as well. (For example, though $P \setminus X$ hides visible actions of P, in the operational semantics these are turned into τ actions, which still count for our purposes.)

   Therefore once $Q[\sigma]$ has performed n operational steps, none of the components $\sigma[\![q]\!]$ can have performed more than n.

2. If the first step of $C[P]$ depends on what actions P has available, and P can immediately diverge, then so can $C[P]$. This is because the operational semantics of every CSP operator ($P \oplus Q$ say) whose immediate actions depend on one of its operands (P say) also has a clause which promotes a τ action of P ($P \xrightarrow{\tau} P'$) to one of $P \oplus Q$, namely $(P \oplus Q) \xrightarrow{\tau} (P' \oplus Q)$.

Let $\underline{Q}$ be any sequence of states that $Q'[\sigma]$ can go through executing a behaviour in which there is no divergent state between step n and any final divergence (namely one reflected in $\Psi_n Q'[\sigma]$. Necessarily, once a component $\sigma[p]$ has been driven (in $\underline{Q}$) to a point where it diverges beyond its own step n (which because of 1. above is beyond step n overall), then either the overall behaviour does not depend on it at all or is within its own divergent tail. For the component of $\underline{Q}$ which contains this state of $\sigma[p]$ is itself divergent by 2.

We can infer that no behaviour in any $\Psi(\sigma[p]) \setminus \Psi_n(\sigma[p])$ is necessary to deduce a behaviour of $\Psi_n(Q'[\sigma])$.

To give a fully formal proof of the Lemma 4.2 part of our main induction requires separate lemmas for each CSP operator very much in the style of those given in [11] for the corresponding result. For recursion it is necessary to perform inductions which follow the derivation of the fixed point: a transfinite one for the fixed point in $\mathcal{I}$ followed by an ordinary one for the fixed point within $\Pi^{-1}(\Omega)$ we are now justifying. We omit these here for brevity.

The overall fixed point calculation is summarised in Figure 1. The left-hand side of the picture shows the entire model $\mathcal{SBD}$, and the small diamonds in it are the regions $\Pi^{-1}(X)$ for each value X in the $\subseteq$-greatest or

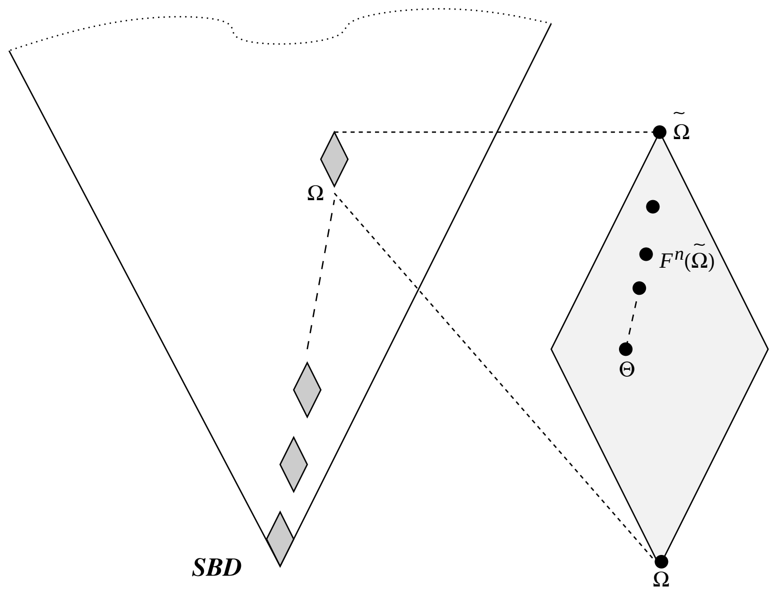

Figure 1: Illustration of the fixed point calculation

$\sqsubseteq$-least fixed point iteration within $\mathcal{I}$, whose ultimate fixed point is Ω. Note that since these diamonds are the preimages of the members of $\mathcal{I}$, they are necessarily disjoint; of course the preimage of a divergence-free member of $\mathcal{I}$ is a singleton set. The right-hand side is an expanded view of $\Pi^{-1}(\Omega)$, showing the iteration from its top towards the operationally correct value Θ. The left-hand side is not a very accurate picture of $\mathcal{I}$, since the latter has a top element. Our argument nowhere uses this however, and would work just as well for an expanded model that replaced finite traces by failures. This has no top (and indeed, like the model $\mathcal{U}$ which it extends is incomplete).

It is mainly because of this potential generalisation that the picture is drawn the way up it is, with refinement from bottom to top (i.e. upside down with respect to $\subseteq$).

The remaining part of Theorem 4.1 is the monotonicity of $\mathbf{SBD}[\![\mu\, p. Q]\!]\eta$ as a function of the environment η. To prove this we show that each part of the construction illustrated in Figure 1 is monotone in η. If $\eta \subseteq \eta'$ then $\Pi(\eta) \subseteq \Pi(\eta')$, and the monotonicity of the semantics over $\mathcal{I}$ implies $\Omega \subseteq \Omega'$. (We use the obvious convention that primed terms are derived from η' in the same way that unprimed ones are derived from η.) This immediately implies $\tilde{\Omega} \subseteq \tilde{\Omega}'$. The fact that $\mathbf{SBD}[\![Q]\!]$ is monotonic by (structural) induction then shows by induction (on n) that

$$F^n_{\mathcal{SBD}}(\tilde{\Omega}) \subseteq (F'_{\mathcal{SBD}})^n(\tilde{\Omega}')$$

for all n.

This concludes our demonstration that Theorem 4.1 and Lemma 4.2 hold.

The shape of the process illustrated in Figure 1 leads us to term the new fixed point method the *reflected fixed point*. Clearly it is highly specialised to this application but we hope it might find uses elsewhere.

Obviously it only produces interesting answers in cases where the value of a recursion is (not necessarily immediately) divergent, since otherwise $\Pi^{-1}(\tilde{\Omega})$ has exactly one point.

The simplest example to check is the recursion $Q_1 = \mu\, p.p$. In this case Ω is the refinement-least element of $\mathcal{I}$ and $\mathcal{SBD}$, and $\tilde{\Omega} = (\{\langle\rangle\}, \{\langle\rangle\}, \emptyset)$, the process which diverges immediately and has no other trace. Since the the function of this recursion is the identity function, it follows that $\tilde{\Omega}$ is the value of the reflected fixed point, which is of course operationally accurate.

As a second example consider

$$Q_2 = ((a \to Q_2;\ b \to \mathit{SKIP}) \sqcap \mathit{SKIP}) \setminus \{a\}$$

Like P_1, this can diverge immediately, and so has the same Ω and $\tilde{\Omega}$ as the first example. This time, however, the first iteration from $\tilde{\Omega}$ brings in an extra trace $\langle\checkmark\rangle$, and each subsequent one brings longer and longer ones of the forms $\langle b\rangle^n$ and $\langle b\rangle^n{}^\frown\langle\checkmark\rangle$, but no further divergence and no infinite trace. Therefore the reflected fixed point has the single divergence $\langle\rangle$, all these finite traces and no infinite trace. This is operationally correct since the behaviour of Q_2 is rather like that of *FA* earlier in the sense that once a b is performed there is already a limit on how many more are allowed.

Finally, consider the recursions for the processes P_1 and P_2 defined using *FA* earlier. Once again we get the same Ω and $\tilde{\Omega}$, but this time every iterate from $\tilde{\Omega}$ in both recursions after the zeroth equals *FA*, and therefore so do both limits. But of course the model axioms tell us that *FA* has the infinite trace a^ω thanks to D2, meaning that within the understanding of $\mathcal{SBD}$ this value is operationally accurate for both.

5 Conclusions

It is fascinating to contemplate the iteration towards a reflected fixed point. The first stage, potentially requiring any ordinal length, manages to characterise accurately all the operational behaviour up to and including the first divergence. Observing what behaviours this proves (under any finite number of iterations) must be present in the fixed point then gives us the correct value in $\mathcal{SBD}$. Nothing can then distinguish the one remaining thing one might want, namely which ω-divergent infinite traces are operationally present.

In a real sense the second stage of our fixed point process is lifting the calculation done in the first to the post-divergence world, and our (forced) decision to ignore ω-divergent traces means that this can happen relatively simply.

Obviously this fixed point process is too complex to use regularly to find the semantics of particular processes. Generally speaking this is best done in any case by abstraction from operational semantics. What it does is show how CSP with its $\mathcal{SBD}$ congruence can be viewed as a self-contained theory that can exist without the corresponding operational semantics, and give considerable understanding of the nature of recursions. It will also allow us to derive mathematical properties of fixed points (such as their monotonicity, and forms of recursion induction).

Other two-stage fixed point techniques have been proposed, such as that of Yifeng Chen in [15]. Others, such as the *optimal fixed point* [6], have been proposed which yield an answer which is in general between greatest and least.

It seems likely that the reflected fixed point will also apply to other fixed point calculations based around potentially diverging finite and infinite sequences. It may also apply in other forms of programming languages semantics where one wishes to liberalise a strict interpretation of divergence or undefinedness.

Appendix 1: Notation

This paper follows the notation of [9], from which most of the following is taken.

$\mathbb{N}$	natural numbers ($\{0, 1, 2, \ldots\}$)
Σ	(Sigma): alphabet of all communications
τ	(tau): the invisible action
Σ^τ	$\Sigma \cup \{\tau\}$
A^*	set of all finite sequences over A
A^ω	set of all infinite sequences over A
$\langle\rangle$	the empty sequence
$\langle a_1, \ldots, a_n\rangle$	the sequence containing $a_1, \ldots, a_n$ in that order
a^ω	the infinite trace $\langle a, a, a, \ldots\rangle$
$s\hat{\ }t$	concatenation of two sequences
$s \setminus X$	hiding: all members of X deleted from s
$s \underset{X}{\parallel} t$	the set of traces composed from subsequences s and t which share members of X and are disjoint elsewhere.
$s \leq t$	($\equiv \exists\, u.s\hat{\ }u = t$) prefix order
$\overline{S}$	closure of S ($= S \cup \{u \in \Sigma^\omega \mid \{s \in S \mid s < u\}$ is infinite$\}$)

Processes:

$\mu\, p.P$	recursion
$a \rightarrow P$	prefixing
$?x : A \rightarrow P$	prefix choice
$P \,\Box\, Q$	external choice
$P \sqcap Q, \quad \bigsqcap S$	nondeterministic choice
$P \underset{X}{\parallel} Q$	generalised parallel
$P \setminus X$	hiding
$P[\![R]\!]$	renaming (relational)
$P \rhd Q$	'time-out' operator (sliding choice)
$P \bigtriangleup Q$	interrupt
$P[x/y]$	substitution (for a free identifier x)

Transition Systems:

$\hat{C}$	The set of nodes in transition system C.
$P \xrightarrow{a} Q$	($a \in \Sigma \cup \{\tau\}$) single action transition
$P \overset{s}{\Longrightarrow} Q$	($s \in \Sigma^*$) multiple action transition with τ's removed
$P \overset{t}{\longmapsto} Q$	($t \in (\Sigma^\tau))^*$) multiple action transition with τ's retained
P *ref* B	P refuses B
P *div*	P diverges

Models:

$\mathcal{T}$	traces model
$\mathcal{N}$	failures/divergences model (divergence strict)
$\mathcal{F}$	stable failures model
$\mathcal{I}$	finite and infinite traces/divergences model with divergence strictness
$\mathcal{U}$	failures/divergences/infinite traces model with divergence strictness
CFFD	failures/divergences/infinite traces congruence/model without divergence strictness
$\mathrm{CFFD}_{\mathcal{T}}$	finite and infinite traces/divergences congruence/model without divergence strictness
$\mathcal{SBD}$	finite and infinite traces/divergences model strict under ω-divergent infinite traces
$\mathcal{SBD}$	stable failures/divergences/infinite traces model strict under ω-divergent infinite traces
$\bot_{\mathcal{N}}$ (etc.)	bottom elements of models
$\top_{\mathcal{F}}$ (etc.)	top elements of models
$\sqsubseteq$	refinement over whatever model is clear from the context
$P \leq Q$	strong order (over divergence-strict models)
$\mathbf{I}[\![P]\!]\eta$	denotational semantics of P in $\mathcal{I}$
$\mathbf{SBD}[\![P]\!]\rho$	denotational semantics of P in $\mathcal{SBD}$

Appendix B: CSP semantics over the new model

The semantics of recursion has been discussed extensively in the main body of the paper. What remains to be done, therefore, is to provide a recipe for calculating the semantic result of applying any one of the non-recursive operators to the right number of members of $\mathcal{SBD}$. As usual we factor this into recipes for the three components separately.

In general discussions of operators we will refer to a typical binary one $P \oplus Q$. However there is nothing specific to binary operators there and appropriate modifications of the statements hold for all.

In each case the recipe for *traces*$(P \oplus Q)$ is precisely the same for the traces model $\mathcal{T}$, and depends only on the traces component of the arguments to the relevant operator. This is, of course, not surprising, but it is pleasing since it has not been true of any previous CSP model supporting divergence.

$$
\begin{aligned}
traces(STOP) &= \emptyset \\
traces(SKIP) &= \emptyset \\
traces(a \to P) &= \{\langle a\rangle\hat{\ }s \mid s \in traces(P)\} \\
traces(?x : A \to P) &= \{\langle a\rangle\hat{\ }s \mid a \in A \wedge s \in traces(P[a/x])\} \\
traces(P \sqcap Q) &= traces(P) \cup traces(Q) \\
traces(P \underset{X}{\parallel} Q) &= \bigcup\{s \underset{X}{\parallel} t \mid s \in traces(P) \wedge t \in traces(Q)\} \\
traces(P;\ Q) &= traces(P) \cup \\
&\quad \{s\hat{\ }t \mid s\hat{\ }\langle\checkmark\rangle \in traces(P) \wedge t \in traces(Q)\} \\
traces(P[\![R]\!]) &= \{s' \mid \exists\, s \in traces(P) \mid s\, R\, s'\} \\
traces(P \setminus X) &= \{s \setminus X \mid s \in traces(P)\} \\
traces(P \bigtriangleup Q) &= \{s\hat{\ }t \mid s \in traces(P) \wedge t \in traces(Q)\}
\end{aligned}
$$

The recipes for divergences involve, in different cases, all three components. They are the same as previous CSP models, but without the closure constructions used to enforce divergence strictness.

$$
\begin{aligned}
divergences(STOP) &= \emptyset \\
divergences(SKIP) &= \emptyset \\
divergences(a \to P) &= \{\langle a\rangle\hat{\ }s \mid s \in divergences(P)\} \\
divergences(?x : A \to P) &= \{\langle a\rangle\hat{\ }s \mid a \in A \wedge s \in divergences(P[a/x])\} \\
divergences(P \sqcap Q) &= divergences(P) \cup divergences(Q) \\
divergences(P \underset{X}{\parallel} Q) &= \bigcup\{s \underset{X}{\parallel} t \mid s \in divergences(P) \wedge t \in traces(Q)\} \\
&\quad \cup \bigcup\{s \underset{X}{\parallel} t \mid s \in traces(P) \wedge t \in divergences(Q)\} \\
divergences(P;\ Q) &= divergences(P) \cup \\
&\quad \{s\hat{\ }t \mid s\hat{\ }\langle\checkmark\rangle \in traces(P) \wedge t \in divergences(Q)\} \\
divergences(P[\![R]\!]) &= \{s' \mid \exists\, s \in divergences(P) \mid s\, R\, s'\} \\
divergences(P \setminus X) &= \{u \setminus X \mid u \in infinites(P) \wedge u \setminus X \text{ is finite}\} \\
&\quad \cup \{s \setminus X \mid s \in divergences(P) \cap \Sigma^* \wedge t \in \Sigma^{*\checkmark}\} \\
divergences(P \bigtriangleup Q) &= divergences(P) \cup \\
&\quad \{s\hat{\ }t \mid s \in traces(P) \wedge t \in divergences(Q)\}
\end{aligned}
$$

In each case the basic recipe for $infinites(P \oplus Q)$ depends only on the sets $Traces(P)$ and $Traces(Q)$ of all finite and infinite traces of the arguments. However in some cases a clause adding the ω-divergent infinite traces is need to make $D2$ true.

$$
\begin{array}{rcl}
\mathit{infinites}(STOP) & = & \emptyset \\
\mathit{infinites}(SKIP) & = & \emptyset \\
\mathit{infinites}(a \rightarrow P) & = & \{\langle a\rangle\hat{\ }u \mid u \in \mathit{infinites}(P)\} \\
\mathit{infinites}(?x : A \rightarrow P) & = & \{\langle a\rangle\hat{\ }u \mid a \in A \wedge u \in \mathit{infinites}(P[a/x])\} \\
\mathit{infinites}(P \,\Box\, Q) & = & \mathit{infinites}(P) \cup \mathit{infinites}(Q) \\
\mathit{infinites}(P \sqcap Q) & = & \mathit{infinites}(P) \cup \mathit{infinites}(Q) \\
\mathit{infinites}(P \underset{X}{\parallel} Q) & = & \{u \in \Sigma^\omega \mid \exists\, s \in \mathit{Traces}(P), \\
 & & \qquad t \in \mathit{Traces}(Q).u \in s \underset{X}{\parallel} t\} \\
 & & \cup\, (\Sigma^\omega \cap \overline{\mathit{divergences}(P \underset{X}{\parallel} Q)}) \\
\mathit{infinites}(P;\ Q) & = & \mathit{infinites}(P) \\
 & & \cup\, \{s\hat{\ }u \mid s\hat{\ }\langle\checkmark\rangle \in \mathit{traces}(P) \wedge u \in \mathit{infinites}(Q)\} \\
 & & \cup\, (\Sigma^\omega \cap \overline{\mathit{divergences}(P;\ Q)}) \\
\mathit{infinites}(P[\![R]\!]) & = & \{u' \mid \exists\, u \in \mathit{infinites}(P).u\, R\, u'\} \\
 & & \cup\, (\Sigma^\omega \cap \overline{\mathit{divergences}(P[\![R]\!])}) \\
\mathit{infinites}(P \setminus X) & = & \{u' \in \Sigma^\omega \mid \exists\, u \in \mathit{infinites}(P).u \setminus X = u'\} \\
 & & \cup\, (\Sigma^\omega \cap \overline{\mathit{divergences}(P \setminus X)}) \\
\mathit{infinites}(P \bigtriangleup Q) & = & \mathit{infinites}(P) \cup \{s\hat{\ }u \mid s \in \mathit{traces}(P) \wedge u \in \mathit{infinites}(Q)\} \\
 & & \cup\, (\Sigma^\omega \cap \overline{\mathit{divergences}(P \bigtriangleup Q)})
\end{array}
$$

Acknowledgements

This work has benefitted enormously from a number of discussions with Antti Valmari. He, together with others such as Michael Goldsmith, have made helpful suggestions after reading earlier versions of this paper.

References

[1] G. Barrett, *The fixed-point theory of unbounded nondeterminism*, Formal Aspects of Computing, **3**, 110–128, 1991.

[2] S.D. Brookes, C.A.R. Hoare and A.W. Roscoe, *A theory of communicating sequential processes*, Journal of the ACM **31**, 3, 560–599, 1984.

[3] S.D. Brookes and A.W. Roscoe, *An improved failures model for CSP*, Proceedings of the Pittsburgh seminar on concurrency, Springer LNCS 197, 1985.

[4] C.A.R. Hoare, *A model for communicating sequential processes*, in 'On the construction of programs' (McKeag and MacNaughten, *eds*), Cambridge University Press, 1980.

[5] C.A.R. Hoare, *Communicating sequential processes*, Prentice Hall, 1985.

[6] Zohar Manna and A. Shamir. *The theoretical aspects of the optimal fixed-point*, SIAM Journal of Computing, 5:414 - 426, 1976.

[7] A. Puhakka, *Weakest Congruence Results Concerning "Any-Lock"*, Proc. TACS'2001, Fourth International Symposium on Theoretical Aspects of Computer Software, October 29-31 2001, Sendai, Japan, Lecture Notes in Computer Science 2215, Springer-Verlag 2001, pp. 400-419.

[8] A. Puhakka, A and A. Valmari, *Weakest-Congruence Results for Livelock-Preserving Equivalences*, Proceedings of CONCUR '99 (Concurrency Theory), Lecture Notes in Computer Science 1664, Springer-Verlag 1999, pp. 510-524.

[9] A.W. Roscoe *The theory and practice of concurrency*, Prentice-Hall International, 1998.

[10] A.W. Roscoe, *An alternative order for the failures model*, in 'Two papers on CSP', technical monograph PRG-67, Oxford University Computing Laboratory, July 1988. Also Journal of Logic and Computation **2**, 5, 557–577, 1992.

[11] A.W. Roscoe, *Unbounded nondeterminism in CSP*, in 'Two papers on CSP', technical monograph PRG-67, Oxford University Computing Laboratory, July 1988. Also Journal of Logic and Computation, **3**, 2 131–172, 1993.

[12] A. Valmari, *A Chaos-Free Failures Divergences Semantics with Applications to Verification*, Millennial Perspectives in Computer Science: Proceedings of the 1999 Oxford–Microsoft Symposium in honour of Sir Tony Hoare, Palgrave "Cornerstones of Computing" series, 2000, pp. 365-382.

[13] A. Valmari, *The weakest deadlock-preserving congruence*, Information Processing Letters **53**, 341–346, 1995.

[14] A. Valmari and M. Tienari *An improved failures equivalence for finite-state systems with a reduction algorithm*, Protocol Specification, Testing and Verification XI, North-Holland, 1991.

[15] Yifeng Chen, *A fixpoint theory for non-monotonic parallelism,* Theoretical Computer Science, Vol. 308 No 1-3, pp.367-392, 2003.