The three Platonic models of divergence-strict CSP

A.W. Roscoe

Oxford University Computing Laboratory
{Bill.Roscoe@comlab.ox.ac.uk}

Abstract. In an earlier paper [13], the author proved that there were three models of CSP that play a special role amongst the ones based on finite observations: the traces ($\mathcal{T}$), stable failures ($\mathcal{F}$) and stable revivals ($\mathcal{R}$) models are successively more refined, but all further models refine $\mathcal{R}$. In the present paper we prove the corresponding result for the divergence-strict models: ones that treat any process that can diverge immediately as the least in the refinement order. We define what it is to be a divergence-strict model, both for general and finitely nondeterministic CSP, and find that in order to get our result we need to add a new but natural operator into the language.

1 Introduction

The process algebra CSP [5, 10] is traditionally studied via behavioural models, namely combinations of sets of linear observations that might be made of them. The reference point for making these observations is CSP's standard LTS-based operational semantics as set out in Chapter 7 of [10] (perhaps with definitions for operators not considered there). In order to be a *model*, a representation has to be a congruence (it must be possible to deduce the observations of the result of applying any CSP operator from the observations of its arguments) and there must be a way of working out the operationally correct value of any recursive term from the function the recursion represents over the model.

One can divide the models of CSP into three categories:

- The models such as finite traces $\mathcal{T}$ and stable failures $\mathcal{F}$ (representing a process as its sets of finite traces, and stable failures (s, X) where the process can, after trace s, reach a state in which neither an internal τ action nor a member of X is possible). All the observations made of processes in this class of models are *finite*: they can be completed in a finite time.
- The divergence-strict models such as failures/divergences $\mathcal{N}$ in which, in addition to some finite observations, we are allowed to record some behaviours that take infinitely long. At any time when we are recording an observation, we will record the process diverging if it does, and furthermore we choose not to care about what our process might do on any observation that extends such a divergence. (A process diverges when it performs an infinite unbroken sequence of τ actions.)
  This class is particularly important since $\mathcal{N}$ and its extension to include infinite traces are the simplest that allow (respectively for finitely nondeterministic and general processes) one to specify that a process, offered a set X of events, *must* accept one of them. They also give the expressive power to define the concept of *determinism* [10].
  Divergence strictness is useful for two reasons: firstly it permits (as we shall see later) the modelling of finitely branching processes without a separate representation of infinite traces, and secondly because it enormously simplifies the semantics of recursion. If one always regards divergence as an error, there is no need for models that distinguish beyond it.

– Models that record infinite behaviour but are not subject to divergence strictness. The first examples of these that are not just Cartesian products of the other two types were demonstrated in [12].

In [13], the author introduced a new family of models based on observing slightly more than in *failures*-based models and less than in either *ready-set* (sometimes termed *acceptance-set*) models or *refusal testing* models. In this family, we choose to observe not only failures (s, X) (where our process can perform the trace s and then stably refuse X) but also, when X is not the set of all visible events, single actions a that the state witnessing the refusal of X can perform. Thus processes include in their representations their *deadlock* traces (representing the traces on which the process can refuse anything) and their *revivals* (s, X, a). This family of models was inspired by the conformance equivalence of Fournet *et al* [1].

We discovered there that the stable failures model plays a special role in the van Glabbeek hierarchy [2, 3], since it was shown to complete a fundamental initial sequence. Specifically, we showed that, with respect to the CSP language used in that paper, any non-trivial finite observation model that is not one of the increasingly refined sequence $\mathcal{T}$, $\mathcal{N}$ and $\mathcal{R}$ must refine $\mathcal{R}$. Furthermore there is no model $\mathcal{R} \prec \mathcal{M}$ such that every CSP model strictly refining $\mathcal{R}$ must refine $\mathcal{M}$. (For one congruence $\mathcal{B}$ to refine another $\mathcal{A}$ means that any pair of processes identified by $\mathcal{B}$ are identified by $\mathcal{A}$. We will sometimes write this $\mathcal{A} \preceq \mathcal{B}$.) These three initial models are seemingly forced on us and can be compared (perhaps fancifully) to the Platonic solids of geometry.

In that paper the author conjectured that essentially the same result would hold in the class of divergence-strict models. The purpose of the present paper is to resolve that conjecture: in fact it is true, but not quite in the terms that the author envisaged, since the language needs to be extended in a subtle way. The main structural result of this paper is the following theorem.

Theorem 1. *For a suitably extended (see Section 5) language CSP+, the three congruences $\mathcal{T}^{\Downarrow\omega}$, $\mathcal{F}^{\Downarrow\omega}$ and $\mathcal{R}^{\Downarrow\omega}$ (i.e. the finite observation models extended by strict divergence traces and infinite traces) are more abstract than every other nontrivial model of this language.*

The first thing we need to do, even to fully understand this statement, is decide what qualifies as a divergence-strict CSP model. We establish this via the creation of the most refined such model of them all. We also establish relationships between models for the full CSP language and ones for finitely nondeterministic CSP – which we abbreviate **f**CSP and whose standard models are denoted $\mathcal{T}^{\Downarrow}$ etc – that allow us to restrict our attention for the rest of the paper to the latter. This is a considerable bonus since we only have to consider behaviours over which we can use induction, and find we can restrict attention to finite restrictions of processes.

Having done this we are able to prove the first stage of the main result, namely that the divergence-strict traces model $\mathcal{T}^{\Downarrow}$ is refined by *every* nontrivial such CSP congruence, with respect to the same language used in [13].

It came as a surprise to the author that, with this same dialect of CSP, there is a curious congruence that is not quite as refined as the failures-divergences model $\mathcal{N}$. The discovery of this congruence (previously noted in [9]) leads to the observation that no CSP operator treats processes in a specific way that seems operationally natural, namely for some action of a given process P leading directly to the operator turning P off. We therefore add such an operator to the language, obtaining a language we term CSP+. We show that there is

a surprisingly stark contrast between the relative roles of our new operator – $P \Theta_a Q$ that allows P to throw control to Q by communicating a – and the more usual interrupt operator $P \triangle Q$.

With this enhanced language we are able to complete the proof of the main result in two steps, one to prove that $\mathcal{N}$ is the weakest proper refinement of $\mathcal{T}^{\Downarrow}$, and the second to prove that the divergence-revivals model $\mathcal{R}^{\Downarrow}$ is the weakest proper refinement of this.

There is an appendix of notation. We do not give detailed descriptions in this paper of well-established CSP models or the semantics of CSP over them. The interested reader can easily find these in [10] or [13].

Acknowledgements

This is one of a series of papers that was inspired by the work of Jakob Rehof, Sriram Rajamani and others in deriving *conformance*, a revivals-like congruence for a CCS-like language. My work on this paper benefited greatly from conversations with Jakob, Antti Valmari and Tony Hoare.

2 Background

In this paper, as in [13], we restrict ourselves to the study of models where the overall alphabet Σ is finite. However we only consider potential models that make sense for any size of Σ and have the property that a pair of processes defined over Σ_1 are equivalent over a model defined over Σ_1 if and only if they are equivalent over the same model defined over every larger Σ_2, because the model over Σ_1 is a natural restriction of the larger one. This means, for example, that we can establish properties of an equivalence between processes defined over Σ_1 by introducing a finite number of extra events and studying the equivalence over the resulting larger Σ_2. We might also note the following:

- The CSP renaming operator – with its ability to apply an arbitrary permutation to a process's alphabet – implies that any congruence for CSP must be essentially symmetric in events.
- Combinations of prefixing, renaming, parallel and hiding allow CSP to bring differences between processes forward or to postpone them. This suggests that CSP congruences must discriminate behaviours happening at any point in a process's execution uniformly.

Certainly all established models obey these two principles.

2.1 The CSP language

Our starting point for CSP in this paper is the same language as in [13], but without the two (*SKIP* and ;) related to successful termination. This latter omission is just to make our arguments simpler[1] – the results of this paper are still valid with them added. The constant processes are thus

- *STOP* which does nothing – a representation of deadlock.
- **div** which performs (only) an infinite sequence of internal τ actions – a representation of divergence or live-lock.

[1] The chief benefit is that we do not have to allow for processes terminating in the many contexts we create for them in this paper. The reader can see this effect in [13].

– *CHAOS* which can do anything except diverge.
– $RUN(A)$ which always offers the actions A.

and the operators

– $a \rightarrow P$ communicates the event $a \in \Sigma$ before behaving like P. This is *prefixing*.
– $?x : A \rightarrow P(x)$ communicates any event from $A \subseteq \Sigma$ and then behaves like the appropriate $P(x)$. This is *prefix choice*.
– $P \sqcap Q$ lets the process decide to behave like P or like Q: this is *nondeterministic* or *internal* choice. It can be used as a binary operator like this or over nonempty sets of processes $\bigsqcap S$. The only difference between CSP and **f**CSP is that in the latter we may not use $\bigsqcap$ over infinite sets.
– $P \Box Q$ offers the environment the choice between the initial Σ-events of P and Q. If the one selected is unambiguous then it continues to behave like the one chosen; if it is an initial event of both then the subsequent behaviour is nondeterministic. The occurrence of τ in one of P and Q does *not* resolve the choice (unlike CCS +), and if one of P and Q can terminate then so can $P \Box Q$. This is *external* choice.
– $P \rhd Q$ may choose to offer the visible actions of P but, unless one of these is followed, *must* offer the initial choices of Q. This is *asymmetric* or *sliding* choice and can be said to give an abstract (and untimed) representation of P timing out, if none of its initial actions are accepted, and becoming Q. This is considered primitive for reasons set out in [13].
– $P \underset{X}{\parallel} Q$ runs P and Q in parallel, allowing each of them to perform any action in $\Sigma - X$ independently, whereas actions in X must be synchronised between the two. It terminates when both P and Q have, a rule which is equivalent to stating that $\checkmark$ is synchronised like members of X. All other CSP parallel operators can be defined in terms of this one.
– $P \setminus X$, for $X \subseteq \Sigma$, *hides* X by turning all P's X-actions into τs.
– $P[\![R]\!]$ applies the *renaming* relation $R \subseteq \Sigma \times \Sigma$ to P: if $(a, b) \in R$ and P can perform a, then $P[\![R]\!]$ can perform b.
– $P \triangle Q$ runs like P but if at any time the environment communicates an initial visible action of Q, then (nondeterministically if that event is also currently offered by P) P shuts down and the process continues like Q. This is the *interrupt* operator.

We will discover some interesting things about $\triangle$ in Section 5.

The final CSP construct is recursion: this can be single or mutual (including mutual recursions over infinite parameter spaces), can be defined by systems of equations or (in the case of single recursion) in line via the notation $\mu\, p.P$, for a term P that may include the free process identifier p.

2.2 The hierarchy of CSP models

CSP models traditionally represent processes by sets of observations which can be made of a process. These observations are always ones that it is reasonable for someone interacting with the process to see in some finite or infinite linear interaction with it (in other words things are seen in some definite succession, with no branching). We work here under the same postulates as in [13], namely that the things that our observer can see are:

(a) Visible actions from Σ.

(b) The fact that a process is stable (is unable to perform any further actions without the co-operation of its environment), *and then*
   (i) whether it refuses a set of actions $X \subseteq \Sigma$ it is offered and
   (ii) the actual set of actions from Σ it is offering.

We note here that the ability to observe refusal sets is implied by the ability to observe acceptance (sometimes called *ready set*) information.

We specifically exclude the possibility that our observer might see that some action happens when the process is *unstable*. It is hard to justify that one could observe that some τ action was possible without actually following it, and such observations would imply some highly undesirable inequalities between processes.

This means that the most refined model for CSP based on finite observations is $\mathcal{FL}$, in which behaviours of the form

$$\langle A_0, a_0, A_1, a_2, \ldots, A_{n-1}, a_{n-1}, A_n \rangle$$

are recorded, with the a_i visible events and the A_i *generalised acceptances*, being either $\bullet$, meaning that stability was not observed at this point, or the acceptance set of the stable state that occurred at the relevant point. In this second case we expect $a_i \in A_i$. The set of all such sequences will be termed *FLO* (finite linear observations). We will denote them by Greek letters $\beta, \gamma, \ldots$, which will also sometimes denote the same sort of alternating sequence beginning or ending in an event rather than a generalised acceptance, and even infinite sequences of these forms.

The healthiness conditions are that (the representation of) a process P must satisfy

FL0 P is nonempty: specifically $\langle \bullet \rangle \in P$
FL1 P is prefix closed: if $\beta^\frown\gamma \in P$ and β ends in a generalised acceptance, then $\beta \in P$.
FL2 P is closed under observing less stability: if $\beta^\frown\langle A \rangle^\frown\gamma \in P$, then so is $\beta^\frown\langle \bullet \rangle^\frown\gamma$.
FL3 All proper acceptances can be realised: if $\beta^\frown\langle A \rangle \in P$ and $A \neq \bullet$, then $\beta^\frown\langle A, a, \bullet \rangle \in P$ for all $a \in A$.

It is straightforward to construct semantic clauses for all operators in our language over this model.

In [13], the author defined a finite-observation CSP model to be any model that represents a process as a finite tuple of relational images of its image in $\mathcal{FL}$. The number of such relations needs to be independent of the size of the alphabet Σ, and the equivalences induced over processes over Σ must be independent of which $\Sigma' \supseteq \Sigma$ is used to construct the model. We can also expect, thanks to the observations at the start of Section 2, that all the relations will be symmetric under permutations of Σ. All the standard models fit comfortably into this definition. We will find in this paper, however, that we can generalise it a little. The finite observation models other than $\mathcal{FL}$ that were studied in [13] were

$\mathcal{T}$ the finite trace model [4],
$\mathcal{F}$ the stable failures model [10], which records a process's finite traces and stable failures,
$\mathcal{R}$ the stable revivals model, which records a process's finite traces, deadlock traces and stable revivals as described above,
$\mathcal{A}$ the stable acceptances model (based on [7]), which records finite traces and pairs (s, A) in which A is a stable acceptance set at the end of the trace s, and
$\mathcal{RT}$ the stable refusal testing model (based on [8, 6]), in which behaviours have the same appearance as for $\mathcal{FL}$, but where (subset closed) refusal sets replace acceptances.

Each of the above models can be extended to a divergence-strict one in two ways: one that handles only **f**CSP and an extension which handles the whole language. For a given finite-observation model $\mathcal{M}$, these two divergence-strict analogues are written $\mathcal{M}^{\Downarrow}$ and $\mathcal{M}^{\Downarrow\omega}$ respectively. These notations are explained thus:

- A divergence-strict model's role is much more about telling us when a process must stabilise if left alone, rather than when it diverges. After all, the basic assumption of the model is that once a process can diverge we don't care what else it does. In other words, every trace that is not a divergence is one on which the process definitely *converges* or becomes stable. $P \Downarrow$ often means "P is convergent" in the literature.
- A^ω is a common notation for infinite sequences of members of A, and it is necessary to include infinite sequences of actions etc explicitly in models to deal with the combination of divergence and the CSP hiding operator.

$\mathcal{M}^{\Downarrow}$ simply adds a component of "divergences" to $\mathcal{M}$. A *divergence* is generally anything that $\mathcal{M}$ allows us to record during an incomplete behaviour after which the observed process might diverge (perform an infinite unbroken series of τs). Thus, for $\mathcal{T}^{\Downarrow}$, $\mathcal{F}^{\Downarrow}$ $(= \mathcal{N})$, $\mathcal{R}^{\Downarrow}$ and $\mathcal{A}^{\Downarrow}$, a divergence is a trace, for $\mathcal{RT}^{\Downarrow}$ it is a refusal trace ending in $\bullet$ and for $\mathcal{FL}^{\Downarrow}$ it is an acceptance trace ending in $\bullet$. It turns out that the addition of the divergences component to $\mathcal{F}$, $\mathcal{R}$ and $\mathcal{A}$ allows the removal of the finite traces component: after any finite trace a process must either diverge or become stable.

In each case the model is made *divergence strict* by including a healthiness condition that says that if β is any divergence recorded in the model $\mathcal{M}$, then every extension of β (whether a divergence or another type of behaviour) is automatically included in a process P's representation *whether the operational P can actually be observed performing this extension or not.*

A process's representation in $\mathcal{FL}^{\Downarrow}$ therefore takes the form of a pair (B, D) of subsets of FLO, with every member of D ending in $\bullet$: B represents those that can be observed of the process, and D represents the ones on which it can diverge. Both, of course, are extended by extensions of divergences. The healthiness conditions FL0–FL3 still apply, as do:

FLD1 $\beta^\frown\langle\bullet\rangle \in D$ imples $\beta^\frown\gamma \in B$ for all suitably-formed γ.
FLD2 $\beta^\frown\langle\bullet\rangle \in D$ imples $\beta^\frown\gamma'^\frown\langle\bullet\rangle \in D$ for all suitably-formed γ'.
FLD3 $\beta^\frown\langle\bullet\rangle \in B - D$ implies that there is $A \neq \bullet$ such that $\beta^\frown\langle A\rangle \in B$.

The first two of these impose divergence strictness, and the last says that after any observation a process either eventually becomes stable or diverges.

We can represent (B, D) either explicitly like this or as a single set in which the two forms of behaviour are both present, only with the final compulsory $\bullet$ of each divergence replaced by $\Uparrow$. These two are clearly equivalent, and we will move between them as convenient.

The following property of $\mathcal{FL}^{\Downarrow}$ makes the close relationship between it and the CSP language clear, and also clarifies the meaning of some of our later arguments.

Theorem 2. *Every member of $\mathcal{FL}^{\Downarrow}$ is the semantics of a CSP process.*

PROOF The author proved a number of similar results for other models in [13]. The construction we use here is similar to that used for other divergence-strict models there.

Before we start we will observe the following: if $(B, D) \in \mathcal{FL}^{\Downarrow}$ and $\beta \in B$, then we can define a process $(B, D)/\beta$ – the behaviour *after* β by the following, where $\beta = \beta'^\frown\langle A\rangle$ for

some A.

$$\textbf{div} \qquad \text{if } \beta'^\frown\langle\bullet\rangle \in D, \quad \text{and otherwise}$$

$$(\{\gamma \mid \beta'^\frown\gamma \in B\}, \{\gamma \mid \beta'^\frown\gamma \in D\}) \qquad \text{if } A = \bullet$$

$$(\{\langle A'\rangle^\frown\gamma \mid \beta^\frown\gamma \in B \wedge A' \in \{\bullet, A\}\}, \{\langle A'\rangle^\frown\gamma \mid \beta^\frown\gamma \in D \wedge A' \in \{\bullet, A\}\}) \text{ if } A \neq \bullet$$

We can now define a process $INT(B, D)$ that represents a formal interpreter for an arbitrary member of $\mathcal{FL}^{\Downarrow}$. If $\langle\bullet\rangle \in D$ (i.e. the process can diverge immediately) then $IND(B, D) = \textbf{div}$. Otherwise, by FLD3 we know that the set $ACCS = \{A \neq \bullet \mid \langle A\rangle \in B\}$ is nonempty, so we can define $INT(B, D)$ to be as follows, where $B^0 = \{a \mid \langle\bullet, a, \bullet\rangle \in B\}$.

$$?x : B^0 \to INT((B, D)/\langle\bullet, a, \bullet\rangle)$$

$$\rhd \sqcap\{?x : A \to INT((B, D)/\langle A, a, \bullet\rangle) \mid A \in ACCS\}$$

Note that this can perform every action that the target (B, D) can initially, unstably. Also for every stable acceptance A that the target has initially, our interpreter can offer A and then carry on in any way that (B, D) can after observatiing $\langle A\rangle$.. This completes the proof of Theorem 2. ∎

As discussed in [10, 12], in models that involve strict divergence it works far better (for example in finding the fixed points of recursions) to approximate processes from below in the refinement order, or even the "strong order" described in [11] in which the only way to move up (at least amongst models that do not model infinite behaviours *other* than divergences) is to convert some divergent behaviour into non-divergent.

In a related fashion, all of the known finite-nondeterminism models of CSP are naturally turned into (ultra) metric spaces by considering the *restriction* $P \downarrow n$ of any process to $n \in \mathbb{N}$ to be all behaviours of P up to and including the nth events in its traces, with the $P \downarrow n$ becoming divergent after these nth events. So $P \downarrow 0$ is equivalent to the immediately divergent process **div**. The distance between a pair of processes P and Q is

$$d(P, Q) = inf\{2^{-n} \mid P \downarrow n = Q \downarrow n\}$$

Noting that a process's image in $\mathcal{FL}^{\Downarrow}$ is *already* a divergence-strict construction, we can expect that it will usually not be necessary to re-inforce this once more. We can therefore specify that a *natural* divergence-strict model $\mathcal{M}$ for **f**CSP is formed from a finite number of components, the observations of each of which are either a relational image of B or of D, where the process's value in $\mathcal{FL}^{\Downarrow}$ is (B, D). We can describe these two collections of images as NB and ND. These must satisfy:

(i) The induced equivalence is a congruence, with $\sqsubseteq$ (i.e. reverse containment) giving a congruent least-fixed-point semantics for recursion.
(ii) The images of of B and D are separate components of the image.
(iii) If $P \neq_M Q$ then there exists $n \in \mathbb{N}$ such that $P \downarrow n \neq_M Q \downarrow n$

We view (ii) as a clarity assumption: since $D \subset B$ it avoids ambiguity over how to create members of NB. (iii) holds automatically provided (as in all known models) the behaviours of $\mathcal{M}$ partition into lengths that correspond (even to within a constant factor) to the lengths of their pre-images in $\mathcal{FL}^{\Downarrow}$.

The above definition can be generalised in the following way that, as we will find later, allows the concept of divergence strictness to be interpreted more liberally. In other words it will allow a model to be *more* divergence strict than a simple image of $\mathcal{FL}^{\Downarrow}$ would allow.

A *general* divergence-strict model identifies each process P with $f(B, D)$, where (B, D) is its image in $\mathcal{FL}^{\Downarrow}$ and f is a $\subseteq$-continuous function from $\mathcal{FL}^{\Downarrow}$ to a partial order $\mathcal{O}$. Here, by $\subseteq$-*continuous*, we mean that if C is any linearly ordered set of processes over $\mathcal{FL}^{\Downarrow}$, then $f(\bigcup C) = \bigsqcap(\{f(P) \mid P \in C\}$, where this greatest lower bound exists in the range of f. The choice of $\bigsqcap$ rather than $\bigsqcup$ here is a convention – it says that we associate the direction of the order on $\mathcal{O}$ with the refinement order on processes, and indeed will think of it as refinement.

This last continuity property is always true of the relational image definition by construction, and it implies that f is monotone. The resulting model $\mathcal{M}$ (a subset of $\mathcal{O}$) is $\{f(P) \mid P \in \mathcal{FL}^{\Downarrow}\}$: it must be a congruence for **f**CSP with $\sqsubseteq$-least fixed points giving the congruent denotation for recursions, and satisfy $P \neq_M Q \Rightarrow \exists\, n.f(P \downarrow n) \neq f(Q \downarrow n)$.

Note that, as a result of Theorem 2 and our definition above, every member of every general divergence-strict model is expressible in CSP.

We can similarly generalise the definition of finite observation models, again using the $\subseteq$-continuity property. The proofs in [13] still work, with little alteration. The author has yet to find a good reason for *wanting* this generalisation from plain relational images over finite-observation models, however. If we are to model a process as one or more classes of individual finitely and linearly observable things, it is hard to see why these should need to be inferred from sets of members of FLO as opposed to individual ones.

3 Finitary versus general models of CSP

The metric described above works because all the behaviours in $\mathcal{FL}^{\Downarrow}$ have a finite length: the best definition for this is the number of visible actions in the corresponding trace if it ends in divergence, and this number plus one otherwise. The range of behaviours we allow our notional observer to see in constructing $\mathcal{FL}^{\Downarrow}$ do not cover all possibilities, since they do not include the records of interactions that take infinitely long and include an infinite number of visible actions rather than ending in permanent stable refusal or divergence. To create a full record in the spirit of $\mathcal{FL}^{\Downarrow}$ we could also record ones taking the form of sequences $\langle A_0, a_0, A_1, a_1, A_2, \ldots\rangle$ that have the same structure as the $\mathcal{FL}$ behaviours FLO except that they go on for ever.

There is an important reason for this omission: all **f**CSP processes, like the finitely branching LTS's that are their operational semantics, have a natural closure property. Their infinite behaviour can be deduced from the behaviours we record in models like $\mathcal{T}^{\Downarrow}$ and $\mathcal{FL}^{\Downarrow}$ that only explicitly record finite traces and similar; a summary proof of this follows below (for $FL^{\Downarrow}$).

Suppose γ is an infinite behaviour of the above form, all of whose prefixes belong to some node P of a finitely branching LTS. Consider the tree formed by unrolling the behaviour of P, with all parts not reachable in a prefix of γ pruned away. By assumption, since γ has arbitrarily long prefixes, this tree is infinite; it is also finitely branching by assumption. König's Lemma tells us there is an infinite path through it. We consider two possibilities: either the actions of that path contain an infinite sequence of consecutive τs or they do not. If they do then there is a prefix of γ that is divergent in P. γ is then a member of P's $\mathcal{FL}^{\Downarrow\omega}$ by closure under divergence strictness. If they do not then the nodes in this sequence are easily seen to be witnesses of the full behaviour γ. This completes our proof.

This is not true if we extend our interest to general CSP, and we therefore take the obvious step of adding such infinite behaviours into the representation of a process in the

extended model $\mathcal{FL}^{\Downarrow\omega}$. Each process becomes a triple (B, D, I) with I consisting of these infinite behaviours.

The next natural question to ask is when such a triple is the representation of a reasonable process – or, in other words, how to formulate natural *healthiness conditions*. Fortunately we have a well-established way of determining this via the principle that

(*) every CSP process is equivalent to the nondeterministic choice of all its finitely nondeterministic refinements, or equivalently its set of *closed refinements*.

Here, a process (B, D, I) is *closed* if and only if I consists precisely of those infinite behaviours all of whose finite prefixes belong to B. Refinement, as ever, is defined by superset.

To understand this condition, note first that any process of the form of such a nondeterministic composition is, by Theorem 2 and one use of nondeterministic choice, expressible in CSP. To prove that every process's representation can be expressed thus, consider any behaviour γ of the node P of an arbitrary LTS. As above, we can unroll P's behaviour into a tree T (where no node is reachable in more than one way, or from itself through a non-empty path). Identify an infinite path through the tree that either witnesses γ or some divergent prefix. Now systematically prune the tree subject to two constraints:

- In the resulting tree T' no node has more than one outward action with any particular label from $\Sigma \cup \{\tau\}$, but always has exactly the same set of initial actions as the corresponding node in T.
- All nodes of T no longer reachable from its root are discarded.
- Every node and action on the path identified above is preserved.

The behaviour of the root state of T' is a process that (a) has the behaviour γ, (b) refines the original process P, and (c) is finitely branching and therefore has a closed image in our model. This means that every behaviour of P is one of a closed refinement of P, justifying our assertion that every process is just the sum of the behaviours of its closed refinements.

Infinite behaviours make no contribution to the calculation of the restrictions $P \downarrow n$ over $\mathcal{FL}^{\Downarrow\omega}$, although these processes do have infinite behaviours thanks to divergence strictness. Closed processes are precisely those such that $P = \bigsqcup\{P \downarrow n \mid n \in \mathbb{N}\}$.

We can now define a divergence-strict natural model of full CSP to be a finite tuple of relational images of a process's image in $\mathcal{FL}^{\Downarrow\omega}$ satisfying the following:

(i) It provides a congruence.
(ii) The images of the three components (B, D, I) are disjoint, and the images of the components (B, D) provide a natural divergence-strict model for **f**CSP that gives the same congruence as $\mathcal{M}$ itself over these processes and which satisfies our definition of such models above

The rationale behind (ii) is much the same as in the earlier definition: it ensures that the infinite behaviours of $\mathcal{FL}^{\Downarrow\omega}$ are not used to reveal details that could equally have been deduced from the finite behaviour components.

The properties of relational imaging guarantee that every such model $\mathcal{M}$ satisfies property (*), so that with respect to the particular infinite details that have been recorded, the congruence on finitely nondeterministic CSP determines that on the full language.

Given that every model of full CSP is a model of **f**CSP, and the strong results we will prove later showing that there are no interesting general, as opposed to natural, models

for an extended **f**CSP that interfere with our structural result, we choose not to attempt a generalisation of the concept of a "general model" involving infinite behaviours.

It is highly relevant to the subject matter of this paper to ask whether any any finitary model $\mathcal{F}$ can have more than one extension to the full language, through the use of different sets of infinite behaviours. By this, of course, we mean sets of infinite behaviours that give rise to different equivalences over CSP. The main determining factor in this is the semantics of hiding.

We can show that every divergence-strict model of full CSP must distinguish processes based on their infinite traces:

Lemma 1. *Suppose that $\mathcal{M}$ is a divergence-strict congruence for full CSP. Then two processes that have different infinite traces as judged in $\mathcal{T}^{\Downarrow\omega}$ must be mapped to different processes in $\mathcal{M}$. Furthermore, each such natural model for full CSP has a distinct relational image or images for each infinite trace u.*

PROOF Suppose P and Q are processes with different sets of infinite traces but are identified by $\mathcal{M}$. We can assume that P has an infinite trace u that Q lacks. (And for any given u we could easily create a specific P and Q for this u.) We can create a special process XI_u that has every possible behaviour that does not imply the presence of u:

$$XI_u = \sqcap\{ V \mid V \text{ is a closed process without the trace } u\}$$

We can also create a process that performs u but only in unstable states US_u:

$$US_{\langle a\rangle^\frown u} = (a \rightarrow US_u) \rhd STOP$$

Let $PT = (P \underset{\Sigma}{\parallel} US_u) \sqcap XI_u$ and $QT = (Q \underset{\Sigma}{\parallel} US_u) \sqcap XI_u$. These two processes are equivalent in all their finitely observable and deadlock behaviour, and cannot perform the infinite trace u except that PT can do so from unstable states all the way along the trace if it cannot diverge on a prefix of u.

Let T_u be the process that simply steps in turn through the events of u (each offered stably). Consider the context $C[X] = (X \underset{\Sigma}{\parallel} T_u) \setminus \Sigma$. Operationally, it is clear that $C[PT]$ can diverge immediately, but $C[QT]$ cannot: in fact the latter process is equivalent to $STOP$.

Since $R \,\Box\, \mathbf{div} = \mathbf{div}$ and $R \,\Box\, STOP = R$ for all CSP processes R in all CSP models, it follows that our model $\mathcal{M}$ must distinguish **div** and $STOP$. Therefore (from the action of $C[\cdot]$ and the fact that $\mathcal{M}$ is a congruence), it must also distinguish PT and QT; and P and Q in turn. However the only recordable behaviour on which PT and QT differ is the everywhere unstable infinite trace u. It follows that the relations that create $\mathcal{M}$ from $\mathcal{FL}^{\Downarrow\omega}$ must map this behaviour to an image that is distinct from those of all other behaviours other than ones that also contain the same infinite trace.

We can therefore conclude that $\mathcal{M}$ must contain enough information to deduce what all the infinite traces of a process are. This concludes the proof of Lemma 1. ∎

Now suppose that γ_0 is the infinite $\mathcal{FL}^{\Downarrow\omega}$ behaviour representing the observation of the whole of u performed unstably (i.e. the events of the trace u with $\bullet$s between), and that γ_1 is any other behaviour in which u is performed: necessarily γ_1 has some first observation of stability (via a particular acceptance set) in it. We can write γ_1 as $\langle \bullet, a_1, \bullet, \ldots, a_{r-1}, A_r, a_r\rangle^\frown \xi$.

There are processes that contain γ_0 in their $\mathcal{FL}^{\Downarrow\omega}$ representation but not γ_1: an example is the process US_u as defined above.

The following technical lemma is what will allow us to achieve the main result of this section, namely proving that, as far as the main structural result of this paper is concerned, we can restrict our attention to models of **f**CSP.

Lemma 2. *Suppose that γ_0 and γ_1 are as specified above. Then we can find a pair of finitely nondeterministic, closed and divergence-free processes P and Q that are equivalent up to the acceptance set model $\mathcal{A}^{\Downarrow\omega}$, where P has the behaviour γ_1, and Q has γ_0 but not γ_1.*

PROOF We can straightforwardly define a process that has any infinite $\mathcal{FL}^{\Downarrow\omega}$ behaviour η as follows:

$$II(\langle \bullet, a\rangle^\frown \eta) = a \rightarrow II(\eta) \rhd STOP$$
$$II(\langle A, a\rangle^\frown \eta) = STOP \sqcap (a \rightarrow II(\eta) \;\Box ?x : A - \{a\} \rightarrow STOP$$

In the second case necessarily $a \in A$. These behaviours have been designed so that when $\eta_1 \leq \eta_2$ (i.e. η_1 is obtained from η_2 by replacing some A_i with $\bullet$), we have $II(\eta_2) \sqsubseteq II(\eta_1)$.

The following process does not have η unless all the "acceptances" are $\bullet$, but it does have the associated infinite trace and all the trace/acceptance pairs (s, A) that the presence of η implies.

$$FSI(\langle \bullet, a\rangle^\frown \eta) = a \rightarrow FSI(\eta) \rhd STOP$$
$$FSI(\langle A, a\rangle^\frown \eta) = a \rightarrow FSI(\eta) \rhd (STOP \sqcap (?x : A \rightarrow STOP))$$

$FSI(\eta)$ is equivalent, in the finite acceptances model $\mathcal{A}$, to $II(\eta)$:

- Clearly they have the same finite traces: the finite prefixes of η's trace extended by any event a that belongs to an acceptance of η in the appropriate place.
- They have the same infinite traces, namely $\{u\}$.
- Both can deadlock after any trace.
- Both can offer any proper acceptance offered by η at the appropriate point in the trace.

Since these processes are both divergence free and finitely nondeterministic, this equivalence extends to $\mathcal{A}^{\Downarrow\omega}$.

The lemma is therefore established by setting $P = II(\gamma_1)$ and $Q = FSI(\gamma_1)$. ∎

We are now in a position to prove a strong result about the infinite extensions of a class of models that includes all those that are central to our main structural result.

Theorem 3. *Each of the models $\mathcal{T}^{\Downarrow}$, $\mathcal{F}^{\Downarrow} = \mathcal{N}$, $\mathcal{R}^{\Downarrow}$ and $\mathcal{A}^{\Downarrow}$ has, judging by the equivalence represented on processes and transition systems, a* unique *extension to become a natural infinitary model.*

PROOF We know [13] that each of them can be so extended by the addition of the component of infinite traces. By Lemma 1 we know that any such extension contains a distinct relational image for each infinite trace. If any infinite behaviour γ_1 had a relational image distinct from the corresponding infinite trace, then the finitary processes P and Q created by Lemma 2 would be distinguished by our hypothetical extension, even though they are equivalent in $\mathcal{A}^{\Downarrow}$ and hence in each of $\mathcal{T}^{\Downarrow}$, $\mathcal{F}^{\Downarrow}$ and $\mathcal{R}^{\Downarrow}$. This would contradict the fact that an extension must yield the same equivalence on finitary terms as the model being extended. ∎

It follows from this, and Lemma 1, that if our main structural result holds for finitary models, then it also holds for general models: any non-trivial model of full CSP must refine $\mathcal{T}^{\Downarrow\omega}$, and so on.

We note in passing that Theorem 3 does not extend to models of finitary CSP that are richer than those listed. Specifically it does not seem to hold for models where an arbitrarily long series of refusals and/or acceptances are recorded. It turns out, for example, that $\mathcal{FL}^{\Downarrow}$ has at least three different extensions: we can choose to record

- As many as infinitely many acceptance sets in a trace, as in $\mathcal{FL}^{\Downarrow\omega}$.
- An arbitrarily large finite number of acceptance sets in a trace, so that any infinite behaviour has an infinite tail of $\bullet$s.
- An arbitrarily long finite string of acceptance sets or $\bullet$, followed by an infinite string of refusal sets or $\bullet$.

4 Stage 1: every model refines $\mathcal{T}^{\Downarrow}$

What we now seek to prove is that every nontrivial general model of **f**CSP satisfies one of the following:

- It represents the same equivalence as $\mathcal{T}^{\Downarrow}$.
- It represents the same equivalence as $\mathcal{F}^{\Downarrow} = \mathcal{N}$.
- It refines $\mathcal{R}^{\Downarrow}$.

We break our analysis of this into three stages:

1. Showing that every such model refines $\mathcal{T}^{\Downarrow}$.
2. Showing that every such model that is not $\mathcal{T}^{\Downarrow}$ refines $\mathcal{F}^{\Downarrow}$.
3. Showing that every such model that is not $\mathcal{T}^{\Downarrow}$ or $\mathcal{F}^{\Downarrow}$ refines $\mathcal{R}^{\Downarrow}$.

In [13], the author used two different patterns of proof for the corresponding results. In each case he was proving that every congruence $\mathcal{M}$ for CSP that strictly refines some congruence $\mathcal{A}$, must also refine some second congruence $\mathcal{B}$. (Stage 1 has this form if we allow $\mathcal{A}$ to be the trivial congruence that identifies all processes.) In both styles of proof we can start out by assuming that there are a pair of processes P and Q such that $P \neq_M Q$ but $P =_A Q$. From this it is easily deduced, by considering $P \sqcap Q$ (which cannot be $\mathcal{M}$-equivalent to both P and Q), that without loss of generality we can assume $P \sqsubset_M Q$.

In the first pattern of proof we assume we have a pair of processes such that $V \not\sqsubseteq_B U$ and $U =_A V$, and construct a context such that $C[U] = P$ and $C[V] = Q$ (equality holding in all models of the class being considered, so certainly $\mathcal{M}$). Since P and Q are being mapped to those processes that are distinct in $\mathcal{M}$, it follows that P and Q are themselves distinct in $\mathcal{M}$, which is what we wanted to prove.

The second pattern, which we will see in Sections 6 and 7, operates on similar principles but depends on showing by technical analysis that we can choose very special P and Q that make a more difficult construction of $C[\cdot]$ possible.

In [13], the first style of proof was used for the first two steps of the overall result, namely proving that $\mathcal{T}$ is the unique minimally refined finite-observation model and that $\mathcal{F}$ is uniquely minimal amongst the rest of this class of models.

In the case of divergence-strict models, the author has only found a way of doing this for the first stage, though this is remarkably straightforward. Indeed it follows from an argument essentially the same as our proof of Lemma 1 above.

Theorem 4. *If $\mathcal{M}$ is a non-trivial divergence-strict model for CSP, then $\mathcal{T}^{\Downarrow} \preceq \mathcal{M}$.*

PROOF We will follow the first pattern above. However the fact that $\mathcal{M}$ is divergence strict allows us to be specific about P: we can clearly set it equal to **div** and choose Q to be any process that is not $\mathcal{M}$-equivalent to **div**.

If U and V are processes that are distinguished by $\mathcal{T}^{\Downarrow}$, then without loss of generality we can assume that $V \not\sqsubseteq_{TD} U$. In other words *either* U has a divergence trace s not in V, or the divergence sets are equal and U has a trace t not in V.

In the first case let $D[X] = (T(s) \parallel X) \setminus \Sigma$ where $T(\langle\rangle) = STOP$ and $T(\langle a\rangle^\frown s) = a \rightarrow T(s)$. It is easy to see that in general $D[X] = STOP$ unless X has s as a divergent trace, in which case $D[X] = \mathbf{div}$, this equality holding in $\mathcal{FL}^{\Downarrow}$ and hence in all divergence-strict models.

In the second let $D[X] = (T^{\Uparrow}(t) \underset{\Sigma}{\parallel} X) \setminus \Sigma$ where $T^{\Uparrow}(\langle\rangle) = \mathbf{div}$ and $T^{\Uparrow}(\langle a\rangle^\frown t) = a \rightarrow T^{\Uparrow}(t)$. Again, it is easy to see that $D[X] = STOP$ if X does not have the trace t, and $D[X] = \mathbf{div}$ if it does.

In either case let $C[X] = D[X] \,\Box\, Q$. $C[U]$ can diverge immediately because $D[U]$ can. As all immediately divergent processes are equivalent to **div** in divergence-strict models, this tells us that $C[U] = \mathbf{div}$ in all of them. On the other hand $C[V] = STOP \,\Box\, Q$, and since $STOP \,\Box\, Q = Q$ in all known CSP models including $\mathcal{FL}^{\Downarrow}$, we have $C[V] = Q$.

Thus the CSP context $C[\cdot]$ maps U and V to two processes that are distinct in $\mathcal{M}$. We can deduce from the fact that $\mathcal{M}$ is a congruence that U and V must themselves be distinct in $\mathcal{M}$. This completes the proof of Theorem 4. ∎

5 An unexpected congruence and how to avoid it

After establishing Theorem 4, the author moved on to try to prove that every model for **f**CSP that properly refines $\mathcal{T}^{\Downarrow}$ in turn refines $\mathcal{F}^{\Downarrow} = \mathcal{N}$. These efforts failed, and he was disappointed to discover that there is a model that lies strictly between these two models.

The language as defined in Section 2.1 has a modified – and slightly more abstract – version of $\mathcal{N}$ as a model. This has all the usual healthiness conditions plus one more:

$$(s, X) \in F \wedge s^\frown\langle a\rangle \in D \Rightarrow (s, X \cup \{a\}) \in F$$

The interpretation of this is that we choose not to care about whether events that lead immediately to divergence are refused or not. The resulting extended refusals are included in the model rather than excluded so as to make the theory of refinement as set containment work: this decision is analogous to the one to include rather than exclude all post-divergence behaviours.

For example this model identifies $a \rightarrow \mathbf{div}$ with $STOP \sqcap a \rightarrow \mathbf{div}$, which are distinct in $\mathcal{N}$.

It is not a *natural model* in the sense described earlier for two reasons. Firstly, the extra refusals each depend on an arbitrary number of divergences. Secondly there is more cross play between the divergence and non-divergence behaviours than is allowed in the definition of a natural model. What it creates is a strangely amplified notion of divergence strictness: to create this we need to use the machinery set out in the definition of a general model.

The immediate question that comes to mind when seeing this model, which we will call $\mathcal{N}^-$, is "How can this be a congruence?" To answer this we need to look to the operational semantics of CSP (viewable online at [10], Chapter 7). All of the operators in the usual language, and hence the whole language, satisfy the following principle:

– Suppose the context $C[P]$ can perform the initial action a and become the process Q, and P itself performs some action $P \xrightarrow{b} P'$ that is part of a (i.e. the operational rules that generate a depend on P performing b). Then the term Q always involves P' and is divergence strict in it – in other words, if P' can perform an infinite sequence of τs then so can Q.

Putting it another way, no CSP operator ever allows an argument to perform an action and then immediately disposes of that argument. (A number of operators including $\Box$ dispose of *other* arguments when one performs a visible action.)

What this means is that if the argument P of $C[P]$ performing b leads immediately to divergence, then so does the derived action a of $C[P]$. Clearly we would expect the issue of whether P can refuse b or not to affect whether $C[P]$ can refuse a – but what we have discovered is that:

– The refusal by P of an action that leads immediately to divergence can only affect the refusal by $C[P]$ of actions that lead immediately to divergence.

Another way of reading this is that discarding the information about whether P can refuse b or not can only mean that we are unable to discover information about whether $C[P]$ can refuse other actions that lead directly to divergence. It should therefore not come as too much of a surprise to discover that throwing away all such information from all processes (which is what $\mathcal{N}^-$ does) yields a congruence for **f**CSP.

This congruence had previously been identified for a sub-language in [9], but the author was not aware of it until the failure of the proof of the natural step 2 of the structural theorem forced him to rediscover it.

There seems no good reason at all why there is no operator in CSP that throws away a process as soon as it has performed an action. (Actually, in a sense, the sequential composition operator ; does, but the assumptions and restrictions conventionally placed on the termination signal ✓ mean that this exception is not decisive.) Implementing such an operator would not cause any particular problem. Evidently there is just no such concept in concurrency that Hoare thought was necessary to include in CSP. In fact, given the importance of operating system ideas in Hoare's initial work (see, for example [5], Chapters 5–7) including exception handling (Chapter 5), the author expected to find that Hoare had discussed an operator that allowed a process to throw an exception and pass on control to a second process or some external context. In fact there is no such operator, since all the exceptions that Hoare's operators handle are triggered by external events rather than internally generated ones. Correspondence between the author and Hoare ensued, during which we were unable to discover any such operator in previous work but agreed that it would be perfectly natural to add one.

In particular the following *exception throwing* operator seems very natural:

$P \,\Theta_a\, Q$ behaves like P until P communicates a, at which point it starts Q:

$$\frac{P \xrightarrow{x} P'}{P \,\Theta_a\, Q \xrightarrow{x} P' \,\Theta_a\, Q}(x \neq a) \qquad \frac{P \xrightarrow{a} P'}{P \,\Theta_a\, Q \xrightarrow{a} Q}$$

We will add it to the language: we will call the result CSP+.

$\mathcal{N}^-$ is not a model for CSP+ because it is not a congruence: recall that $a \rightarrow \mathbf{div}$ and $STOP \sqcap a \rightarrow \mathbf{div}$ are identified by $\mathcal{N}^-$. On the other hand $(a \rightarrow \mathbf{div}) \,\Theta_a\, STOP = a \rightarrow STOP$ and $(STOP \sqcap a \rightarrow \mathbf{div}) \,\Theta_a\, STOP = STOP \sqcap a \rightarrow STOP$, and these two processes are *not* equivalent over $\mathcal{N}^-$, which they would have to be if it was a congruence.

In a subsequent paper, the author will demonstrate that in an important sense Θ_a can be said to *complete* the CSP language, since it means that every operator which is expressible in a natural class of operational semantics can be expressed in CSP+. For the time being, however, we will examine its relationship with the CSP language described in [13].

In fact, it has a very interesting relationship with $\triangle$. Recall that it was necessary to include $\triangle$ in the CSP language in [13] to obtain the structural result for finite-observation models. This very fact means that $\triangle$ cannot be expressed in terms of the rest of the language in a general finite-observation model. It therefore comes as something of a surprise to discover the following result.

Lemma 3. *In $\mathcal{FL}^{\Downarrow\omega}$, and therefore in every divergence-strict model, $\triangle$ can be expressed using the other operators of CSP.*

PROOF The easiest way to prove this lemma is to give the equivalent expression: extend the alphabet from Σ_0 to $\Sigma = \Sigma_0 \cup \Sigma_1$ where $\Sigma_1 = \{a' \mid a \in \Sigma_0\}$ (the map from a to $'$ being injective and $\Sigma_0 \cap \Sigma_1 = \emptyset$). The relations *Prime* and *Unprime* respectively map every member a of Σ_0 to a', and every $a' \in \Sigma_1$ to a, leaving other events unchanged.

$$P \triangle' Q = (P \,|||\, Q[\![Prime]\!]) \underset{\Sigma}{\parallel} Reg)[\![Unprime]\!], \quad \text{where}$$
$$Reg = (?x : \Sigma_0 \to Reg) \,\Box\, (?x : \Sigma_1 \to RUN(\Sigma_1))$$

What this construct does is to allow P to proceed until Q communicates an event, at which point P is blocked (by *Reg*) from performing any further actions. This is, of course, very nearly the desired effect of the interrupt operator $\triangle$. The only difference is that after Q has performed a visible action, P can still perform internal actions in the above construct whereas in $P \triangle Q$ it is actually turned off. This can make the difference between a process being stable or unstable: $\mathbf{div} \triangle' Q$ can never be stable – and therefore have stable acceptances – and $\mathbf{div} \triangle Q$ can. These two versions are different in any finite observation model that is richer than traces. The difference with divergence-strict models, however, is that for the two versions to be semantically different in finite observation models, P has to be in a state where it can diverge at the point where it is interrupted. It follows that the interruption must be of a potentially divergent state of $P \triangle Q$ also. Thus the differences only appear beyond the point where $P \triangle Q$ can diverge, and so they are eliminated by divergence strictness, which obliterates such distinctions.

So in fact, over divergence-strict models, $\triangle$ and $\triangle'$ are equivalent. This completes the proof of Lemma 3. ∎

$P \,\Theta_a\, Q$ can be defined correctly over all standard CSP models. For example, over $\mathcal{FL}^{\Downarrow}$ we can define:

$$\begin{aligned} P \,\Theta_a\, Q = \{&\beta \in P \mid trace(\beta) \in (\Sigma - \{a\})^*\} \\ &\cup \{\beta^\frown\gamma \mid \beta^\frown\langle\bullet\rangle \in P, \gamma \in Q, trace(\beta) \in (\Sigma - \{a\})^*\{a\}\} \\ &\cup \{\beta^\frown\gamma \mid \beta^\frown\langle\Uparrow\rangle \in P, trace(\beta) \in (\Sigma - \{a\})^*\} \end{aligned}$$

Here, we are using the representation of processes as single sets containing both ordinary and divergent behaviours, and $trace(\beta)$ is the sequence of visible events in β. The third line is needed to achieve divergence strictness.

It is possible, in general, to define $\triangle$ in terms of Θ_a. Define

$$P \triangle'' Q = (((P \,|||\, a' \to STOP) \,\Theta_{a'}\, STOP)[\![R]\!]) \underset{\Sigma_1}{\parallel} Q[\![Prime]\!])[\![Unprime]\!]$$

where a' is an arbitrary member of Σ_1 and $R = \{(a', x) \mid x \in \Sigma_1\}$.

The $\mathcal{N}^-$ model shows that one cannot in general express Θ_a in terms of the other operators, but interestingly one can over finite observation models:

Lemma 4. *The following operator is equivalent to Θ_a over $\mathcal{FL}$, and hence over every finite observation model.*

$$P\,\Theta_a{}'\,Q = ((P \triangle (c \to Q[\![Prime]\!])) \underset{\Sigma_0}{\|} Reg_\theta)[\![Unprime]\!] \setminus \{c\}$$
$$Reg_\theta = ?s : (\Sigma_0 - \{a\} \to Reg_\theta) \,\Box\, (a \to c \to STOP)$$

where c is an event not in either Σ_0 or Σ_1.

PROOF This construction allows P to proceed normally until it has performed an a, whereupon (i) Reg_θ blocks P from further visible actions, (ii) the event c is allowed which permits the interrupt to occur and (iii) after this event Q runs. Since the c is the only event available when it happens, and it is hidden, its effects from the outside are invisible. This behaviour is exactly like that of $P\,\Theta_a\,Q$ except that the argument P is discarded at the point when the hidden-c τ occurs, just after the a when it is discarded in $P\,\Theta_a\,Q$. Since that τ can certainly happen, $P\,\Theta_a{}'\,Q$ has all the real, externally-visible behaviours of $P\,\Theta_a\,Q$ in any of our models. The only thing that $P\,\Theta_a{}'\,Q$ can do extra is have P perform τs between the a and the hidden c. This creates a real difference in models where divergence is recorded, since these τs might create divergence. No extra finitely observable behaviour is created however, since $P\,\Theta_a{}'\,Q$ cannot become stable or perform any visible action after the a until the hidden c has occurred. ∎

This last result is reassuring, since it shows us that adding Θ_a gives no extra expressibility over finite-observation models, the domain where [13] succeeded without it.

From now on in this paper we will be considering the language CSP+, and can be safe in the knowledge that adding an extra operator (with respect to which $\mathcal{T}^{\Downarrow}$ is a congruence) cannot invalidate Theorem 4: that result remains true with **f**CSP replaced by **f**CSP+.

6 Stage 2: $\mathcal{N}$ is the weakest proper refinement of $\mathcal{T}^{\Downarrow}$

For this step of the proof it is clear (thanks to the existence of $\mathcal{N}^-$) that Θ_a will need to play a role. As stated earlier, we will use a more technical style of proof since the author has failed to find a way of following the first proof outline here.

We begin with a lemma that has much in common with the ideas used to prove full abstraction results.

In this section and the next, when we write "$P = Q$" or "$P \sqsubseteq Q$" between two **f**CSP terms or finitely branching transition system nodes, we will mean equality or refinement as judged over $\mathcal{FL}^{\Downarrow}$: the most refined relevant model. We will write other forms as $P =_{FD} Q$ or similar (this meaning failures-divergences, in other words equivalence over $\mathcal{N}$). So, in particular, the "=" in the conclusion of the following lemma means equivalence over $\mathcal{FL}^{\Downarrow}$.

Lemma 5.
If $U =_{TD} V$ but $U \not\sqsupseteq_{FD} V$, then there is a context $C[\cdot]$ such that $C[U] = STOP \sqcap (a \to STOP)$ and $C[V] = a \to STOP$.

PROOF Under these assumptions we know that U and V have the same divergence-strict sets of traces and divergences, but that there is some failure (s, X) (necessarily with s not

in the common divergence set and with $X \neq \emptyset$) such that (s, X) is a failure of U but not V. We can assume that the event a can never be communicated by either U or V other than through divergence strictness, since if not we can apply a renaming (perhaps extending the alphabet) to obtain U' and V' satisfying this. Let $\Sigma_0 = \Sigma - \{a\}$.

Let $IdDp = \{(x, b), (x, c) \mid x \in \Sigma_0\}$ be the renaming that maps every member of Σ_0 to a fixed pair of further additional events b and c. Define

$$
\begin{aligned}
FT(\emptyset, Y) &= ?x : \Sigma_0 - Y \rightarrow STOP \\
FT(\langle x \rangle^\frown t, Y) &= (x \rightarrow FT(t, Y)) \rhd a \rightarrow STOP \\
CF0(t, Y)[P] &= (FT(s, Y) \underset{\Sigma_0}{\parallel} P)[\![IdDp]\!] \\
Reg_{FT}(0) &= c \rightarrow STOP \quad \text{and} \quad Reg_{FT}(n+1) = b \rightarrow Reg_{FT}(n) \\
CF1(t, Y)[P] &= (((CF0(t, Y)[P] \underset{\{b,c\}}{\parallel} Reg_{FT}(\#t)) \setminus \{b\}) \Theta_c \, STOP)[\![a/c]\!] \\
CF2(t, Y)[P] &= CF1(t, Y)[P] \sqcap a \rightarrow STOP
\end{aligned}
$$

$CF2(s, X)$ can serve as the context required by the lemma, as we now demonstrate.

Consider first $CF0(s, X)[V]$. This process cannot diverge until perhaps after it has performed one more event than $\#s$, because we know that V cannot on any prefix of s. Imagine the progress of the process V within this context. If it has completed the trace s then, since it cannot then refuse X, it cannot deadlock with $FT(s, X)$ when offered $\Sigma_0 - X$. So in this state there is certainly an action in Σ_0 available at the level of the parallel operator, meaning that some event(s) are offered stably. Thus, after $\#s$ copies of b or c, $CF0(s, X)[V]$ definitely offers $\{b, c\}$.

The effect of $CF1(s, X)[V]$ is to hide the first $\#s$ of these, and only allow the next one to be c, and then turn this into a through renaming. The effect of the Θ_c operator is to cut off this behaviour immediately after this renamed c, in particular ensuring that any divergence of V at that point does not map to a divergence of the context. Any a's arising from the choice in $\rhd$ not to pursue a proper prefix of s remain available: whatever route of internal progress this process follows, a will eventually be offered stably and the process will then $STOP$. Thus $CF1(s, X)[V] = a \rightarrow STOP$ and so $CF2(s, X)[V] = a \rightarrow STOP$ also.

On the other hand $CF0(s, X)[U]$ evidently can deadlock after the trace s inside the renaming, so $CF1(s, X)[U]$ can deadlock on the empty trace thanks to the hiding. Depending on whether $s = \langle\rangle$ and what other refusals U has after s, $CF1(s, X)[U]$ may or may not be able to offer and perform an a. But $CF2(s, X)[U]$ certainly can, meaning that $CF2(s, X)[U] = STOP \sqcap a \rightarrow STOP$ as required. This completes the proof of Lemma 5. ∎

Without the Θ_c, we could have proved an analogous lemma mapping the two processes to $a \rightarrow \mathbf{div}$ and $STOP \sqcap a \rightarrow \mathbf{div}$ but this would not have been strong enough to use in our later proof. Note in particular that this pair of processes are equivalent in $\mathcal{N}^-$.

We are now in a position to prove the main result of this section.

Theorem 5. *Any divergence-strict model $\mathcal{M}$ of $\mathbf{f}CSP+$ that is not $\mathcal{T}^\Downarrow$ is a refinement of $\mathcal{N}$: in other words if $\mathcal{N}$ distinguishes a pair of processes then so does $\mathcal{M}$.*

PROOF We may, following the outline proofs set out in Section 4, assume that P and Q are a pair of processes that are identified by $\mathcal{T}^\Downarrow$, distinguished by $\mathcal{M}$ and such that $P \sqsubseteq Q$. By our assumptions about the nature of divergence-strict models $\mathcal{M}$, we can assume that

$P = P \downarrow N$ and $Q = Q \downarrow N$ for some $N \in \mathbb{N}$. This means that every behaviour of P and Q that is longer than N is implied by one of length N through divergence strictness.

There is a countable infinity of possible members of the two components of a member of $\mathcal{FL}^{\Downarrow}$ thanks to our assumption that the overall alphabet is finite, and the fact that only finite traces are involved. Only finitely many of them have length N or less.

We can therefore list the ones of length N or less that belong to P and not Q as $\beta_1, \beta_2\, \beta_3 \ldots \beta_K$. To enable these behaviours to appear in a single list, we assume the representation of processes as single sets with divergences ending in $\Uparrow$.

By our assumption that P and Q are equivalent in $\mathcal{T}^{\Downarrow}$, it is certain that every β_i contains at least one non-$\bullet$ acceptance. Denote the first position of one in β_i by $fa(i)$ (i.e. if $\beta_i = \langle A_1, a_1, \ldots A_{r-1}, a_{r-1}, A_r\rangle$ then $A_{fa(i)}$ is a proper acceptance and $A_j = \bullet$ for all $j < fa(i)$.

We make a further assumption about this series: if $fa(i) > fa(j)$ then $j < i$. In other words we arrange this finite list so the ones with the most delayed first acceptance come early. This means that if we take β_i and replace the $A_{fa(i)}$ by $\bullet$, then either the resulting behaviour is in Q or it comes earlier in the list.

We will construct a series of processes $Q_i \sqsupseteq P$ where $Q_0 = Q$ and $Q_{n+1} \sqsubseteq Q_n$ has the behaviour β_{n+1}. We need to show how to build Q_{n+1} in general.

If Q_{n+1} already contains β_{n+1} then we need do nothing. Otherwise consider the behaviours Ψ_{n+1} of P that agree with β_{n+1} up to and including the acceptance at $fa(n+1)$.

We know by our choice of enumeration of the β_i, the observation above and elementary consequences of divergence strictness that Q_n contains each $\gamma \in \Psi_{n+1}$ with the acceptance at $fa(n+1)$ replaced by $\bullet$.

Let $Q_{n+1} = Q_n \cup \Psi_{n+1}$. This belongs to $\mathcal{FL}^{\Downarrow}$ and contains β_{n+1}. Theorem 2 means that we do not need to worry about giving a CSP construction for this process, as there is one automatically. This completes our construction of the Q_i. Clearly $Q_K = P$.

Over $\mathcal{M}$, the Q_n cannot all be equivalent, by our assumption that $P \neq_M Q$. So choose n so that Q_{n+1} is the first to be $\mathcal{M}$-inequivalent to Q. It follows that adding Ψ_{n+1} to Q_n creates a process that is different in $\mathcal{M}$ from it.

What we therefore have, in Q_{n+1} and Q_n, are a pair of processes that are differentiated by $\mathcal{M}$, and identified by $\mathcal{T}^{\Downarrow}$, but where the relationship between them is much more constrained than in a general pair such that $P \sqsubset_M Q$ and $P =_{TD} Q$. Now that we have constructed them we will essentially run through the same structure of proof as Theorem 4, with Q_{n+1} and Q_n playing the roles that **div** and Q did there.

We know that Q_n has the behaviour γ which consists of all the actions before $fa(n+1)$ (with all acceptances $\bullet$).

Now add an extra element a to the alphabet of our processes, and let Q^* be the process as Q_{n+1} except that after γ, when offering $A_{fa(n+1)}$, it can additionally perform a (as an addition to acceptance sets), and this a leads to the behaviour Q_n/γ.) Q^* can be defined in terms of CSP operators, Q_{n+1} and Q_n in a similar fashion to our earlier constructions. This extra behaviour is not available if any of the members of γ have been performed from *stable* states.

The crucial properties of Q^* are (i) $Q^* \underset{\{a\}}{\parallel} STOP = Q_{n+1}$ because all the extra behaviour is blocked, and (ii) $Q^* \setminus \{a\} = Q_n$ because this process cannot become stable after γ until after the hidden a.

Let Σ_0 be all visible events other than a.

If $U \neq_{FD} V$ but $U =_{TD} V$ then, by Lemma 5 we can assume without loss of generality that there is $C1[\cdot]$ such that

$$C1[U] = STOP \sqcap a \rightarrow STOP$$
$$C1[V] = a \rightarrow STOP$$

Suppose X is either $a \rightarrow STOP$ or $STOP \sqcap a \rightarrow STOP$ in

$$C2[X] = ((Q^* \underset{\{a\}}{\parallel} X) \setminus \{a\})$$

As Q^* cannot perform a more than once, it is clear that $Q^* \underset{\{a\}}{\parallel} a \rightarrow STOP = Q^*$. It follows by our earlier remarks about $Q^* \setminus \{a\}$ and $Q^* \underset{\{a\}}{\parallel} STOP$ that

- $C2[a \rightarrow STOP] = Q^* \setminus \{a\} = Q_n$
- $C2[STOP \sqcap a \rightarrow STOP] = (Q^* \underset{\{a\}}{\parallel} STOP) \sqcap Q^* \setminus \{a\} = Q_{n+1} \sqcap Q_n = Q_{n+1}$

Let $C[X] = C2[C1[X]]$. Then, by what we have already shown, $C[U] = Q_{n+1}$ and $C[V] = Q_n$. So $C[U] \neq_M C[V]$. This completes the proof of Theorem 5. ∎

7 Stage 3: every proper refinement of $\mathcal{N}$ refines $\mathcal{R}^{\Downarrow}$

The final stage in our proof follows along very similar lines to the second, only just a little bit more intricate. First we establish a lemma very similar to Lemma 5.

Lemma 6.
If $U =_{FD} V$ but $U \not\sqsupseteq_{RD} V$, then there is a context $C[\cdot]$ such that

$$C[U] = STOP \sqcap (a \rightarrow STOP) \text{ and } C[V] = (a \rightarrow STOP) \rhd STOP$$

PROOF Note that, as one would expect, the two result processes here are failures but not revivals equivalent, just as the two used in Lemma 5 are traces but not failures equivalent. These two processes are identical in $\mathcal{FL}^{\Downarrow}$ except that $STOP \sqcap a \rightarrow STOP$ has the observations $\langle \{a\}, a, \bullet \rangle$ and $\langle \{a\}, a, \emptyset \rangle$ unlike $(a \rightarrow STOP) \rhd STOP$, where a can only happen after $\bullet$.

Since U and V are equivalent in $\mathcal{N}$, it follows that they have the same sets of traces, deadlock traces and divergence traces. We know, therefore, that there is some revival (s, X, b) of U but not V. (This means U can perform the trace s, refuse the set X in a stable state, and then perform the visible action $b \notin X$.) On the other hand (s, X) is certainly a failure of V.

The following context forces a process W down the trace s (which is hidden from the outside), then offers both X and b. This may very well deadlock *before* reaching the possibility of X and b.

$$CR1[W] = ((FT(s, \Sigma - (X \cup \{b\})) \underset{\Sigma}{\parallel} W)[\![D]\!] \underset{\Sigma}{\parallel} Reg_R(\#s)) \setminus \Sigma_0, \quad \text{where}$$
$$\Sigma_1 = \{x' \mid x \in \Sigma_0\} \quad \Sigma = \Sigma_0 \cup \Sigma_1$$
$$D = \{(x, x'), (x, x) \mid x \in \Sigma_0\}$$
$$Reg_R(0) = ?x : \Sigma_0 \rightarrow STOP \quad Reg_R(n+1) = ?x : \Sigma_1 \rightarrow Reg_R(n)$$

If $W \in \{U, V\}$ then this process definitely has the trace $\langle b \rangle$, and does not diverge on $\langle\rangle$. If $W = V$ then it can definitely deadlock on the empty trace (because, after s, V can refuse X but not offer b). In this case it might also be able to offer some sets that include b', but definitely not $\{b'\}$ since V cannot offer b without some member of X. If $W = U$ then $CR1[W]$ can definitely offer $\{b'\}$ on $\langle\rangle$, because W can refuse X and then perform b.

$CR1[U]$ and $CR1[V]$ might well diverge after a single event, because U or V can diverge after a trace of the form $s\hat{}\{x\}$ for $x \in X \cup \{b\}$. We can eliminate this possibility using the Θ_a operator, as we had to in Section 6:

$$CR2[W] = (CR1[W][\![R]\!] \,\Theta_c\, STOP)\,\Theta_a\, STOP, \quad \text{where}$$

$$R = \{(x', c) \mid x \in X\} \cup \{(b', a)\}$$

This can now perform the event a from the statement of the lemma when W performs its special b, and an arbitrary fixed event c when W accepts a member of X. Observe that $W[U]$ can offer just $\{a\}$, while if $W[V]$ offers a stably its acceptance set is $\{a, c\}$. Now let

$$CR3[W] = ((a \rightarrow STOP) \rhd STOP) \sqcap CR2[W] \setminus \{c\}$$

Every behaviour of $CR2[V] \setminus \{c\}$ is one of $(a \rightarrow STOP) \rhd STOP$, but since

$$a \rightarrow STOP \sqsupseteq CR3[U] \sqsubseteq STOP \sqcap a \rightarrow STOP$$

and $((a \rightarrow STOP) \rhd STOP) \sqcap a \rightarrow STOP = STOP \sqcap a \rightarrow STOP$ we know that $CR3[U] = STOP \sqcap a \rightarrow STOP$. Thus $CR3[\cdot]$ is the context required by the statement of our lemma.∎

Theorem 6. *Every divergence-strict model $\mathcal{M}$ of* **f***CSP+ that is a proper refinement of $\mathcal{N}$ is in turn a refinement of $\mathcal{R}^{\Downarrow}$.*

PROOF This time we will have a pair of processes such that $P =_{FD} Q$, $P \sqsubset Q$ and $Q \neq_M P$. Again we can assume that $P = P \downarrow N$ and $Q = Q \downarrow N$ for some N, so that the difference between the behaviour sets of P and Q is finite apart from ones implied by divergence strictness. Once again we choose an enumeration $\beta_1, \beta_2, \ldots, \beta_K$ of this difference so that $fa(i) > fa(j) \Rightarrow i < j$.

Notice that, for each i, β_i is a witness for P having the failure $(s, \Sigma - A_{fa(i)})$, where s are the events in β_i preceding the first proper acceptance $A_{fa(i)}$. Since P and Q are failures equivalent, it follows that Q must have a behaviour in which the events of s are followed by a proper acceptance $B_i \subseteq A_{fa(i)}$.

We use exactly the same construction as in the proof of Theorem 5 to create the series of processes Q_i where $Q_0 = Q$, $Q_{n+1} \sqsubseteq Q_n$ contains β_{n+1} and $Q_K = P$. Once again we can therefore concentrate on the first pair Q_n and $Q_{n+1} = Q_n \cup \Psi_{n+1}$ of processes distinguished by $\mathcal{M}$.

Let s be the trace represented by β_{n+1} up to the first non-$\bullet$ acceptance $A\,(= A_{fa(n+1)})$. Let γ be β_{n+1} up to and including this first acceptance A. We know that all the differences between Q_{n+1} and Q_n are extensions of γ, and in particular all the behaviours obtained by changing the first A in a member of Ψ_{n+1} to $\bullet$ are already in Q_n by the structure of our enumeration of the β_i.

As we have done a number of times before, we will extend our alphabet to $\Sigma_0 \cup \Sigma_1$ where $\Sigma_1 = \{x' \mid x \in \Sigma_0\}$ and Σ_0 contains all the events used by our processes. We extend the priming notation x' to sets, behaviours etc, simply meaning it is applied to all their members.

Let ρ be the behaviour that consists of all the events of s preceded by $\bullet$, with the acceptance B at the end, namely the witness in Q of the failure $(s, \Sigma_0 - A)$. And let σ be the same except that the final acceptance is $A' \cup B$. Now define

$$R = Q_n \cup \{\sigma \hat{} \nu^\dagger \mid \gamma \hat{} \nu \in Q_{n+1}\} \cup \{\sigma \hat{} \nu \mid \rho \hat{} \nu \in Q\}$$

where $\nu^\dagger$ is the same as ν except that the first event only is primed.

In other words, R behaves like Q_{n+1} except (i) that events picked from the special acceptance A after s are primed and (ii) that it only has the option to behave outside the range allowed by Q_n when it has offered $A' \cup B$ after ρ, and selecting a member of B leads it to behave like Q would in analogous circumstances.

Consider, then

$$C4[X] = (R \underset{\Sigma_1}{\parallel} X[\![AP]\!])[\![Unprime]\!]$$

where $AP = \{(a, x') \mid x \in B\}$ maps the event a from the statement of Lemma 6 to every member of A'. Note that the parallel composition allows R to run completely freely except that any member of A' is affected by how X offers a if at all.

If $X = (a \rightarrow STOP) \rhd STOP$ then we need to consider two cases of what happens when R has completed s and is offering $A' \cup B$.

- X might deadlock, meaning that R is blocked from performing events from A'. In this case the context just offers B and continues like Q would in the same circumstances.
- X might perform a unstably, meaning that the offer of $A' \cup B$ becomes $\bullet$ in the combination. The continuing behaviour is one of Q if a member of B is chosen, or one of Q_n with one event primed if a member of A' is chosen, the latter because of our choice of enumeration.

It follows easily from this that $C4[(a \rightarrow STOP) \rhd STOP] = Q_n$.

On the other hand, if $X = (a \rightarrow STOP) \sqcap STOP$, then X has the option of offering a stably. This means that R's complete offer of $A' \cup B$ goes forward, which becomes A after the *Unprime* renaming. Since this offer of A can be followed by every behaviour Q_{n+1} can exhibit after γ, it follows that $C4[(a \rightarrow STOP) \sqcap STOP] \sqsubseteq Q_{n+1}$

It would be nice if this were an equality, but it may not be since Q may have behaviours after ρ than Q_{n+1}, and indeed P, need not have after γ. This does not matter in the big picture of our proof, however, since

$$(P_1 \sqsubseteq P_2 \sqsubseteq P_3 \wedge P_2 \neq_M P_3) \Rightarrow P_1 \neq_M P3$$

by the monotonicity of the assumed abstraction map from $\mathcal{FL}^{\Downarrow}$ to $\mathcal{M}$.

It follows that if $U \neq_{RD} V$ then without loss of generality we can, using $C4[C3[\cdot]]$, map U to $C4[(a \rightarrow STOP) \sqcap STOP]$ and Q_n respectively, two processes known to be distinct in $\mathcal{M}$. Hence $U \neq_M V$, so $\mathcal{M}$ refines $\mathcal{R}^{\Downarrow}$. This completes the proof of Theorem 6. ∎

8 Conclusions

In this paper we have given details of the most refined divergence-strict models for both finitary **f**CSP and the language that allows infinite nondeterminism, as well as proposing definitions for what a divergence-strict model looks like in general. We found a rather

counter-intuitive congruence that in essence is created because CSP has no operator of a sort that seems, with the benefit of hindsight, to be natural. We therefore added an extra operator Θ_a from this extra class, creating CSP+. Interestingly, this new operator adds no semantic expressive power over the class of *finite observation* models that the earlier paper [13] considered.

We studied the relationship between finitary models such as $\mathcal{T}^{\Downarrow}$ and $\mathcal{R}^{\Downarrow}$ and their infinitary extensions, in particular proving the uniqueness of this extension for some of more abstract models including all those that play a key role in our structural theorem. We are therefore able to restrict attention, in the proof of that theorem, to the finitary models.

This structural result was completed using three separate Theorems, each a qualified uniqueness theorem for one of the three models we identify as "Platonic". As one would expect, these arguments are sometimes delicate and require many intricate CSP+ contexts to be created.

In [13], the author proved a further result, namely that the stable revivals model $\mathcal{R}$ is the greatest lower bound (as a congruence) of the stable acceptances model $\mathcal{A}$ and the stable refusal testing model $\mathcal{RT}$. A corollary of this result is that the initial linear sequence of models does not continue beyond $\mathcal{R}$. The proof of that result carries forward easily to the class of divergence-strict models, from which we can deduce that the initial sequence is again limited to length 3.

In [13], the author conjectured that the classification problem for CSP models would become significantly more complex once one moves beyond the initial three models, and if one ventures outside the relatively controlled and homogeneous worlds of finite-observation, and divergence-strict models. His suspicion has only grown stronger during the investigations underlying the present paper, both because of something we have written about and something we have not mentioned yet. The first of these was the observation that beyond the realm of Theorem 3 we can expect multiple infinitary extensions of a given **f**CSP model. The second is that we may similarly have freedom to vary how much information we record about divergences: for example, it seems likely that the variant of $\mathcal{FL}^{\Downarrow}$ in which only trace divergences, as opposed to ones with acceptances too, would be a congruence.

Since the results of the present paper and the corresponding ones from [13] were largely unanticipated by the author, he does not exclude the possibility that there may be nice classification results in the reaches beyond revivals. However, he doubts there are!

There is no space in the present paper to report on a fascinating by-product of our work here. That is the idea that our extended language CSP+ can be shown to be a universal language for a wide class of languages of concurrency, namely ones with *CSP-like* operational semantics. Thus, for any such language, all the usual models of CSP together with their refinement properties, and susceptibility to FDR and CSP compression functions, will apply just as much as they do for CSP. The author expects to report on this further work soon.

Appendix: Notation

This paper follows the notation of [10], from which most of the following is taken.

Σ	(Sigma): alphabet of all communications
τ	(tau): the invisible action
Σ^τ	$\Sigma \cup \{\tau\}$
A^*	set of all finite 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
$s \hat{} t$	concatenation of two sequences
$s \leq t$	$(\equiv \exists\, u.s \hat{} u = t)$ prefix order
$\bullet$	non-observation of stability
FLO	the alternating sequences of acceptances/$\bullet$ and members of Σ.

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)
$P[\![a \mapsto A]\!]$	renaming in which a maps to every $b \in A$
$P[\![A \mapsto a]\!]$	renaming in which every member of A maps to a
$P \triangleright Q$	"time-out" operator (sliding choice)
$P \,\triangle\, Q$	interrupt
$P \,\Theta_a\, Q$	exception throwing
$P[x/y]$	substitution (for a free identifier x)
$P \xrightarrow{a} Q$	$(a \in \Sigma \cup \{\tau\})$ single action transition in an LTS

Models:

$\mathcal{T}$	traces model
$\mathcal{N}$	failures/divergences model (divergence strict)
$\mathcal{F}$	stable failures model
$\mathcal{R}$	stable revivals model
$\mathcal{A}$	stable ready sets, or acceptances, model
$\mathcal{RT}$	stable refusal testing model
$\mathcal{FL}$	the finite linear observation model
$\mathcal{M}^\Downarrow$	the model $\mathcal{M}$ extended by strict divergence information
$\mathcal{M}^{\Downarrow,\omega}$	$\mathcal{M}$ extended by strict divergences and infinite traces or similar
$\mathcal{M}^\#$	$\mathcal{M}$ extended by non-strict divergences and infinite traces or similar
$\mathcal{X} \preceq \mathcal{Y}$	$\mathcal{X}$ identifies all processes identified by $\mathcal{Y}$
$\sqsubseteq$	refinement (over $\mathcal{FL}^{\Downarrow\omega}$ by default)

References

1. C. Fournet, C.A.R. Hoare, S.K. Rajamani and J. Rehof, *Stuck-free conformance*, Proceedings CAV 04, 16th International Conference on Computer Aided Verification, Boston, USA, July 2004.
2. R.J. van Glabbeek, *The linear time - Branching time spectrum I* The handbook of process algebra, Elsevier 2001.
3. R.J. van Glabbeek, *The linear time - Branching time spectrum II* Proceedings of CONCUR 1993. Cambridge University Press, 1980.

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. Abida Mukkaram, *A refusal testing model for CSP*, Oxford University D.Phil thesis, 1993.
7. E.R. Olderog and C.A.R. Hoare, *Specification-oriented semantics for communicating processes*, *Acta Informatica*, **23**, 9–66, 1986.
8. I. Phillips, *Refusal testing*, Theoretical Computer Science **50** pp241-284 (1987).
9. A. Puhakka, *Weakest congruence results concerning "any-lock"*, Proc TACAS 2001, Springer LNCS 2215 (2001).
10. A.W. Roscoe, *The theory and practice of concurrency*, Prentice-Hall International, 1998. Updated version available via `web.comlab.ox.ac.uk/oucl/work/bill.roscoe/publications/68b.pdf`
11. 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 appeared in Journal of Logic and Computation **2**, 5 pp557-577.
12. A.W. Roscoe, *Seeing beyond divergence*, in Proceedings of "25 Years of CSP", LNCS3525 (2005).
13. A.W. Roscoe, *Revivals, stuckness and the hierarchy of CSP models*, Submitted for publication. Available at `http://www.comlab.ox.ac.uk/people/bill.roscoe/publications/105.pdf`.