This full text version, available on TeesRep, is the post-print (final version prior to publication) of:


For details regarding the final published version please click on the following DOI link: [http://dx.doi.org/10.1016/j.entcs.2009.12.016](http://dx.doi.org/10.1016/j.entcs.2009.12.016)

When citing this source, please use the final published version as above.

This document was downloaded from [http://tees.openrepository.com/tees/handle/10149/95954](http://tees.openrepository.com/tees/handle/10149/95954)

Please do not use this version for citation purposes.

All items in TeesRep are protected by copyright, with all rights reserved, unless otherwise indicated.
Of wlp and CSP

Steve Dunne
School of Computing
University of Teesside, Middlesbrough, UK
s.e.dunne@tees.ac.uk

Abstract. We extend Morgan’s well-known derivation of the Failures-Divergences semantics of an action system endowed with a wp sequential semantics, by showing how various other CSP semantics can be extracted from an action system endowed with an appropriate sequential semantics. In doing so we expose the close but hitherto largely overlooked correspondence between the various CSP semantic models and their sequential correctness counterparts.

1 Communicating Sequential Processes

CSP or Communicating Sequential Processes [6, 14] is a well-known process algebra typifying the event-based approach to concurrency, in which a process is characterised entirely by its externally observable possible patterns of interaction with its environment via shared primitive events drawn from a specified alphabet of possible such events. Such a purely behavioural characterisation abstracts away from any operational notion of a process as an evolving entity with an evolving internal state which at any given point determines how the process will react to the next stimulus from its environment; rather, it provides a “God’s eye view” of the process whereby all its potential patterns of behaviour are simultaneously exhibited in their entirety. What particular features of this behaviour are actually observed depends on the purposes of the observer. In CSP the observer is deemed to be concerned by safety and liveness properties of his system, so his observations are therefore confined to any or all of the following:

- traces, a trace being any particular sequence of events in which the process is observed successively to engage;
- failures, a failure being a trace leading to a refusal set, i.e. a given set of events simultaneously proposed by the environment in each of which the process refuses to engage;
- divergences, a divergence being a trace after which the process becomes unstable by being as it were livelocked in an endless succession of hidden internal events so that no further meaningful observation of trace or refusal behaviour is possible.

If \( \Sigma \) is the event alphabet of the process concerned then \( \Sigma^* \) denotes the set of all finite traces on \( \Sigma \), i.e. the set of all finite sequences of events in \( \Sigma \) including
the empty trace \( \langle \rangle \). Each failure of a process is characterised by an ordered pair \((tr, X)\) where \(tr\) is a finite trace and \(X\) is a refusal, \(i.e.\) a set of events simultaneously refused by the process. Hence the failures of process collectively comprise a relation of type \(\Sigma^* \to \mathcal{P} \Sigma\), known as the failures relation of that process. In this paper we restrict our attention to (at most) finitely nondeterministic processes, so we don’t need to consider infinite traces.

1.1 CSP semantic models

There are several recognised semantic models in the CSP literature, each of which induces its own particular congruence over process terms of the CSP language, a congruence being an equivalence relation which is compositional with respect to the language operators. Each of these congruences is fully abstract with respect to some characteristic simple but significant operational test which usefully distinguishes processes in some way, see [14][Thm 9.3.1]. For describing these models it is useful to introduce at this point the CSP processes Stop, which deadlocks immediately without diverging, and Div which immediately diverges.

- **Traces** (TR) this is the simplest of our recognised models. In it each process \(P\) is denoted by its set of traces \(\text{traces}(P)\). This model completely ignores divergent behaviour, even to the extent that it equates Div with Stop. It is sufficient for reasoning about safety, \(i.e.\) whether any given event can occur, but not about liveness, \(i.e.\) in what circumstances any given event must occur.

- **Failures-Divergences** (FD) this is the de facto standard semantic model for finitely nondeterministic CSP. In contrast to TR it takes a drastic view of divergence which regards a process as utterly unpredictable, and therefore capable of any behaviour, once divergence has occurred. Thus FD interprets Div as the least reliable of all processes, with the consequence that whenever divergence is even a possibility any other specific behaviour associated with the process at that point is completely occluded by all the possible behaviours associated with divergence. In this model each finitely nondeterministic process \(P\) is characterised by its divergence-augmented failures relation \(\text{failures}_{\perp}(P)\), and its extension-closed set of divergences \(\text{divergences}_{\perp}(P)\).

- **Stable Failures** (SF) although less prominent in the CSP literature than FD, this model is theoretically and practically significant as the weakest congruence which respects deadlock [17]. Although not quite as indifferent to divergence as TR, to quote from [15] SF certainly “turns something of blind eye” to divergence, and so, in contrast to FD, permits other specific behaviour the process may possess alongside the divergence to be discerned. Although it deems Stop and Div to have the same traces, in fact just the empty trace \(\langle \rangle\), it distinguishes between them on the basis of their refusals, Stop initially refusing everything, but Div having no refusals since it never even achieves a stable initial state from which to refuse anything. The properties of SF are analysed in detail by Roscoe in [14], although Valmari [19]
attributes its origins back as far as [2]. In the SF model each process $P$ is characterised by its traces $\text{traces}(P)$ and its stable failures, i.e. its ordinary non-divergence-augmented failures $\text{failures}(P)$.

- **Chaos-free Failures Divergences (CFFD)** this model was originally introduced by Valmari [19] for finitely nondeterministic processes only, then later extended by him to encompass unboundedly nondeterministic processes too [18]. In this paper we consider only the earlier version of the model, in which each finitely nondeterministic process $P$ is characterised by its minimal, i.e. non-extension-closed, divergences $\text{divergences}(P)$ and its stable failures $\text{failures}(P)$.

1.2 **Refinement relationships between the models**

![Refinement Graph of the CSP semantic models](image)

Figure 1 depicts the refinement relationships between the four CSP semantic models above. Generally speaking the more refined the model the more distinctions it makes between processes. Any of these semantic models is said to refine another if it makes a least as many distinctions between processes as the latter. Thus any pair of processes which are distinguished by the coarser model being refined must also be distinguished by the refining model, while conversely any pair of processes equated by the refining model must also be equated by the coarser model being refined.
Not all pairs of models are refinement-comparable. For example, SF and FD are incomparable, since each makes distinctions between some processes which the other equates. Neither does FD refine TR: although it is tempting to think that in FD the traces of a process can be extracted as the domain of its failures relation, in fact this is not so because the traces so obtained, unlike those in TR, will include all finite extensions of minimal divergences.

As can be seen in Figure 1, the most refined of our models is CFFD since this can distinguish between any processes which any of the other models can. For example CFFD distinguishes between all three of the process \( \text{Stop} \), \( \text{Div} \) and \( \text{Stop} \sqcap \text{Div} \), whereas SF equates \( \text{Stop} \sqcap \text{Div} \) with \( \text{Stop} \) while FD equates it with \( \text{Div} \).

2 State-based Representations of Processes

A duality has long been recognised between the purely behavioural description of a concurrent process discussed in Section 1 and its corresponding more operational state-based description, for example as a labelled transition system (LTS) [20], action system [1] or abstract data type (ADT) [3, 16]. The first explicit state-based characterisations of a behavioural process to attract widespread attention were those of Josephs [11] describing how a non-divergent CSP process could be represented as an LTS, and of He [8] who developed a similar state-based relational process model which did accommodate divergence. The motivation in both cases was to provide a state-based process model in respect of which it would be possible to adapt He et al’s ADT refinement proof methods of forward and backward simulation [10] to the failures-divergences refinement of processes.

Soon afterwards Morgan [12] showed how to interpret an action system as a CSP process by describing precisely how to derive its failures and divergences from its wp semantics. What was particularly striking about Morgan’s approach is the way the phenomenon of divergence is handled so uniformly by it; the wp semantics gives us divergence at no extra cost, so to speak.

2.1 Morgan’s behavioural interpretation of an action system

We start with an action system \( A \) endowed with a total-correctness semantics, more specifically a weakest precondition (wp) predicate-transformer semantics. \( A \) therefore possesses a state \( S \) together with both an initialisation \( \text{init} \) and a repertoire of actions \( a_i \ (i \in I) \) on \( S \) which can fire individually when enabled and modify \( S \). The initialisation \( \text{init} \) and each of the actions \( a_i \) is characterised as a wp predicate transformer. We will describe a predicate transformer as \emph{drastic} if it maps all predicates to either of the extreme predicates true or false. The initialisation \( \text{init} \) of an action system is always characterised as a drastic wp predicate transformer, and it must be fully \emph{feasible}, i.e. strict with respect to false, so that it is always enabled.

The sequential composition \( a_1 ; a_2 \) of two actions is defined by

\[
\wp(a_1 ; a_2, q) = \wp(a_1 , \wp(a_2 , q)) \quad \text{seq comp}
\]
For convenience we introduce the notion of a conjugate wp predicate transformer, denoted cwp, where given an action \( a \) and a postcondition \( q \) we have

\[
\text{cwp}(a, q) = df \quad \neg \text{wp}(a, \neg q)
\]

conjugate wp

We note that whereas wp is positively conjunctive for demonically nondeterministic programs since it distributes through all non-empty conjunctions of postconditions, cwp is positively disjunctive since it distributes through all non-empty disjunctions of postconditions.

We also define the following characteristic predicates of an action \( a \)

\[
\text{trm}(a) = df \quad \text{wp}(a, \text{true})
\]

termination

\[
\text{fis}(a) = df \quad \text{cwp}(a, \text{true})
\]

feasibility guard

We extend the domain of fis in an obvious way to sets of actions:

\[
\text{fis}(X) = df \quad \bigvee_{a \in X} \text{fis}(a)
\]

overall feasibility

In particular, since the empty disjunction is trivially false, we have fis(\( \{\} \)) = false.

We can now describe how Morgan [12] extracts the FD semantics of the action system \( A \):

- A sequence \( \langle a_1, a_2, \ldots, a_n \rangle \) of actions is a trace of action system \( A \) precisely if fis(init; \( a_1; a_2; \ldots; a_n \)) holds.

- A sequence \( \langle a_1, a_2, \ldots, a_n \rangle \) of actions is a divergence of action system \( A \) precisely if \( \neg \text{trm}(\text{init}; a_1; a_2; \ldots; a_n) \) holds.

- A pair \( \langle \langle a_1, a_2, \ldots, a_n \rangle, X \rangle \), where \( \langle a_1, a_2, \ldots, a_n \rangle \) is a sequence of actions and \( X \) is a set of actions, is a failure of action system \( A \) precisely if cwp(init; \( a_1; a_2; \ldots; a_n; \neg \text{fis}(X) \)) holds.

Note that the drastic nature of \( A \)'s initialisation \( \text{init} \) ensures that all three conditions above are absolute rather than contingent. That is, each of them is either wholly true or wholly false.

2.2 Limitation of total correctness

Given the striking mathematical elegance and conciseness of Morgan’s method of extracting the FD semantics of an action system described in the previous subsection, the question naturally arises of whether a semantics corresponding to any of the other semantic models we described in Section 1 can be similarly extracted from the same action system. The answer is no, as long as the wp semantics with which our action system in Section 2.1 is endowed is a total-correctness semantics. Such a semantics doesn’t provide a rich enough representation of the actual operational behaviour of the action system to enable us to extract its denotations in the other semantic models TR, SF and CFFD.

In fact, as we will see, to extract the TR semantics of an action system we require at least a partial-correctness specification of all its actions, while
to extract its CFFD semantics we need a full general-correctness specification of all its actions. Even more interestingly, to extract the SF semantics of an action system it turns out we require a specification of all its actions in a form of correctness strictly intermediate between partial and general correctness, for which as far as we are aware no-one has ever coined a name. For want of a better name, therefore, we will call this simply intermediate correctness. We will describe all these various concepts of correctness for sequential computations in detail in the next section.

3 Concepts of Sequential Correctness

When Dijkstra [4] invented his weakest-precondition (wp) predicate-transformer semantics he also introduced (but seems at first to have made little use of) a second predicate transformer which he called a weakest-liberal-precondition (wlp) predicate transformer; while wp provides a total-correctness semantics, wlp in contrast provides a partial-correctness semantics. Unlike wp, wlp is strict with respect to true, i.e. for any program a we have wlp(a, true) = true. So whereas wp is positively conjunctive for demonically nondeterministic programs, wlp is universally conjunctive for those programs since it also distributes through the vacuously true empty conjunction. In combination wp and wlp provide a general-correctness semantics [7, 13, 5]. The two are linked by the following rule

\[ wp(a, q) = wp(a, true) \land wlp(a, q) \]

Since wp(a, true) is also denoted by trm(a) we can exploit this linkage rule by choosing to deem trm and wlp as the fundamental components of our general-correctness semantics while relegating wp to the status of a derived component:

\[ wp(a, q) = df trm(a) \land wlp(a, q) \]

For convenience we also introduce the notion of conjugate wlp predicate transformer, denoted by cwlp, where given an action a and a postcondition q we have

\[ cwlp(a, q) = df \neg wlp(a, \neg q) \]

We note that cwlp is universally disjunctive for demonically nondeterministic programs.

We also define the following further characteristic predicates of an action a in the context of general correctness:

\[ fec(a) = df cwlp(a, true) \]

\[ fit(a) = df trm(a) \land fec(a) \]

We extend the domain of fec in the obvious way to sets of actions:

\[ fec(X) = df \bigvee_{a \in X} fec(a) \]
Again, since the empty disjunction is trivially false, we have in particular that \( \text{fec}(\{\}) = \text{false} \). We are now in a position to characterise the various sequential correctness concepts we will subsequently need:

- \( \text{wlp} \): partial correctness
- \( \text{wp} \): total correctness
- \( \text{wlp, trm} \): general correctness
- \( \text{wlp, fit} \): intermediate correctness

The relationship between these correctness concepts is depicted in the graph shown in Figure 2. The astute reader will notice the close resemblance in shape between this graph and the one in Figure 1. This resemblance is no accident.

![Graph showing the relationship between sequential correctness concepts](image)

**Fig. 2.** Relationship between sequential correctness concepts

### 4 Extracting the other CSP semantics

In this section we show how its TR, SF and CFFD semantics can respectively be extracted from an action system endowed with an appropriate sequential-correctness semantics. But first we introduce a further extension of our notation. If \( tr \) is a sequence of actions \( \langle a_1, a_2, \ldots, a_n \rangle \) we will write \( \text{cwlp}(\text{init} ; tr, q) \) to denote \( \text{cwlp}(\text{init} ; a_1 ; a_2 ; \ldots ; a_n, q) \). In the special case where \( tr \) is the empty sequence \( \text{cwlp}(\text{init} ; tr, q) \) means \( \text{cwlp}(\text{init}, q) \).
4.1 TR semantics

We start with an action system $A$ endowed with a partial-correctness semantics, specifically a wlp semantics, and whose initialisation $\text{init}$ is a drastic wlp predicate transformer such that $\text{fec}(\text{init})$ holds. A sequence $tr$ of actions is a trace of $A$ precisely if $\text{fec}(\text{init} ; tr)$ holds.

4.2 SF semantics

We start with an action system $A$ endowed with an intermediate-correctness semantics, specifically a (wlp, fit) semantics, and whose initialisation $\text{init}$ is a drastic wlp predicate transformer such that $\text{fec}(\text{init})$ holds. The traces and stable failures of $A$ can be extracted as follows, noting that we separate the cases of a stable failure associated with an empty trace and a non-empty trace:

- A sequence $tr$ of actions is a trace of $A$ precisely if $\text{fec}(\text{init} ; tr)$ holds.

- An ordered pair $(\langle \rangle , X)$ is a stable failure of $A$ if $\text{fit}(\text{init})$ and $\text{cwlp}(\text{init} , \neg \text{fec}(X))$ both hold.

- An ordered pair $(tr \rightarrow \langle a \rangle , X)$ is a stable failure of $A$ if $\text{cwlp}(\text{init} ; tr , (\text{fit}(a) \land \text{cwlp}(a , \neg \text{fec}(X))))$ holds.

4.3 CFFD semantics

We start with an action system $A$ endowed with a general-correctness semantics, specifically a (wlp, trm) semantics, and whose initialisation $\text{init}$ is a drastic wlp predicate transformer such that $\text{fec}(\text{init})$ holds. The stable failures of $A$ can be extracted as per the SF semantics in the previous subsection. The divergences of $A$ can be extracted as follows, noting that we separate the cases of an empty-trace and a non-empty-trace divergence:

- $\langle \rangle$ is a divergence of $A$ if $\neg \text{trm}(\text{init})$ holds;

- $tr \rightarrow \langle a \rangle$ is a divergence of $A$ if $\text{cwlp}(\text{init} ; tr , \neg \text{trm}(a))$ holds.

5 Conclusion

We have extended Morgan’s derivation of the Failures-Divergences semantics of an action system endowed with a wp sequential semantics, by showing how various other CSP semantics can be extracted from an action system which is endowed in each case with an appropriate corresponding sequential semantics other than wp. In doing so we have exposed the close but perhaps hitherto largely overlooked correspondence which exists between the various CSP semantic models and their sequential-correctness counterparts.

References
