A Branching Time Model of CSP

Rob van Glabbeek[1,2(✉)]

[1] Data61, CSIRO, Sydney, Australia
[2] Computer Science and Engineering,
University of New South Wales, Sydney, Australia
rvg@unsw.edu.au

Abstract. I present a branching time model of CSP that is finer than all other models of CSP proposed thus far. It is obtained by taking a semantic equivalence from the linear time – branching time spectrum, namely divergence-preserving coupled similarity, and showing that it is a congruence for the operators of CSP. This equivalence belongs to the bisimulation family of semantic equivalences, in the sense that on transition systems without internal actions it coincides with strong bisimilarity. Nevertheless, enough of the equational laws of CSP remain to obtain a complete axiomatisation for closed, recursion-free terms.

1 Introduction

The process algebra CSP—*Communicating Sequential Processes*—was presented in Brookes, Hoare & Roscoe [4]. It is sometimes called *theoretical* CSP, to distinguish it from the earlier language CSP of Hoare [10]. It is equipped with a denotational semantics, mapping each CSP process to an element of the failures-divergences model [4,5]. The same semantics can also be presented operationally, by mapping CSP processes to states in a labelled transition system (LTS), and then mapping LTSs to the failures-divergences model. Olderog & Hoare [13] shows that this yields the same result. Hence, the failures-divergences model of CSP can alternatively be seen as a semantic equivalence on LTSs, namely by calling two states in an LTS equivalent iff they map to the same element of the failures-divergences model.

Several other models of CSP are presented in the literature, and each can be cast as a semantic equivalence on LTSs, which is a congruence for the operators of CSP. One such model is called *finer* than another if its associated equivalence relation is finer, i.e., included in the other one, or more discriminating. The resulting hierarchy of models of CSP has two pillars: the divergence-strict models, most of which refine the standard failures-divergences model, and the stable models, such as the model based on stable failures equivalence from Bergstra, Klop & Olderog [2], or the stable revivals model of Roscoe [16].

Here I present a new model, which can be seen as the first branching time model of CSP, and the first that refines all earlier models, i.e. both pillars mentioned above. It is based on the notion of coupled similarity from Parrow & Sjödin [14]. What makes it an interesting model of CSP—as opposed to, say,

T. Gibson-Robinson et al. (Eds.): Roscoe Festschrift, LNCS 10160, pp. 272–293, 2017.
DOI: 10.1007/978-3-319-51046-0_14

strong or divergence-preserving weak bisimilarity—is that it allows a complete equational axiomatisation for closed recursion-free CSP processes that fits within the existing syntax of that language.

2 CSP

CSP [4,5,11] is parametrised with a set Σ of *communications*. In this paper I use the subset of CSP given by the following grammar.

$$P, Q ::= STOP \mid \mathbf{div} \mid a \to P \mid P \sqcap Q \mid P \,\Box\, Q \mid P \rhd Q \mid P \|_A Q \mid P \backslash A \mid f(P) \mid P \vartriangle Q \mid P \,\Theta_A\, Q \mid p \mid \mu p.P$$

Here P and Q are CSP expressions, $a \in \Sigma$, $A \subseteq \Sigma$ and $f : \Sigma \to \Sigma$. Furthermore, p ranges over a set of *process identifiers*. A CSP *process* is a CSP expression in which each occurrence of a process identifier p lays within a recursion construct $\mu p.P$. The operators in the above grammar are *inaction*, *divergence*, *action prefixing*, *internal*, *external* and *sliding choice*, *parallel composition*, *concealment*, *renaming*, *interrupt* and *throw*. Compared to [15,17], this leaves out

- successful termination (*SKIP*) and sequential composition (;),
- infinitary guarded choice,
- prefixing operators with name binding, conditional choice,
- relational renaming, and
- the version of internal choice that takes a possibly infinite set of arguments.

The operators $STOP$, $a \to$, $\sqcap$, $\Box$, $\backslash A$, $f(\_)$ and recursion stem from [4], and **div** and $\|_A$ from [13], whereas $\rhd$, $\vartriangle$ and Θ_A were added to CSP by Roscoe [15,17]. The operational semantics of CSP is given by the binary transition relations $\xrightarrow{\alpha}$ between CSP processes. The transitions $P \xrightarrow{\alpha} Q$ are derived by the rules in Table 1. Here a, b range over Σ and α, β over $\Sigma \dot\cup \{\tau\}$, and relabelling operators f are extended to $\Sigma \dot\cup \{\tau\}$ by $f(\tau) = \tau$. The transition labels α are called *actions*, and τ is the *internal action*.

3 The Failures-Divergences Model of CSP

The process algebra CSP stems from Brookes, Hoare & Roscoe [4]. It is also called *theoretical* CSP, to distinguish it from the language CSP of Hoare [10]. Its semantics [5] associates to each CSP process a pair $\langle F, D\rangle$ of *failures* $F \subseteq \Sigma^* \times \mathscr{P}(\Sigma)$ and *divergences* $D \subseteq \Sigma^*$, subject to the conditions:

$$(\varepsilon, \emptyset) \in F \tag{N1}$$
$$(st, \emptyset) \in F \Rightarrow (s, \emptyset) \in F \tag{N2}$$
$$(s, X) \in F \wedge Y \subseteq X \Rightarrow (s, Y) \in F \tag{N3}$$
$$(s, X) \in F \wedge \forall c \in Y.\, (sc, \emptyset) \notin F \Rightarrow (s, X \cup Y) \in F \tag{N4}$$
$$\forall Y \in \mathscr{P}_{fin}(X).\, (s, Y) \in F \Rightarrow (s, X) \in F \tag{N5}$$
$$s \in D \Rightarrow st \in D \tag{D1}$$
$$s \in D \Rightarrow (st, X). \tag{D2}$$

Table 1. Structural operational semantics of CSP

$\mathbf{div} \xrightarrow{\tau} \mathbf{div}$	$(a \to P) \xrightarrow{a} P$	$P \sqcap Q \xrightarrow{\tau} P$	$P \sqcap Q \xrightarrow{\tau} Q$
$\dfrac{P \xrightarrow{a} P'}{P \,\Box\, Q \xrightarrow{a} P'}$	$\dfrac{P \xrightarrow{\tau} P'}{P \,\Box\, Q \xrightarrow{\tau} P' \,\Box\, Q}$	$\dfrac{Q \xrightarrow{a} Q'}{P \,\Box\, Q \xrightarrow{a} Q'}$	$\dfrac{Q \xrightarrow{\tau} Q'}{P \,\Box\, Q \xrightarrow{\tau} P \,\Box\, Q'}$
$\dfrac{P \xrightarrow{a} P'}{P \triangleright Q \xrightarrow{a} P'}$	$\dfrac{P \xrightarrow{\tau} P'}{P \triangleright Q \xrightarrow{\tau} P' \triangleright Q}$	$P \triangleright Q \xrightarrow{\tau} Q$	$\dfrac{P \xrightarrow{\alpha} P'}{f(P) \xrightarrow{f(\alpha)} f(P')}$
$\dfrac{P \xrightarrow{\alpha} P' \ (\alpha \notin A)}{P \|_A Q \xrightarrow{\alpha} P' \|_A Q}$	$\dfrac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q' \ (a \in A)}{P \|_A Q \xrightarrow{a} P' \|_A Q'}$		$\dfrac{Q \xrightarrow{\alpha} Q' \ (\alpha \notin A)}{P \|_A Q \xrightarrow{\alpha} P \|_A Q'}$
$\dfrac{P \xrightarrow{\alpha} P' \ (\alpha \notin A)}{P \backslash A \xrightarrow{\alpha} P' \backslash A}$	$\dfrac{P \xrightarrow{a} P' \ (a \in A)}{P \backslash A \xrightarrow{\tau} P' \backslash A}$	$\dfrac{P \xrightarrow{\alpha} P' \ (\alpha \notin A)}{P \,\Theta_A\, Q \xrightarrow{a} P' \,\Theta_A\, Q}$	$\dfrac{P \xrightarrow{a} P' \ (a \in A)}{P \,\Theta_A\, Q \xrightarrow{a} Q}$
$\dfrac{P \xrightarrow{\alpha} P'}{P \triangle Q \xrightarrow{\alpha} P' \triangle Q}$	$\dfrac{Q \xrightarrow{\tau} Q'}{P \triangle Q \xrightarrow{\tau} P' \triangle Q'}$	$\dfrac{Q \xrightarrow{a} Q'}{P \triangle Q \xrightarrow{a} Q'}$	$\mu p.P \xrightarrow{\tau} P[\mu p.P/p]$

Here $\varepsilon \in \Sigma^*$ is the empty sequence of communications and st denotes the concatenation of sequences s and $t \in \Sigma^*$. If $\langle F, D\rangle$ is the semantics of a process P, $(s, \emptyset) \in F$, with $s \notin D$, tells that P can perform the sequence of communications s, possibly interspersed with internal actions. Such a sequence is called a *trace* of P, and Conditions N1 and N2 say that the set of traces of any processes is non-empty and prefix-closed. A failure $(s, X) \in F$, with $s \notin D$, says that after performing the trace s, P may reach a state in which it can perform none of the actions in X, nor the internal action. A communication $x \in \Sigma$ is thought to occur in cooperation between a process and its environment. Thus $(s, X) \in F$ indicates that deadlock can occur if after performing s the process runs in an environment that allows the execution of actions in X only. From this perspective, Conditions N3 and N4 are obvious.

A divergence $s \in D$ is a trace after which an infinite sequence of internal actions is possible. In the failures-divergences model of CSP divergence is regarded *catastrophic*: all further information about the process' behaviour past a divergence trace is erased. This is accomplished by *flooding*: all conceivable failures (st, X) and divergences st that have s as a prefix are added to the model (regardless whether P actually has a trace st).

A CSP process P from the syntax of Sect. 2 has the property that for any trace s of P, with $s \notin D$, the set $next(s)$ of actions c such that sc is also a trace of P is finite. By (N3–4), $(s, X) \in F$ iff $(s, X \cap next(s)) \in F$. It follows that if $(s, X) \notin F$, then there is a finite subset Y of X, namely $X \cap next(s)$, such that $(s, Y) \notin F$. This explains Condition (N5).

In Brookes & Roscoe [5] the semantics of CSP is defined denotationally: for each n-ary CSP operator Op, a function is defined that extracts the failures and divergences of $Op(P_1, \ldots, P_n)$ out of the failures and divergences of the

argument processes $P_1, \ldots, P_n$. The meaning of a recursively defined CSP process $\mu p.P$ is obtained by means of fixed-point theory. Alternatively, the failures and divergences of a CSP process can be extracted from its operational semantics:

Definition 1. Write $P \Longrightarrow Q$ if there are processes $P_0, \ldots, P_n$, with $n \geq 0$, such that $P = P_0$, $P_i \xrightarrow{\tau} P_{i+1}$ for all $0 \leq i < n$, and $P_n = Q$.

Write $P \stackrel{\alpha}{\Longrightarrow} Q$ if there are processes P', Q' with $P \Longrightarrow P' \xrightarrow{\alpha} Q' \Longrightarrow Q$.

Write $P \stackrel{\hat{\alpha}}{\Longrightarrow} Q$ if either $\alpha \in \Sigma$ and $P \stackrel{\alpha}{\Longrightarrow} Q$, or $\alpha = \tau$ and $P \Longrightarrow Q$.

Write $P \stackrel{s}{\Longrightarrow} Q$, for $s = a_1 a_2 \ldots a_n \in \Sigma^*$ with $n \geq 0$, if there are processes $P_0, \ldots, P_n$ such that $P = P_0$, $P_i \stackrel{a_i}{\Longrightarrow} P_{i+1}$ for all $0 \leq i < n$, and $P_n = Q$.

Let $I(P) = \{\alpha \in \Sigma \cup \{\tau\} \mid \exists Q.\, P \xrightarrow{\alpha} Q\}$.

Write $P\Uparrow$ if there are processes P_i for all $i \geq 0$ with $P \stackrel{s}{\Longrightarrow} P_0 \xrightarrow{\tau} P_1 \xrightarrow{\tau} \ldots$.

$s \in \Sigma^*$ is a *divergence trace* of a process P if there is a Q with $P \stackrel{s}{\Longrightarrow} Q\Uparrow$.

The *divergence set* of P is $\mathscr{D}(P) := \{st \mid s \text{ is a divergence trace of } P\}$.

A *stable failure* of a process P is a pair $(s, X) \in \Sigma^* \times \mathscr{P}(\Sigma)$ such that $P \stackrel{s}{\Longrightarrow} Q$ for some Q with $I(Q) \cap (X \cup \{\tau\}) = \emptyset$. The *failure set* of a process P is $\mathscr{F}(p) = \{(s, X) \mid s \in \mathscr{D}(P) \text{ or } (s, X) \text{ is a stable failure of} P\}$.

The semantics $[\![P]\!]_{\mathscr{F}\mathscr{D}}$ of a CSP process P is the pair $\langle \mathscr{F}(P), \mathscr{D}(P) \rangle$.

Processes P and Q are *failures-divergences equivalent*, notation $P \equiv_{FD} Q$, iff $[\![P]\!]_{\mathscr{F}\mathscr{D}} = [\![Q]\!]_{\mathscr{F}\mathscr{D}}$. Process P is a *failures-divergences refinement* of Q, notation $P \sqsupseteq_{FD} Q$, iff $\mathscr{F}(P) \subseteq \mathscr{F}(Q) \wedge \mathscr{D}(P) \subseteq \mathscr{D}(Q)$.

The operational semantics of Sect. 2 (then without the operators $\triangleright$, $\triangle$ and Θ_A) appears, for instance, in [13], and was created after the denotational semantics. In Olderog & Hoare [13] it is shown that the semantics $[\![P]\!]$ of a CSP process defined operationally through Definition 1 equals the denotational semantics given in [5]. The argument extends smoothly to the new operators $\triangleright$, $\triangle$ and Θ_A [17]. This can be seen as a justification of the operational semantics of Sect. 2.

In Brookes, Hoare & Roscoe [4] a denotational semantics of CSP was given involving failures only. Divergences were included only implicitly, namely by thinking of a trace s as a divergence of a process P iff P has all failures (st, X). So the semantics of **div** or $\mu X.X$ is simply the set of all failure pairs. As observed in De Nicola [6], this approach invalidates a number of intuitively valid laws, such as $P \,\Box\, \mathbf{div} = \mathbf{div}$. The improved semantics of [5] solves this problem.

In Hoare [11] a slightly different semantics of CSP is given, in which a process is determined by its failures, divergences, as well as its *alphabet*. The latter is a superset of the set of communications the process can ever perform. Rather than a parallel composition $\|_A$ for each set of synchronising actions $A \subseteq \Sigma$, this approach has an operator $\|$ where the set of synchronising actions is taken to be the intersection of the alphabets of its arguments. Additionally, there is an operator $|||$, corresponding to $\|_\emptyset$. This approach is equally expressive as the one of [5], in the sense that there are semantics preserving translations in both directions. The work reported in this paper could just as well have been carried out in this *typed* version of CSP.

4 A Complete Axiomatisation

In [4–6,11,15,17] many algebraic laws $P = Q$, resp. $P \sqsubseteq Q$, are stated that are *valid* w.r.t. the failures-divergences semantics of CSP, meaning that $P \equiv_{FD} Q$, resp. $P \sqsubseteq_{FD} Q$. If Th is a collection of equational laws $P = Q$ then $Th \vdash R = S$ denotes that the equation $R = S$ is derivable from the equations in Th using reflexivity, symmetry, transitivity and the rule of congruence, saying that if Op is an n-ary CSP operator and $P_i = Q_i$ for $i = 1, \ldots, n$ then $Op(P_1, \ldots, P_n) = Op(Q_1, \ldots, Q_n)$. Likewise, if Th is a collection of inequational laws $P \sqsubseteq Q$ then $Th \vdash R \sqsubseteq S$ denotes that the inequation $R \sqsubseteq S$ is derivable from the inequations in Th using reflexivity, transitivity and the rule saying that if Op is an n-ary CSP operator and $P_i \sqsubseteq Q_i$ for $i = 1, \ldots, n$ then $Op(P_1, \ldots, P_n) \sqsubseteq Op(Q_1, \ldots, Q_n)$.

Definition 2. An equivalence $\sim$ on process expressions is called a *congruence* for an n-ary operator Op if $P_i \sim Q_i$ for $i = 1, \ldots, n$ implies $Op(P_1, \ldots, P_n) \sim Op(Q_1, \ldots, Q_n)$. A preorder $\preceq$ is a *precongruence* for Op, or Op is *monotone* for $\preceq$, if $P_i \preceq Q_i$ for $i = 1, \ldots, n$ implies $Op(P_1, \ldots, P_n) \preceq Op(Q_1, \ldots, Q_n)$.

If $\sim$ is a congruence for all operators of CSP (resp. $\preceq$ is a precongruence for all operators of CSP) and Th is a set of (in)equational laws that are valid for $\sim$ (resp. $\preceq$) then any (in)equation $R = S$ with $Th \vdash R = S$ (resp. $R \sqsubseteq S$ with $Th \vdash R \sqsubseteq S$) is valid for $\sim$ (resp. $\preceq$).

$\equiv_{FD}$ is a congruence for all operators of CSP. This follows immediately from the existence of the denotational failures-divergences semantics. Likewise, $\sqsubseteq_{FD}$ is a precongruence for all operators of CSP [4–6,11,13,15,17].

Definition 3. A set Th of (in)equational laws—an *axiomatisation*—is *sound and complete* for an equivalence $\sim$ (or a preorder $\preceq$) if $Th \vdash R = S$ iff $R \sim S$ (resp. $Th \vdash R \sqsubseteq S$ iff $R \preceq S$). Here "$\Rightarrow$" is *soundness* and "$\Leftarrow$" completeness.

In De Nicola [6] a sound and complete axiomatisation of $\sqsubseteq_{FD}$ for recursion-free CSP, and no process identifiers or variables, is presented. It is quoted in Table 2. As this axiomatisation consist of a mix of equations and inequations, formally it is an inequational axiomatisation, where an equation $P = Q$ is understood as the conjunction of $P \sqsubseteq Q$ and $Q \sqsubseteq P$. This mixed use is justified because $\equiv_{FD}$ is the *kernel* of $\sqsubseteq_{FD}$: one has $P \equiv_{FD} Q$ iff $P \sqsubseteq_{FD} Q \wedge Q \sqsubseteq_{FD} P$.

In [6], following [4,5], two parallel composition operators $\|$ and $|||$ were considered, instead of the parametrised operator $\|_A$. Here $\| = \|_\Sigma$ and $||| = \|_\emptyset$. In Table 2 the axioms for these two operators are unified into an axiomatisation of $\|_A$. Additionally, I added axioms for sliding choice, renaming, interrupt and throw—these operators were not considered in [6]. The associativity of parallel composition (Axiom **P0**) is not included in [6] and is not needed for completeness. I added it anyway, because of its importance in equational reasoning.

The soundness of the axiomatisation of Table 2 follows from $\sqsubseteq_{FD}$ being a precongruence, and the validity of the axioms—a fairly easy inspection using the denotational characterisation of $[\![\_]\!]$. To obtain completeness, write $\Box_{i \in I} P_i$, with

Table 2. A complete axiomatisation of $\sqsubseteq_{FD}$ for recursion-free CSP

$\bot$	$\mathbf{div} \sqsubseteq P$	
I1	$P \sqcap P = P$	
I2	$P \sqcap Q = Q \sqcap P$	
I3	$P \sqcap (Q \sqcap R) = (P \sqcap Q) \sqcap R$	
I4	$P \sqcap Q \sqsubseteq P$	
E1	$P \,\Box\, P = P$	
E2	$P \,\Box\, Q = Q \,\Box\, P$	
E3	$P \,\Box\, (Q \,\Box\, R) = (P \,\Box\, Q) \,\Box\, R$	
E4	$P \,\Box\, STOP = P$	
E5	$P \,\Box\, \mathbf{div} = \mathbf{div}$	
D1	$P \,\Box\, (Q \sqcap R) = (P \,\Box\, Q) \sqcap (P \,\Box\, R)$	
D2	$P \sqcap (Q \,\Box\, R) = (P \sqcap Q) \,\Box\, (P \sqcap R)$	
D3	$(a \to P) \,\Box\, (a \to Q) = a \to (P \sqcap Q)$	
D4	$(a \to P) \sqcap (a \to Q) = a \to (P \sqcap Q)$	
SC	$P \rhd Q = (P \,\Box\, Q) \sqcap Q$	
P0	$P \|\|_A (Q \|\|_A R) = (P \|\|_A Q) \|\|_A R$	
P1	$P \|\|_A Q = Q \|\|_A P$	
P2	$(P \sqcap Q) \|\|_A R = (P \|\|_A R) \sqcap (Q \|\|_A R)$	
P3	$P \|\|_A \mathbf{div} = \mathbf{div}$	
P4	If $P = \Box_{i \in I} (a_i \to P_i)$ and $Q = \Box_{j \in J} (b_j \to Q_j)$ then :	
	$P \| Q = \Box_{a_i \notin A} (a_i \to (P_i \|\|_A Q)) \Box$	
	$\Box_{a_j = b_j \in A} (a_i \to (P_i \|\|_A Q_j)) \Box$	
	$\Box_{b_j \notin A} (b_j \to (P \|\|_A Q_j))$	
H1	$(P \sqcap Q) \backslash A = (P \backslash A) \sqcap (Q \backslash A)$	
H2	$(P \,\Box\, a \to Q) \backslash A = ((P \,\Box\, Q) \backslash A) \sqcap (Q \backslash A)$	
H3	$\big(\Box_{i \in I} (b_i \to P_i)\big) \backslash A = \big(\Box_{i \in I} (b_i \to (P_i \backslash A))\big)$	if $\forall i \in I.\, b_i \notin A$
H4	$\mathbf{div} \backslash A = \mathbf{div}$	
R1	$f(P \sqcap Q) = f(P) \sqcap f(Q)$	
R2	$f(P \,\Box\, Q) = f(P) \,\Box\, f(Q)$	
R3	$f(a \to P) = f(a) \to f(P)$	
R4	$f(STOP) = STOP$	
R5	$f(\mathbf{div}) = \mathbf{div}$	
T1	$(P \sqcap Q) \,\Theta_A\, R = (P \,\Theta_A\, R) \sqcap (Q \,\Theta_A\, R)$	
T2	$(P \,\Box\, Q) \,\Theta_A\, R = (P \,\Theta_A\, R) \,\Box\, (Q \,\Theta_A\, R)$	
T3	$(a \to P) \,\Theta_A\, Q = a \to (P \,\Theta_A\, Q)$	if $a \notin A$
T4	$(a \to P) \,\Theta_A\, Q = a \to Q$	if $a \in A$
T5	$STOP \,\Theta_A\, Q = STOP$	
T6	$\mathbf{div} \,\Theta_A\, Q = \mathbf{div}$	
U1	$(P \sqcap Q) \,\triangle\, R = (P \,\triangle\, R) \sqcap (Q \,\triangle\, R)$	
U2	$(P \,\Box\, Q) \,\triangle\, R = (P \,\triangle\, R) \,\Box\, (Q \,\triangle\, R)$	
U3	$(a \to P) \,\triangle\, Q = (a \to (P \,\triangle\, Q)) \,\Box\, Q$	
U4	$STOP \,\triangle\, P = P$	
U5	$\mathbf{div} \,\triangle\, P = \mathbf{div}$	

$I = \{i_1, \ldots, i_n\}$ any finite index set, for $P_{i_1} \Box P_{i_2} \Box \ldots \Box P_{i_n}$, where $\Box_{i \in \emptyset} P_i$ represents *STOP*. This notation is justified by Axioms **E2**–**4**. Furthermore, $\bigsqcap_{j \in J} P_j$, with $J = \{j_1, .., j_m\}$ any finite, nonempty index set, denotes $P_{j_1} \sqcap P_{j_2} \sqcap \ldots \sqcap P_{j_m}$. This notation is justified by Axioms **I2** and **I3**. Now a *normal form* is a defined as a CSP expression of the form **div** or $\bigsqcap_{j \in J} R_j$, with $R_j = \big(\Box_{k \in K_j} (a_{kj} \to R_{kj})\big)$ for $j \in J$, where the subexpressions R_{kj} are again in normal form. Here J and the K_j are finite index sets, J nonempty.

Axioms $\perp$ and **I4** derive $P \sqcap \mathbf{div} = \mathbf{div}$. Together with Axioms **D1**, **SC**, **P1**–**4**, **H1**–**4**, **R1**–**5**, **T1**–**6** and **U1**–**5** this allows any recursion-free CSP expression to be rewritten into normal form. In [6] it is shown that for any two normal forms P and Q with $P \sqsubseteq_{FD} Q$, Axioms $\perp$, **I1**–**4**, **E1**–**5** and **D1**–**4** derive $\vdash P = Q$. Together, this yields the completeness of the axiomatisation of Table 2.

5 Other Models of CSP

Several alternative models of CSP have been proposed in the literature, including the readiness-divergences model of Olderog & Hoare [13] and the stable revivals model of Roscoe [16]. A hierarchy of such models is surveyed in Roscoe [17]. Each of these models corresponds with a preorder (and associated semantic equivalence) on labelled transition systems. In [7] I presented a survey of semantic equivalences and preorders on labelled transition systems, ordered by inclusion in a lattice. Each model occurring in [17] correspond exactly with with one of the equivalences of [7], or—like the stable revivals model—arises as the meet or join of two such equivalences.

In the other direction, not every semantic equivalence or preorder from [7] yields a sensible model of CSP. First of all, one would want to ensure that it is a (pre)congruence for the operators of CSP. Additionally, one might impose sanity requirements on the treatment of recursion.

The hierarchy of models in [17] roughly consist of two hierarchies: the stable models, and the divergence-strict ones. The failures-divergences model could be seen as the centre piece in the divergence-strict hierarchy, and the stable failures model [15], which outside CSP stems from Bergstra, Klop & Olderog [2], plays the same role in the stable hierarchy. Each of these hierarchies has a maximal (least discriminating) element, called $\mathcal{FL}^{\Downarrow}$ and $\mathcal{FL}$ in [17]. These correspond to the ready trace models $RT^{\downarrow}$ and RT of [7].

The goal of the present paper is to propose a sensible model of CSP that is strictly finer than all models thus far considered, and thus unites the two hierarchies mentioned above. As all models of CSP considered so far have a distinctly linear time flavour, I here propose a branching time model, thereby showing that the syntax of CSP is not predisposed towards linear time models. My model can be given as an equivalence relation on labelled transition system, provided I show that it is a congruence for the operators of CSP. I aim for an equivalence that allows a complete axiomatisation in the style of Table 2, obtained by replacing axioms that are no longer valid by weaker ones.

One choice could be to base a model on strong bisimulation equivalence [12]. Strong bisimilarity is a congruence for all CSP operators, because their operational semantics fits the tyft/tyxt format of [9]. However, this is an unsuitable equivalence for CSP, because it fails to abstract from internal actions. Even the axiom **I1** would not be valid, as the two sides differ by an internal action.

A second proposal could be based on weak bisimilarity [12]. This equivalence abstracts from internal activity, and validates **I1**. The default incarnation of weak bisimilarity is not finer than failures-divergences equivalence, because it satisfies $\mathbf{div} = STOP$. Therefore, one would take a divergence-preserving variant of this notion: the *weak bisimulation with explicit divergence* of Bergstra, Klop & Olderog [2]. Yet, some crucial CSP laws are invalidated, such as **I3** and **D1**. This destroys any hope of a complete axiomatisation along the lines of Table 2.

My final choice is *divergence-preserving coupled similarity* [7], based on coupled similarity for divergence-free processes from Parrow & Sjödin [14]. This is the finest equivalence in [7] that satisfies **I3** and **D1**. In fact, it satisfies all of the axioms of Table 2, except for the ones marked red: $\bot$, **I4**, **E1**, **E5**, **D2**–**4**, **SC**, **P2**, **P3**, **H2**, **U2**, **U3** and **U5**.

Divergence-preserving coupled similarity belongs to the bisimulation family of semantic equivalences, in the sense that on transition systems without internal actions it coincides with strong bisimilarity.

In Sect. 6 I present divergence-preserving coupled similarity. In Sect. 7 I prove that it is a congruence for the operators of CSP, and in Sect. 8 I present a complete axiomatisation for recursion-free CSP processes without interrupts.

6 Divergence-Preserving Coupled Similarity

Definition 4. A *coupled simulation* is a binary relation $\mathscr{R}$ on CSP processes, such that, for all $\alpha \in \Sigma \cup \{\tau\}$,

- if $P \mathscr{R} Q$ and $P \xrightarrow{\alpha} P'$ then there exists a Q' with $Q \stackrel{\hat{\alpha}}{\Longrightarrow} Q'$ and $P' \mathscr{R} Q'$,
- and if $P \mathscr{R} Q$ then there exists a Q' with $Q \Longrightarrow Q'$ and $Q' \mathscr{R} P$.

It is *divergence-preserving* if $P \mathscr{R} Q$ and $P\Uparrow$ implies $Q\Uparrow$. Write $P \sqsupseteq^{\Delta}_{CS} Q$ if there exists a divergence-preserving coupled simulation $\mathscr{R}$ with $P \mathscr{R} Q$. Two processes P and Q are *divergence-preserving coupled similar*, notation $P \equiv^{\Delta}_{CS} Q$, if $P \sqsupseteq^{\Delta}_{CS} Q$ and $Q \sqsupseteq^{\Delta}_{CS} P$.

Note that the union of any collection of divergence-preserving coupled simulations is itself a divergence-preserving coupled simulation. In particular, $\sqsupseteq^{\Delta}_{CS}$ is a divergence-preserving coupled simulation. Also note that in the absence of the internal action τ, coupled simulations are symmetric, and coupled similarity coincides with strong bisimilarity (as defined in [12]).

Intuitively, $P \sqsupseteq^{\Delta}_{CS} Q$ says that P is "ahead" of a state matching Q, where P' is ahead of P if $P \Longrightarrow P'$. The first clause says that if P is ahead of a state matching Q, then any transition performed by P can be matched by Q—possibly after Q "caught up" with P by performing some internal transitions.

The second clause says that if P is ahead of Q, then Q can always catch up, so that it is ahead of P. Thus, if P and Q are in stable states—where no internal actions are possible—then $P \sqsubseteq^{\Delta}_{CS} Q$ implies $Q \sqsubseteq^{\Delta}_{CS} P$. In all other situations, P and Q do not need to be matched exactly, but there do exists under- and overapproximations of a match. The result is that the relation behaves like a weak bisimulation w.r.t. visible actions, but is not so pedantic in matching internal actions.

Proposition 1. *$\sqsupseteq^{\Delta}_{CS}$ is reflexive and transitive, and thus a preorder.*

Proof. The identity relation Id is a divergence-preserving coupled simulation, and if $\mathscr{R}$, $\mathscr{R}'$ are divergence-preserving coupled simulations, then so is $\mathscr{R}$; $\mathscr{R}' \cup \mathscr{R}'$; $\mathscr{R}$. Here $\mathscr{R}$; $\mathscr{R}'$ is defined by P $\mathscr{R}$; $\mathscr{R}'$ R iff there is a Q with $P \mathscr{R} Q \mathscr{R}' R$.

$\mathscr{R}$; $\mathscr{R}'$ is divergence-preserving: if $P \mathscr{R} Q \mathscr{R}' R$ and $P{\Uparrow}$, then $Q{\Uparrow}$, and thus $R{\Uparrow}$. The same holds for $\mathscr{R}'$; $\mathscr{R}$, and thus for $\mathscr{R}$; $\mathscr{R}' \cup \mathscr{R}'$; $\mathscr{R}$.

To check that $\mathscr{R}$; $\mathscr{R}' \cup \mathscr{R}'$; $\mathscr{R}$ satisfies the first clause of Definition 4, note that if $Q \mathscr{R}' R$ and $Q \stackrel{\hat{\alpha}}{\Longrightarrow} Q'$, then, by repeated application of the first clause of Definition 4, there is an R' with $R \stackrel{\hat{\alpha}}{\Longrightarrow} R'$ and $Q' \mathscr{R}' R'$.

Towards the second clause, if $P \mathscr{R} Q \mathscr{R}' R$, then, using the second clause for $\mathscr{R}$, there is a Q' with $Q \Longrightarrow Q'$ and $Q' \mathscr{R} P$. Hence, using the first clause for $\mathscr{R}'$, there is an R' with $R \Longrightarrow R'$ and $Q' \mathscr{R}' R'$. Thus, using the second clause for $\mathscr{R}'$, there is an R'' with $R' \Longrightarrow R''$ and $R'' \mathscr{R}' Q'$, and hence R'' $\mathscr{R}'$; $\mathscr{R}$ P'. □

Proposition 2. *If $P \Longrightarrow Q$ then $P \sqsubseteq^{\Delta}_{CS} Q$.*

Proof. I show that $Id \cup \{(Q, P)\}$, with Id the identity relation, is a coupled simulation. Namely if $Q \stackrel{\alpha}{\longrightarrow} Q'$ then surely $P \stackrel{\alpha}{\Longrightarrow} Q'$. The second clause of Definition 4 is satisfied because $P \Longrightarrow Q$. Furthermore, if $Q{\Uparrow}$ then certainly $P{\Uparrow}$, so the relation is divergence-preserving. □

Proposition 3. *$P \sqsupseteq^{\Delta}_{CS} Q$ iff $P \sqcap Q \equiv^{\Delta}_{CS} Q$.*

Proof. "⇒": Let $\mathscr{R}$ be the smallest relation such that, for any P and Q, $P \sqsupseteq^{\Delta}_{CS} Q$ implies $P \mathscr{R} Q$, $(P \sqcap Q) \mathscr{R} Q$ and $Q \mathscr{R} (P \sqcap Q)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

That $\mathscr{R}$ is divergence-preserving is trivial, using that $(P \sqcap Q){\Uparrow}$ iff $P{\Uparrow} \vee Q{\Uparrow}$.

Suppose $P^* \mathscr{R} Q$ and $P^* \stackrel{\alpha}{\longrightarrow} P'$. The case that $P^* = P$ with $P \sqsupseteq^{\Delta}_{CS} Q$ is trivial. Now let Q be $Q^* \sqcap P^*$. Since $P^* \stackrel{\alpha}{\longrightarrow} P'$, surely $Q \stackrel{\alpha}{\Longrightarrow} P'$, and $P' \mathscr{R} P'$. Finally, let $P^* = (P \sqcap Q)$ with $P \sqsupseteq^{\Delta}_{CS} Q$. Then $\alpha = \tau$ and P' is either P or Q. Both cases are trivial, taking $Q' = Q$.

Towards the second clause of Definition 4, suppose $P^* \mathscr{R} Q$. The case $P^* = P$ with $P \sqsupseteq^{\Delta}_{CS} Q$ is trivial. Now let Q be $Q^* \sqcap P^*$. Then $Q \Longrightarrow P^*$ and $P^* \mathscr{R} P^*$. Finally, let $P^* = (P \sqcap Q)$ with $P \sqsupseteq^{\Delta}_{CS} Q$. Then $Q \Longrightarrow Q$ and $Q \mathscr{R} (P \sqcap Q)$.

"⇐": Suppose $P \sqcap Q \sqsupseteq^{\Delta}_{CS} Q$. Since $P \sqcap Q \stackrel{\tau}{\longrightarrow} P$ there exists a Q' with $Q \Longrightarrow Q'$ and $P \sqsupseteq^{\Delta}_{CS} Q'$. By Proposition 2 $Q' \sqsupseteq^{\Delta}_{CS} Q$ and by Proposition 1 $P \sqsupseteq^{\Delta}_{CS} Q$. □

7 Congruence Properties

Proposition 4. $\equiv^{\Delta}_{CS}$ *is a congruence for action prefixing.*

Proof. I have to show that $P \equiv^{\Delta}_{CS} Q$ implies $(a \to P) \equiv^{\Delta}_{CS} (a \to Q)$.

Let $\mathscr{R}$ be the smallest relation such that, for any P and Q, $P \sqsubseteq^{\Delta}_{CS} Q$ implies $P \mathscr{R} Q$, and $P \equiv^{\Delta}_{CS} Q$ implies $(a \to P) \mathscr{R} (a \to Q)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

Checking the conditions of Definition 4 for the case $P \mathscr{R} Q$ with $P \sqsubseteq^{\Delta}_{CS} Q$ is trivial. So I examine the case $(a \to P) \mathscr{R} (a \to Q)$ with $P \equiv^{\Delta}_{CS} Q$.

Suppose $(a \to P) \xrightarrow{\alpha} P'$. Then $\alpha = a$ and $P' = P$. Now $(a \to Q) \xrightarrow{\alpha} Q$ and $P \mathscr{R} Q$, so the first condition of Definition 4 is satisfied.

For the second condition, $(a \to Q) \Longrightarrow (a \to Q)$, and, since $Q \equiv^{\Delta}_{CS} P$, $(a \to Q) \mathscr{R} (a \to P)$. Thus, $\mathscr{R}$ is a coupled simulation.

As $a \to P$ does not diverge, $\mathscr{R}$ moreover is divergence-preserving. □

Since $STOP \sqsupseteq^{\Delta}_{CS} (a \to STOP) \rhd STOP$ but $STOP \not\sqsupseteq^{\Delta}_{CS} (a \to STOP) \rhd STOP$, and thus $b \to STOP \not\sqsupseteq^{\Delta}_{CS} b \to \big((a \to STOP) \rhd STOP\big)$, the relation $\sqsupseteq^{\Delta}_{CS}$ is *not* a precongruence for action prefixing.

It is possible to express action prefixing in terms of the throw operator: $a \to P$ is strongly bisimilar with $(a \to STOP) \Theta_{\{a\}} P$. Consequently, $\sqsupseteq^{\Delta}_{CS}$ is not a precongruence for the throw operator.

Proposition 5. $\equiv^{\Delta}_{CS}$ *is a congruence for the throw operator.*

Proof. Let $A \subseteq \Sigma$. Let $\mathscr{R}$ be the smallest relation such that, for any P_1, P_2, Q_1, Q_2, $P_1 \sqsupseteq^{\Delta}_{CS} Q_1$ and $P_2 \equiv^{\Delta}_{CS} Q_2$ implies $P_1 \mathscr{R} Q_1$ and $(P_1 \Theta_A P_2) \mathscr{R} (Q_1 \Theta_A Q_2)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

So let $P_1 \sqsupseteq^{\Delta}_{CS} Q_1$, $P_2 \equiv^{\Delta}_{CS} Q_2$ and $(P_1 \Theta_A P_2) \xrightarrow{\alpha} P'$. Then $P_1 \xrightarrow{\alpha} P'_1$ for some P'_2, and either $\alpha \notin A$ and $P' = P'_1 \Theta_A P_2$, or $\alpha \in A$ and $P' = P_2$. So there is a Q'_1 with $Q_1 \stackrel{\hat{\alpha}}{\Longrightarrow} Q'_1$ and $P'_1 \sqsupseteq^{\Delta}_{CS} Q'_1$. If $\alpha \notin A$ it follows that $(Q_1 \Theta_A Q_2) \stackrel{\hat{\alpha}}{\Longrightarrow} (Q'_1 \Theta_A Q_2)$ and $(P'_1 \Theta_A P_2) \mathscr{R} (Q'_1 \Theta_A Q_2)$. If $\alpha \in A$ it follows that $(Q_1 \Theta_A Q_2) \stackrel{\alpha}{\Longrightarrow} Q_2$ and $P_2 \mathscr{R} Q_2$.

Now let $P_1 \sqsupseteq^{\Delta}_{CS} Q_1$ and $P_2 \equiv^{\Delta}_{CS} Q_2$. Then there is a Q'_1 with $Q_1 \Longrightarrow Q'_1$ and $Q'_1 \sqsupseteq^{\Delta}_{CS} P_1$. Hence $Q_1 \Theta_A Q_2 \Longrightarrow Q'_1 \Theta_A Q_2$ and $(Q'_1 \Theta_A Q_2) \mathscr{R} (P_1 \Theta_A P_2)$.

The same two conditions for the case $P \mathscr{R} Q$ because $P \sqsupseteq^{\Delta}_{CS} Q$ are trivial. Thus $\mathscr{R}$ is a coupled simulation. That $\mathscr{R}$ is divergence-preserving follows because $P_1 \Theta_A P_2 \Uparrow$ iff $P_1 \Uparrow$. □

I proceed to show that $\sqsupseteq^{\Delta}_{CS}$ is a precongruence for all the other operators of CSP. This implies that $\equiv^{\Delta}_{CS}$ is a congruence for all the operators of CSP.

Proposition 6. $\sqsupseteq^{\Delta}_{CS}$ *is a precongruence for internal choice.*

Proof. Let $\mathscr{R}$ be the smallest relation such that, for any P_i and Q_i, $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ implies $P_i \mathscr{R} Q_i$ $(i = 1, 2)$ and $(P_1 \sqcap P_2) \mathscr{R} (Q_1 \sqcap Q_2)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

So let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ and $(P_1 \sqcap P_2) \xrightarrow{\alpha} P'$. Then $\alpha = \tau$ and $P' = P_i$ for $i = 1$ or 2. Now $Q_1 \sqcap Q_2 \Longrightarrow Q_i$ and $P_i \mathscr{R} Q_i$.

Now let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$. Then there is a Q_1' with $Q_1 \Longrightarrow Q_1'$ and $Q_1' \sqsupseteq^{\Delta}_{CS} P_1$. By Proposition 2 $P_1 \sqsupseteq^{\Delta}_{CS} P_1 \sqcap P_2$ and by Proposition 1 $Q_1' \sqsupseteq^{\Delta}_{CS} P_1 \sqcap P_2$.

The same two conditions for the case $P \mathscr{R} Q$ because $P \sqsupseteq^{\Delta}_{CS} Q$ are trivial. Thus $\mathscr{R}$ is a coupled simulation. That $\mathscr{R}$ is divergence-preserving follows because $P_1 \sqcap P_2 \Uparrow$ iff $P_1 \Uparrow \vee P_2 \Uparrow$. □

Proposition 7. *$\sqsupseteq^{\Delta}_{CS}$ is a precongruence for external choice.*

Proof. Let $\mathscr{R}$ be the smallest relation such that, for any P_i and Q_i, $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ implies $P_i \mathscr{R} Q_i$ $(i = 1, 2)$ and $(P_1 \,\Box\, P_2) \mathscr{R} (Q_1 \,\Box\, Q_2)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

So let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ and $(P_1 \,\Box\, P_2) \xrightarrow{\alpha} P'$. If $\alpha \in \Sigma$ then $P_i \xrightarrow{\alpha} P'$ for $i = 1$ or 2, and there exists a Q' with $Q_i \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \sqsupseteq^{\Delta}_{CS} Q'$. Hence $Q_1 \,\Box\, Q_2 \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \mathscr{R} Q'$. If $\alpha = \tau$ then either $P_1 \xrightarrow{\tau} P_1'$ for some P_1' with $P' = P_1' \,\Box\, P_2$, or $P_2 \xrightarrow{\tau} P_2'$ for some P_2' with $P' = P_1 \,\Box\, P_2'$. I pursue only the first case, as the other follows by symmetry. Here $Q_1 \Longrightarrow Q_1'$ for some Q_1' with $P_1' \sqsupseteq^{\Delta}_{CS} Q_1'$. Thus $Q_1 \,\Box\, Q_2 \Longrightarrow Q_1' \,\Box\, Q_2$ and $(P_1' \,\Box\, P_2) \mathscr{R} (Q_1' \,\Box\, Q_2)$.

Now let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$. Then, for $i = 1, 2$, there is a Q_i' with $Q_i \Longrightarrow Q_i'$ and $Q_i' \sqsupseteq^{\Delta}_{CS} P_i$. Hence $Q_1 \,\Box\, Q_2 \Longrightarrow Q_1' \,\Box\, Q_2'$ and $(Q_1' \,\Box\, Q_2') \mathscr{R} (P_1 \,\Box\, P_2)$.

Thus $\mathscr{R}$ is a coupled simulation. That $\mathscr{R}$ is divergence-preserving follows because $P_1 \,\Box\, P_2 \Uparrow$ iff $P_1 \Uparrow \vee P_2 \Uparrow$. □

Proposition 8. *$\sqsupseteq^{\Delta}_{CS}$ is a precongruence for sliding choice.*

Proof. Let $\mathscr{R}$ be the smallest relation such that, for any P_i and Q_i, $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ implies $P_i \mathscr{R} Q_i$ $(i = 1, 2)$ and $(P_1 \rhd P_2) \mathscr{R} (Q_1 \rhd Q_2)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

So let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ and $(P_1 \rhd P_2) \xrightarrow{\alpha} P'$. If $\alpha \in \Sigma$ then $P_1 \xrightarrow{\alpha} P'$, and there exists a Q' with $Q_1 \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \sqsupseteq^{\Delta}_{CS} Q'$. Hence $Q_1 \rhd Q_2 \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \mathscr{R} Q'$. If $\alpha = \tau$ then either $P' = P_2$ or $P_1 \xrightarrow{\tau} P_1'$ for some P_1' with $P' = P_1' \rhd P_2$. In the former case $Q_1 \rhd Q_2 \Longrightarrow Q_2$ and $P_2 \mathscr{R} Q_2$. In the latter case $Q_1 \Longrightarrow Q_1'$ for some Q_1' with $P_1' \sqsupseteq^{\Delta}_{CS} Q_1'$. Thus $Q_1 \rhd Q_2 \Longrightarrow Q_1' \rhd Q_2$ and $(P_1' \rhd P_2) \mathscr{R} (Q_1' \rhd Q_2)$.

Now let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$. Then there is a Q_2' with $Q_2 \Longrightarrow Q_2'$ and $Q_2' \sqsupseteq^{\Delta}_{CS} P_2$. By Proposition 2 $P_2 \sqsupseteq^{\Delta}_{CS} P_1 \rhd P_2$ and by Proposition 1 $Q_2' \sqsupseteq^{\Delta}_{CS} P_1 \rhd P_2$.

Thus $\mathscr{R}$ is a coupled simulation. That $\mathscr{R}$ is divergence-preserving follows because $P_1 \rhd P_2 \Uparrow$ iff $P_1 \Uparrow \vee P_2 \Uparrow$. □

Proposition 9. *$\sqsupseteq^{\Delta}_{CS}$ is a precongruence for parallel composition.*

Proof. Let $A \subseteq \Sigma$. Let $\mathscr{R}$ be the smallest relation such that, for any P_i and Q_i, $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ implies $(P_1 \|_A P_2) \mathscr{R} (Q_1 \|_A Q_2)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

So let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ and $(P_1 \|_A P_2) \xrightarrow{\alpha} P'$. If $\alpha \notin A$ then $P_i \xrightarrow{\alpha} P_i'$ for $i = 1$ or 2, and $P' = P_1' \|_A P_2'$, where $P'_{3-i} := P_{3-i}$. Hence there exists a Q_i'

with $Q_i \stackrel{\hat{\alpha}}{\Longrightarrow} Q_i'$ and $P_i' \sqsupseteq^{\Delta}_{CS} Q_i'$. Let $Q'_{3-i} := Q_{3-i}$. Then $Q_1 \|_A Q_2 \stackrel{\hat{\alpha}}{\Longrightarrow} Q_1' \| Q_2'$ and $(P_1' \| P_2') \mathscr{R} (Q_1' \| Q_2')$. If $\alpha \in A$ then $P_i \stackrel{\alpha}{\longrightarrow} P_i'$ for $i = 1$ and 2. Hence, for $i = 1, 2$, $Q_i \stackrel{\alpha}{\Longrightarrow} Q_i'$ for some Q_i' with $P_i' \sqsupseteq^{\Delta}_{CS} Q_i'$. Thus $Q_1 \|_A Q_2 \stackrel{\alpha}{\Longrightarrow} Q_1' \|_A Q_2'$ and $(P_1' \|_A P_2') \mathscr{R} (Q_1' \|_A Q_2')$.

Now let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$. Then, for $i = 1, 2$, there is a Q_i' with $Q_i \Longrightarrow Q_i'$ and $Q_i' \sqsupseteq^{\Delta}_{CS} P_i$. Hence $Q_1 \|_A Q_2 \Longrightarrow Q_1' \|_A Q_2'$ and $(Q_1' \|_A Q_2') \mathscr{R} (P_1 \|_A P_2)$.

Thus $\mathscr{R}$ is a coupled simulation. That $\mathscr{R}$ is divergence-preserving follows because $P_1 \|_A P_2 \Uparrow$ iff $P_1 \Uparrow \vee P_2 \Uparrow$. □

Proposition 10. *$\sqsupseteq^{\Delta}_{CS}$ is a precongruence for concealment.*

Proof. Let $A \subseteq \Sigma$. Let $\mathscr{R}$ be the smallest relation such that, for any P and Q, $P \sqsubseteq^{\Delta}_{CS} Q$ implies $(P \backslash A) \mathscr{R} (Q \backslash A)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

So let $P \sqsubseteq^{\Delta}_{CS} Q$ and $P \backslash A \stackrel{\alpha}{\longrightarrow} P^*$. Then $P^* = P' \backslash A$ for some P' with $P \stackrel{\beta}{\longrightarrow} P'$, and either $\beta \in A$ and $\alpha = \tau$, or $\beta = \alpha \notin A$. Hence $Q \stackrel{\beta}{\longrightarrow} Q'$ for some Q' with $P' \sqsubseteq^{\Delta}_{CS} Q'$. Therefore $Q \backslash A \stackrel{\alpha}{\longrightarrow} Q' \backslash A$ and $(P' \backslash A) \mathscr{R} (Q' \backslash A)$.

Now let $P \sqsubseteq^{\Delta}_{CS} Q$. Then there is a Q' with $Q \Longrightarrow Q'$ and $Q' \sqsupseteq^{\Delta}_{CS} P$. Hence $Q \backslash A \Longrightarrow Q' \backslash A$ and $(Q' \backslash A) \mathscr{R} (P \backslash A)$.

To check that $\mathscr{R}$ is divergence-preserving, suppose $(P \backslash A) \Uparrow$. Then there are P_i and $\alpha_i \in A \cup \{\tau\}$ for all $i > 0$ such that $P \stackrel{\alpha_1}{\longrightarrow} P_1 \stackrel{\alpha_2}{\longrightarrow} P_2 \stackrel{\alpha_3}{\longrightarrow} \ldots$. By the first condition of Definition 4, there are Q_i for all $i > 0$ such that $P_i \mathscr{R} Q_i$ and $Q \stackrel{\hat{\alpha}_1}{\Longrightarrow} Q_1 \stackrel{\hat{\alpha}_2}{\Longrightarrow} Q_2 \stackrel{\hat{\alpha}_3}{\Longrightarrow} \ldots$. This implies $Q \backslash A \Longrightarrow Q_1 \backslash A \Longrightarrow Q_2 \backslash A \Longrightarrow \ldots$.

In case $\alpha_i \in \Sigma$ for infinitely many i, then for infinitely many i one has $Q_{i-1} \stackrel{\alpha_i}{\Longrightarrow} Q_i$ and thus $Q_{i-1} \backslash A \stackrel{\tau}{\Longrightarrow} Q_i \backslash A$. This implies that $(Q \backslash A) \Uparrow$.

Otherwise there is an $n > 0$ such that $\alpha_i = \tau$ for all $i \geq n$. In that case $P_n \Uparrow$ and thus $Q_n \Uparrow$. Hence $(Q_n \backslash A) \Uparrow$ and thus $(Q \backslash A) \Uparrow$. □

Proposition 11. *$\sqsupseteq^{\Delta}_{CS}$ is a precongruence for renaming.*

Proof. Let $f : \Sigma \to \Sigma$. Let $\mathscr{R}$ be the smallest relation such that, for any P and Q, $P \sqsubseteq^{\Delta}_{CS} Q$ implies $f(P) \mathscr{R} f(Q)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

So let $P \sqsubseteq^{\Delta}_{CS} Q$ and $f(P) \stackrel{\alpha}{\longrightarrow} P^*$. Then $P^* = f(P')$ for some P' with $P \stackrel{\beta}{\longrightarrow} P'$ and $f(\beta) = \alpha$. Hence $Q \stackrel{\beta}{\longrightarrow} Q'$ for some Q' with $P' \sqsubseteq^{\Delta}_{CS} Q'$. Therefore $f(Q) \stackrel{\alpha}{\longrightarrow} f(Q')$ and $f(P') \mathscr{R} f(Q')$.

Now let $P \sqsubseteq^{\Delta}_{CS} Q$. Then there is a Q' with $Q \Longrightarrow Q'$ and $Q' \sqsupseteq^{\Delta}_{CS} P$. Hence $f(Q) \Longrightarrow f(Q')$ and $f(Q') \mathscr{R} f(P)$.

To check that $\mathscr{R}$ is divergence-preserving, suppose $f(P) \Uparrow$. Then $P \Uparrow$, so $Q \Uparrow$ and $f(Q) \Uparrow$. □

Proposition 12. *$\sqsupseteq^{\Delta}_{CS}$ is a precongruence for the interrupt operator.*

Proof. Let $\mathscr{R}$ be the smallest relation such that, for any P_i and Q_i, $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ implies $P_2 \mathscr{R} Q_2$ and $(P_1 \triangle P_2) \mathscr{R} (Q_1 \triangle Q_2)$. It suffices to show that $\mathscr{R}$ is a divergence-preserving coupled simulation.

So let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$ and $(P_1 \mathbin{\triangle} P_2) \xrightarrow{\alpha} P'$. Then either $P' = P'_1 \mathbin{\triangle} P_2$ for some P'_1 with $P_1 \xrightarrow{\alpha} P'_1$, or $\alpha = \tau$ and $P' = P_1 \mathbin{\triangle} P'_2$ for some P'_2 with $P_2 \xrightarrow{\tau} P'_2$, or $\alpha \in \Sigma$ and $P_2 \xrightarrow{\alpha} P'$.

In the first case there is a Q'_1 with $Q_1 \stackrel{\hat{\alpha}}{\Longrightarrow} Q'_1$ and $P'_1 \sqsupseteq^{\Delta}_{CS} Q'_1$. It follows that $(Q_1 \mathbin{\triangle} Q_2) \stackrel{\hat{\alpha}}{\Longrightarrow} (Q'_1 \mathbin{\triangle} Q_2)$ and $(P'_1 \mathbin{\triangle} P_2) \mathscr{R} (Q'_1 \mathbin{\triangle} Q_2)$.

In the second case there is a Q'_2 with $Q_2 \Longrightarrow Q'_2$ and $P'_2 \sqsupseteq^{\Delta}_{CS} Q'_2$. It follows that $(Q_1 \mathbin{\triangle} Q_2) \Longrightarrow (Q_1 \mathbin{\triangle} Q'_2)$ and $(P_1 \mathbin{\triangle} P'_2) \mathscr{R} (Q_1 \mathbin{\triangle} Q'_2)$.

In the last case there is a Q'_2 with $Q_2 \stackrel{\alpha}{\Longrightarrow} Q'_2$ and $P'_2 \sqsupseteq^{\Delta}_{CS} Q'_2$. It follows that $(Q_1 \mathbin{\triangle} Q_2) \stackrel{\alpha}{\Longrightarrow} Q'_2$ and $P'_2 \mathscr{R} Q'_2$.

Now let $P_i \sqsupseteq^{\Delta}_{CS} Q_i$ for $i = 1, 2$. Then, for $i = 1, 2$, there is a Q'_i with $Q_i \Longrightarrow Q'_i$ and $Q'_i \sqsupseteq^{\Delta}_{CS} P_i$. Hence $Q_1 \mathbin{\triangle} Q_2 \Longrightarrow Q'_1 \mathbin{\triangle} Q'_2$ and $(Q'_1 \mathbin{\triangle} Q'_2) \mathscr{R} (P_1 \mathbin{\triangle} P_2)$.

Thus $\mathscr{R}$ is a coupled simulation. That $\mathscr{R}$ is divergence-preserving follows because $P_1 \mathbin{\triangle} P_2{\Uparrow}$ iff $P_1{\Uparrow} \vee P_2{\Uparrow}$. □

8 A Complete Axiomatisation of $\equiv^{\Delta}_{CS}$

A set of equational laws valid for $\equiv^{\Delta}_{CS}$ is presented in Table 3. It includes the laws from Table 2 that are still valid for $\equiv^{\Delta}_{CS}$. I will show that this axiomatisation is sound and complete for $\equiv^{\Delta}_{CS}$ for recursion-free CSP without the interrupt operator. The axioms **U2** and **U3**, which are not valid for $\equiv^{\Delta}_{CS}$, played a crucial rôle in reducing CSP expressions with interrupt into normal form. It is not trivial to find valid replacements, and due to lack of space and time I do not tackle this problem here.

The axiom **H5** replaces the fallen axiom **H2**, and is due to [17]. Here the result of hiding actions results in a process that cannot be expressed as a normal form built up from $a \rightarrow$, $\sqcap$ and $\Box$. For this reason, one needs a richer normal form, involving the sliding choice operator. It is given by the context-free grammar

$$\begin{aligned} N &\to D \mid D \rhd I \\ I &\to D \mid I \sqcap I \\ D &\to \mathit{STOP} \mid \mathbf{div} \mid E \mid \mathbf{div} \,\Box\, E \\ E &\to (a \to N) \mid (a \to N) \,\Box\, E \,. \end{aligned}$$

Definition 5. A CSP expression is in *head normal form* if it is of the form $\big([\mathbf{div}\,\Box]\,\Box_{i\in I}(a_i \to R_i)\big) \rhd \bigsqcap_{j\in J} R_j$, with $R_j = \big([\mathbf{div}\,\Box]\,\Box_{k\in K_j}(a_{kj} \to R_{kj})\big)$ for $j \in J$. Here I, J and the K_j are finite index sets, and the parts between square brackets are optional. Here, although $\bigsqcap_{i\in\emptyset} P_i$ is undefined, I use $P \rhd \bigsqcap_{i\in\emptyset} P_i$ to represent P. An expression is in *normal form* if it has this form and also the subexpressions R_i and R_{kj} are in normal form.

A head normal form is *saturated* if the **div**-summand on the left is present whenever any of the R_j has a **div**-summand, and for any $j \in J$ and any $k \in K_j$ there is an $i \in I$ with $a_i = a_{kj}$ and $R_i = R_{kj}$.

My proof strategy is to ensure that there are enough axioms to transform any CSP process without recursion and interrupt operators into normal form, and

to make these forms saturated; then to equate saturated normal forms that are divergence-preserving coupled simulation equivalent.

Due to the optional presence in head normal forms of a **div**-summand and a sliding choice, I need four variants of the axiom **H5**; so far I have not seen a way around this. Likewise, there are 4×4 variants of the axiom **P4** from Table 2, of which 6 could be suppressed by symmetry (**P4**–**P13**). There are also 3 axioms replacing **P2** (**P14**–**P16**).

9 Soundness

Since divergence-preserving coupled similarity is a congruence for all CSP operators, to establish the soundness of the axiomatisation of Table 3 it suffices to show the validity w.r.t. $\equiv^{\Delta}_{CS}$ of all axioms. When possible, I show validity w.r.t. strong bisimilarity, which is a strictly finer equivalence.

Definition 6. Two processes are *strongly bisimilar* [12] if they are related by a binary relation $\mathscr{R}$ on processes such that, for all $\alpha \in \Sigma \cup \{\tau\}$,

– if $P \mathscr{R} Q$ and $P \xrightarrow{\alpha} P'$ then there exists a Q' with $Q \xrightarrow{\alpha} Q'$ and $P' \mathscr{R} Q'$,
– if $P \mathscr{R} Q$ and $Q \xrightarrow{\alpha} Q'$ then there exists a P' with $P \xrightarrow{\alpha} P'$ and $P' \mathscr{R} Q'$.

Proposition 13. *Axiom* **I1** *is valid for* $\equiv^{\Delta}_{CS}$.

Proof. $\{(P \sqcap P, P), (P, P \sqcap P) \mid P \text{a process}\} \cup Id$ is a divergence-preserving coupled simulation. □

Proposition 14. *Axiom* **I2** *is valid even for strong bisimilarity.*

Proof. $\{(P \sqcap Q, Q \sqcap P) \mid P, Q \text{ processes}\} \cup Id$ is a strong bisimulation. □

Proposition 15. *Axiom* **I3** *is valid for* $\equiv^{\Delta}_{CS}$.

Proof. The relation $\{\big(P \sqcap (Q \sqcap R), (P \sqcap Q) \sqcap R\big), \big((P \sqcap Q) \sqcap R, P \sqcap (Q \sqcap R)\big), \big(Q \sqcap R, (P \sqcap Q) \sqcap R\big), \big(P \sqcap Q, P \sqcap (Q \sqcap R)\big), \big(R, Q \sqcap R\big), \big(P, P \sqcap Q\big) \mid P, Q, R \text{ processes}\} \cup Id$ is a divergence-preserving coupled simulation. □

Proposition 16. *Axioms* **E2**–**4** *are valid for strong bisimilarity.*

Proof. The relation $\{\big(P \,\Box\, (Q \,\Box\, R), (P \,\Box\, Q) \,\Box\, R\big) \mid P, Q, R \text{ processes}\} \cup Id$ is a strong bisimulation. So is $\{(P \,\Box\, Q, Q \,\Box\, P) \mid P, Q \text{ processes}\} \cup Id$, as well as $\{(P \,\Box\, STOP, P) \mid P \text{ a process}\} \cup Id$. □

Proposition 17. *Axiom* **S1** *is valid for* $\equiv^{\Delta}_{CS}$.

Proof. $\{(P' \rhd P, P), (P, P' \rhd P) \mid P' \sqsupseteq^{\Delta}_{CS} P\} \cup Id$ is a divergence-preserving coupled simulation. This follows from Proposition 2. □

Proposition 18. *Axiom* **S2** *is valid for* $\equiv^{\Delta}_{CS}$.

Table 3. A complete axiomatisation of $\equiv^{\mathcal{A}}_{CS}$ for recursion-free CSP without interrupt

I1	$P \sqcap P = P$
I2	$P \sqcap Q = Q \sqcap P$
I3	$P \sqcap (Q \sqcap R) = (P \sqcap Q) \sqcap R$
E2	$P \mathbin{\Box} Q = Q \mathbin{\Box} P$
E3	$P \mathbin{\Box} (Q \mathbin{\Box} R) = (P \mathbin{\Box} Q) \mathbin{\Box} R$
E4	$P \mathbin{\Box} STOP = P$
S1	$P \rhd P = P$
S2	$P \rhd (Q \rhd R) = (P \rhd Q) \rhd R$
S3	$(P \rhd Q) \rhd R = (P \mathbin{\Box} Q) \rhd R$
S4	$(P \sqcap Q) \rhd R = (P \mathbin{\Box} Q) \rhd R$
S5	$STOP \rhd P = P$
S6	$(P \rhd Q) \sqcap (R \rhd S) = (P \mathbin{\Box} R) \rhd (Q \sqcap S)$
S7	$(P \rhd Q) \mathbin{\Box} (R \rhd S) = (P \mathbin{\Box} R) \rhd (Q \mathbin{\Box} S)$
D1	$P \mathbin{\Box} (Q \sqcap R) = (P \mathbin{\Box} Q) \sqcap (P \mathbin{\Box} R)$
Prune	$(a \to P) \mathbin{\Box} a \to (P \sqcap Q) = a \to (P \sqcap Q)$
P0	$P \|_A (Q \|_A R) = (P \|_A Q) \|_A R$
P1	$P \|_A Q = Q \|_A P$
P4–P13	*more axioms for parallel composition follow on the next page*
H1	$(P \sqcap Q) \backslash A = (P \backslash A) \sqcap (Q \backslash A)$
H5	$\big(\mathop{\Box}_{i \in I} (a_i \to P_i)\big) \backslash A = \big(\mathop{\Box}_{a_i \notin A} (a_i \to (P_i \backslash A))\big) \rhd \bigsqcap_{a_i \in A} (P_i \backslash A)$
H6	$\big(\mathbf{div} \mathbin{\Box} \mathop{\Box}_{i \in I} (a_i \to P_i)\big) \backslash A = \big(\mathbf{div} \mathbin{\Box} \mathop{\Box}_{a_i \notin A} (a_i \to (P_i \backslash A))\big) \rhd \bigsqcap_{a_i \in A} (P_i \backslash A)$
H7	$\Big(\big(\mathop{\Box}_{i \in I} (a_i \to P_i)\big) \rhd P'\Big) \backslash A = \big(\mathop{\Box}_{a_i \notin A} (a_i \to (P_i \backslash A))\big) \rhd \big(P' \backslash A \sqcap \bigsqcap_{a_i \in A} (P_i \backslash A)\big)$
H8	$\Big(\big(\mathbf{div} \mathbin{\Box} \mathop{\Box}_{i \in I} (a_i \to P_i)\big) \rhd P'\Big) \backslash A = \big(\mathbf{div} \mathbin{\Box} \mathop{\Box}_{a_i \notin A} (a_i \to (P_i \backslash A))\big) \rhd \big(P' \backslash A \sqcap \bigsqcap_{a_i \in A} (P_i \backslash A)\big)$
R0	$f(P \rhd Q) = f(P) \rhd f(Q)$
R1	$f(P \sqcap Q) = f(P) \sqcap f(Q)$
R2	$f(P \mathbin{\Box} Q) = f(P) \mathbin{\Box} f(Q)$
R3	$f(a \to P) = f(a) \to f(P)$
R4	$f(STOP) = STOP$
R5	$f(\mathbf{div}) = \mathbf{div}$
T0	$(P \rhd Q) \mathbin{\Theta_A} R = (P \mathbin{\Theta_A} R) \rhd (Q \mathbin{\Theta_A} R)$
T1	$(P \sqcap Q) \mathbin{\Theta_A} R = (P \mathbin{\Theta_A} R) \sqcap (Q \mathbin{\Theta_A} R)$
T2	$(P \mathbin{\Box} Q) \mathbin{\Theta_A} R = (P \mathbin{\Theta_A} R) \mathbin{\Box} (Q \mathbin{\Theta_A} R)$
T3	$(a \to P) \mathbin{\Theta_A} Q = a \to (P \mathbin{\Theta_A} Q)$ if $a \notin A$
T4	$(a \to P) \mathbin{\Theta_A} Q = a \to Q$ if $a \in A$
T5	$STOP \mathbin{\Theta_A} Q = STOP$
T6	$\mathbf{div} \mathbin{\Theta_A} Q = \mathbf{div}$

(*continued*)

Table 3. (*continued*)

Below $P = \Box_{i\in I}(a_i \to P_i)$ and $Q = \Box_{j\in J}(b_j \to Q_j)$.

(**P4**)
$$\begin{aligned} P\|_A Q = {} & \Box_{a_i\notin A}(a_i \to (P_i\|_A Q))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to (P\|_A Q_j)) \end{aligned}$$

(**P5**)
$$\begin{aligned} (\mathbf{div}\,\Box\, P)\|_A Q = {} & \mathbf{div}\,\Box\,\Box_{a_i\notin A}(a_i \to (P_i\|_A Q))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to ((\mathbf{div}\,\Box\, P)\|_A Q_j)) \end{aligned}$$

(**P6**)
$$\begin{aligned} (\mathbf{div}\,\Box\, P)\|_A(\mathbf{div}\,\Box\, Q) = {} & \mathbf{div}\,\Box\,\Box_{a_i\notin A}(a_i \to (P_i\|_A(\mathbf{div}\,\Box\, Q)))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to ((\mathbf{div}\,\Box\, P)\|_A Q_j)) \end{aligned}$$

(**P7**)
$$\begin{aligned} (P \rhd P')\|_A Q = {} & (\Box_{a_i\notin A}(a_i \to (P_i\|_A Q))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to ((P \rhd P')\|_A Q_j))) \rhd P'\|_A Q \end{aligned}$$

(**P8**)
$$\begin{aligned} ((\mathbf{div}\,\Box\, P) \rhd P')\|_A Q = {} & (\mathbf{div}\,\Box\,\Box_{a_i\notin A}(a_i \to (P_i\|_A Q))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to (((\mathbf{div}\,\Box\, P) \rhd P')\|_A Q_j))) \\ & \rhd P'\|_A Q \end{aligned}$$

(**P9**)
$$\begin{aligned} (P \rhd P')\|_A(\mathbf{div}\,\Box\, Q) = {} & ((\mathbf{div}\,\Box\,\Box_{a_i\notin A}(a_i \to (P_i\|_A(\mathbf{div}\,\Box\, Q)))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to ((P \rhd P')\|_A Q_j))) \\ & \rhd P'\|_A(\mathbf{div}\,\Box\, Q) \end{aligned}$$

(**P10**)
$$\begin{aligned} ((\mathbf{div}\,\Box\, P) \rhd P')\|_A(\mathbf{div}\,\Box\, Q) = {} & (\mathbf{div}\,\Box\,\Box_{a_i\notin A}(a_i \to (P_i\|_A(\mathbf{div}\,\Box\, Q)))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to (((\mathbf{div}\,\Box\, P) \rhd P')\|_A Q_j))) \\ & \rhd P'\|_A(\mathbf{div}\,\Box\, Q) \end{aligned}$$

(**P11**)
$$\begin{aligned} (P \rhd P')\|_A(Q \rhd Q') = {} & (\Box_{a_i\notin A}(a_i \to (P_i\|_A(Q \rhd Q')))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to ((P \rhd P')\|_A Q_j))) \\ & \rhd (P'\|_A(Q \rhd Q') \sqcap (P \rhd P')\|_A Q') \end{aligned}$$

(**P12**)
$$\begin{aligned} ((\mathbf{div}\,\Box\, P) \rhd P')\|_A(Q \rhd Q') = {} & (\mathbf{div}\,\Box\,\Box_{a_i\notin A}(a_i \to (P_i\|_A(Q \rhd Q')))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to (((\mathbf{div}\,\Box\, P) \rhd P')\|_A Q_j))) \\ & \rhd (P'\|_A(Q \rhd Q') \sqcap ((\mathbf{div}\,\Box\, P) \rhd P')\|_A Q') \end{aligned}$$

(**P13**)
$$\begin{aligned} ((\mathbf{div}\,\Box\, P) \rhd P')\|_A((\mathbf{div}\,\Box\, Q) \rhd Q') = {} & (\mathbf{div}\Box \\ & \Box_{a_i\notin A}(a_i \to (P_i\|_A((\mathbf{div}\,\Box\, Q) \rhd Q')))\Box \\ & \Box_{a_j=b_j\in A}(a_i \to (P_i\|_A Q_j))\Box \\ & \Box_{b_j\notin A}(b_j \to (((\mathbf{div}\,\Box\, P) \rhd P')\|_A Q_j))) \\ & \rhd (P'\|_A((\mathbf{div}\,\Box\, Q) \rhd Q') \sqcap ((\mathbf{div}\,\Box\, P) \rhd P')\|_A Q') \end{aligned}$$

(*continued*)

Table 3. (*continued*)

Below $P = \Box_{i\in I}(a_i \to P_i)$ and $Q = \Box_{j\in J} Q_j$.

(**P14**) $$P\|_A Q = \Box_{a_i\notin A}(a_i \to (P_i\|_A Q)) \rhd \Box_{j\in J}(P\|_A Q_j))$$

(**P15**) $$(\mathbf{div} \Box P)\|_A Q = \big(\mathbf{div} \Box \Box_{a_i\notin A}(a_i \to (P_i\|_A Q))\big) \rhd \Box_{j\in J}((\mathbf{div} \Box P)\|_A Q_j))$$

Below $P = \Box_{i\in I} P_i$ and $Q = \Box_{j\in J} Q_j$.

(**P16**) $$P\|_A Q = \Box_{a_i\notin A}(P_i\|_A Q) \sqcap \Box_{j\in J}(P\|_A Q_j))$$

Proof. $\{\big(P \rhd (Q \rhd R), (P \rhd Q) \rhd R\big), \big((P \rhd Q) \rhd R, P \rhd (Q \rhd R)\big) \mid P, Q, R \text{ processes}\} \cup Id$ is a divergence-preserving coupled simulation. □

Proposition 19. *Axiom* **S3** *is valid for* $\equiv^{\Delta}_{CS}$.

Proof. $\{\big((P \rhd Q) \rhd R, (P \Box Q) \rhd R\big), \big((P \Box Q') \rhd R, (P \rhd Q) \rhd R\big), \big(Q \rhd R, (P \Box Q) \rhd R\big), \big(R, Q \rhd R\big) \mid Q' \sqsupseteq^{\Delta}_{CS} Q\} \cup Id$ is a divergence-preserving coupled simulation. □

Proposition 20. *Axiom* **S4** *is valid for* $\equiv^{\Delta}_{CS}$.

Proof. $\{\big((P \sqcap Q) \rhd R, (P \Box Q) \rhd R\big), \big((P' \Box Q') \rhd R, (P \sqcap Q) \rhd R\big), \big(P \rhd R, (P \Box Q) \rhd R\big), \big(Q \rhd R, (P \Box Q) \rhd R\big), \big(R, Q \rhd R\big) \mid P' \sqsupseteq^{\Delta}_{CS} P \wedge Q' \sqsupseteq^{\Delta}_{CS} Q\} \cup Id$ is a divergence-preserving coupled simulation. Checking this involves Proposition 2. □

Proposition 21. *Axiom* **S5** *is valid for* $\equiv^{\Delta}_{CS}$.

Proof. The relation $\{(STOP \rhd P, P), (P, STOP \rhd P) \mid P \text{ a process}\} \cup Id$ is a divergence-preserving coupled simulation.

□

Proposition 22. *Axiom* **S6** *is valid for* $\equiv^{\Delta}_{CS}$.

Proof. $\{\big((P \rhd Q) \sqcap (R \rhd S), (P \Box R) \rhd (Q \sqcap S)\big), \big((P' \Box R') \rhd (Q \sqcap S), (P \rhd Q) \sqcap (R \rhd S)\big), \big(P \rhd Q, (P \Box R) \rhd (Q \sqcap S)\big), \big(R \rhd S, (P \Box R) \rhd (Q \sqcap S)\big), \big(Q \sqcap S, (P \rhd Q) \sqcap (R \rhd S)\big), \big(S, (P' \Box R') \rhd (Q \sqcap S)\big), \big(S, R \rhd S\big), \big(S, Q \sqcap S\big) \mid P' \sqsupseteq^{\Delta}_{CS} P \wedge R' \sqsupseteq^{\Delta}_{CS} R\} \cup Id$ is a divergence-preserving coupled simulation. □

Proposition 23. *Axiom* **S7** *is valid for* $\equiv^{\Delta}_{CS}$.

Proof. $\{\big((P \rhd Q) \Box (R \rhd S), (P \Box R) \rhd (Q \Box S)\big), \big((P \Box R) \rhd (Q \Box S), (P \rhd Q) \Box (R \rhd S)\big), \big(Q' \Box (R \rhd S), (P \Box R) \rhd (Q \Box S)\big), \big((P \rhd Q) \Box S', (P \Box R) \rhd (Q \Box S)\big), \big(Q' \Box S', Q' \Box (R \rhd S)\big), \big(Q' \Box S', (P \rhd Q) \Box S'\big), \mid Q \Longrightarrow Q' \wedge S \Longrightarrow S'\} \cup Id$ is a divergence-preserving coupled simulation. □

Proposition 24. *Axiom* **D1** *is valid for* $\equiv_{CS}^{\Delta}$.

Proof. $\{\big(P' \,\Box\, (Q \sqcap R), (P \,\Box\, Q) \sqcap (P \,\Box\, R)\big), \big((P \,\Box\, Q) \sqcap (P \,\Box\, R), P \,\Box\, (Q \sqcap R)\big), \big(P' \,\Box\, Q, P' \,\Box\, (Q \sqcap R)\big) \mid P \Longrightarrow P'\} \cup Id$ is a divergence-preserving coupled simulation. □

Proposition 25. *Axiom* **Prune** *is valid for* $\equiv_{CS}^{\Delta}$.

Proof. $\{\big((a{\rightarrow}P){\Box}a{\rightarrow}(P{\sqcap}Q), a{\rightarrow}(P{\sqcap}Q)\big), \big(a{\rightarrow}(P{\sqcap}Q), (a{\rightarrow}P){\Box}a{\rightarrow}(P{\sqcap}Q)\big)\} \cup Id$ is a divergence-preserving coupled simulation. □

Proposition 26. *Axioms* **P0–1** *and* **P4–10** *are valid for strong bisimilarity. Axioms* **P11–16** *are valid for* $\equiv_{CS}^{\Delta}$.

Proof. Straightforward. □

Proposition 27. *Axioms* **U4**, **H1**, **R0–5** *and* **T0–6** *are valid for strong bisimilarity. Axioms* **H5–8** *are valid for* $\equiv_{CS}^{\Delta}$.

Proof. Straightforward. □

Proposition 28. *Axiom* **U1** *is valid for* $\equiv_{CS}^{\Delta}$.

Proof. $\{\big((P \sqcap Q) \vartriangle R', (P \vartriangle R) \sqcap (Q \vartriangle R)\big), \big((P \vartriangle R) \sqcap (Q \vartriangle R), (P \sqcap Q) \vartriangle R\big), \big(P \vartriangle R', (P \sqcap Q) \vartriangle R'\big) \mid R \Longrightarrow R'\} \cup Id$ is a divergence-preserving coupled simulation. □

10 Completeness

Let *Th* be the axiomatisation of Table 3.

Proposition 29. *For each recursion-free CSP process P without interrupt operators there is a CSP process Q in normal form such that* $Th \vdash P = Q$.

Proof. By structural induction on P it suffices to show that for each n-ary CSP operator Op, and all CSP processes $P_1, ..., P_n$ in normal form, also $Op(P_1, ..., P_n)$ can be converted to normal form. This I do with structural induction on the arguments P_i.

- Let $P = STOP$ or **div**. Then P is already in normal form. Take $Q := P$.
- Let $P = a \rightarrow P'$. By assumption P' is in normal form; therefore so is P.
- Let $P = P_1 \sqcap P_2$. By assumption P_1 and P_2 are in normal form. So $P = \Big(\big([\mathbf{div}\,\Box]\,\Box_{i\in I}(a_i \rightarrow R_i)\big) \rhd \sqcap_{j\in J} R_j\Big) \sqcap \Big(\big([\mathbf{div}\,\Box]\,\Box_{l\in L}(a_l \rightarrow R_l)\big) \rhd \sqcap_{j\in M} R_j\Big)$ with $R_j = \big([\mathbf{div}\,\Box]\,\Box_{k\in K_j}(a_{kj} \rightarrow R_{kj})\big)$ for $j \in J \cup M$. With Axiom **S5** I may assume that $J, M \neq \emptyset$. Now Axiom **S6** converts P to normal form.

– Let $P = P_1 \,\Box\, P_2$. By assumption P_1 and P_2 are in normal form. So $P = \Big(\big([\mathbf{div}\,\Box]\,\Box_{i\in I}(a_i \to R_i)\big) \rhd \bigsqcap_{j\in J} R_j\Big) \,\Box\, \Big(\big([\mathbf{div}\,\Box]\,\Box_{l\in L}(a_l \to R_l)\big) \rhd \bigsqcap_{j\in M} R_j\Big)$ with $R_j = \big([\mathbf{div}\,\Box]\,\Box_{k\in K_j}(a_{kj} \to R_{kj})\big)$ for $j \in J \cup M$. With **S5** I may assume that $J, M \neq \emptyset$. Now Axioms **S7** and **D1** convert P to normal form.
– Let $P = P_1 \rhd P_2$. Axioms **S2**–**4** and **D1** convert P to normal form.
– Let $P = P_1 \|_A P_2$. Axioms **P1** and **P4**–**16**, together with the induction hypothesis, convert P to normal form.
– Let $P = P\backslash A$. Axioms **H1** and **H5**–**8**, together with the induction hypothesis, convert P to normal form.
– Let $P = f(P)$. Axioms **R0**–**5**, together with the induction hypothesis, convert P to normal form.
– Let $P = P_1 \,\Theta_A\, P_2$. Axioms **T0**–**6**, together with the induction hypothesis, convert P to normal form.

Lemma 1. *For any CSP expression P in head normal form there exists a saturated CSP expression Q in head normal form.*

Proof. Let $P = \big([\mathbf{div}\,\Box]\,\Box_{i\in I}(a_i \to R_i)\big) \rhd \bigsqcap_{j\in J} R_j$. Then P has the form $S \rhd R$. By Axioms **S1**–**3** $Th \vdash P = (S \,\Box\, R) \rhd R$. By means of Axioms **D1** and **S4** the subexpression $S \,\Box\, R$ can be brought in the form $[\mathbf{div}\,\Box]\,\Box_{l\in L}(a_l \to R_l)$. The resulting term is saturated. □

Definition 7. A CSP expression $\big(\Box_{i\in I}(b_i \to P_i)\big)$ is *pruned* if, for all $i, h \in I$, $b_i = b_h \wedge P_i \sqsupseteq^{\Delta}_{CS} P_h \Rightarrow i = h$.

Theorem 1. *Let P and Q be recursion-free CSP processes without interrupt operators. Then $P \equiv^{\Delta}_{CS} Q$ iff $Th \vdash P = Q$.*

Proof. "⇐" is an immediate consequence of the soundness of the axioms of Th, and the fact that $\equiv^{\Delta}_{CS}$ is a congruence for all operators of CSP.

"⇒": Let $depth(P)$ be the length of the longest trace of P—well-defined for recursion-free processes P. If $P \equiv^{\Delta}_{CS} Q$ then $depth(P) = depth(Q)$. Given $P \equiv^{\Delta}_{CS} Q$, I establish $Th \vdash P = Q$ with induction on $depth(P)$.

By Proposition 29 I may assume, without loss of generality, that P and Q are in normal form. By Lemma 1 I furthermore assume that P and Q are saturated. Let $P = \big([\mathbf{div}\,\Box]\,\Box_{i\in I}(a_i \to R_i)\big) \rhd \bigsqcap_{j\in J} R_j$ and $Q = \big([\mathbf{div}\,\Box]\,\Box_{l\in L}(a_l \to R_l)\big) \rhd \bigsqcap_{j\in M} R_j$ with $R_j = \big([\mathbf{div}\,\Box]\,\Box_{k\in K_j}(a_{kj} \to R_{kj})\big)$ for $j \in J \cup M$, where R_i, R_l and R_{kj} are again in normal form.

Suppose that there are $i, h \in I$ with $i \neq h$, $a_i = a_h$ and $R_i \sqsupseteq^{\Delta}_{CS} R_h$. Then $R_i \sqcap R_h \equiv^{\Delta}_{CS} R_h$ by Proposition 3. Since $depth(R_i \sqcap R_h) < depth(P)$, the induction hypothesis yields $Th \vdash R_i \sqcap R_h = R_h$. Hence Axiom **Prune** allows me to prune the summand $a_i \to R_i$ from $\Box_{i\in I}(a_i \to R_i)$. Doing this repeatedly makes $\Box_{i\in I}(a_i \to R_i)$ pruned. By the same reasoning I may assume that $\Box_{l\in L}(a_l \to R_l)$ is pruned.

Since $P\Uparrow \Leftrightarrow Q\Uparrow$ and P and Q are saturated, P has the **div**-summand iff Q does. I now define a function $f : I \to L$ such that $a_{f(i)} = a_i$ and $R_i \sqsupseteq^{\Delta}_{CS} R_{f(i)}$ for all $i \in I$.

Let $i \in I$. Since $P \xrightarrow{a_i} R_i$, by Definition 4 $Q \stackrel{a_i}{\Longrightarrow} Q'$ for some Q' with $R_i \sqsupseteq^{\Delta}_{CS} Q'$. Hence either there is an $l \in L$ such that $a_l = a_i$ and $R_l \Longrightarrow Q'$, or there is a $j \in M$ and $k \in K_j$ such that $a_{kj} = a_i$ and $R_{kj} \Longrightarrow Q'$. Since P is saturated, the first of these alternatives must apply. By Proposition 2 $Q' \sqsupseteq^{\Delta}_{CS} R_l$ and by Proposition 1 $R_i \sqsupseteq^{\Delta}_{CS} R_l$. Take $f(i) := l$.

By the same reasoning there is a function $g : L \to I$ such that $a_{g(l)} = a_l$ and $R_l \sqsupseteq^{\Delta}_{CS} R_{g(l)}$ for all $l \in L$. Since $\Box_{i\in I}(a_i \to R_i)$ and $\Box_{l\in L}(a_l \to R_l)$ are pruned, there are no different $i, h \in I$ (or in L) with $a_i = a_h$ and $R_i \sqsupseteq^{\Delta}_{CS} R_h$. Hence the functions f and g must be inverses of each other. It follows that $Q = \big([\mathbf{div}\,\Box]\,\Box_{i\in I}(a_i \to R_{f(i)})\big) \rhd \bigsqcap_{j\in M} R_j$ with $R_i \equiv^{\Delta}_{CS} R_{f(i)}$ for all $i \in I$. By induction $Th \vdash R_i = R_{f(i)}$ for all $i \in I$.

So in the special case that $I = M = \emptyset$ I obtain $Th \vdash P = Q$. (*)

Next consider the case $J = \emptyset$ but $M \neq \emptyset$. Let $j \in M$. Since $Q \Longrightarrow R_j$, there is a P' with $P \Longrightarrow P'$ and $R_j \sqsupseteq^{\Delta}_{CS} P'$. Moreover, there is a P'' with $P' \Longrightarrow P''$ and $P'' \sqsupseteq^{\Delta}_{CS} R_j$. Since $J = \emptyset$, $P'' = P' = P$, so $P \equiv^{\Delta}_{CS} R_j$. By (*) above $Th \vdash P{=}R_j$. This holds for all $j \in J$, so by Axiom **I1** $Th \vdash Q = \big([\mathbf{div}\,\Box]\,\Box_{i\in I}(a_i \to R_i)\big) \rhd P$. By Axiom **S1** one obtains $Th \vdash P = Q$.

The same reasoning applies when $M = \emptyset$ but $J \neq \emptyset$. So henceforth I assume $J, M \neq \emptyset$. I now define a function $h : J \to M$ with $Th \vdash R_j = R_{h(j)}$ for all $j \in J$.

Let $j \in J$. Since $P \stackrel{\tau}{\Longrightarrow} R_j$, by Definition 4 $Q \Longrightarrow Q'$ for some Q' with $R_j \sqsupseteq^{\Delta}_{CS} Q'$, and $Q' \Longrightarrow Q''$ for some Q'' with $Q'' \sqsupseteq^{\Delta}_{CS} R_j$. There must be an $m \in M$ with $Q'' \Longrightarrow R_m$. By Definition 4 $R_j \Longrightarrow R'$ for some R' with $R_m \sqsupseteq^{\Delta}_{CS} R'$, and $R' \Longrightarrow R''$ for some R'' with $R'' \sqsupseteq^{\Delta}_{CS} R_m$. By the shape of R_j one has $R'' = R' = R_j$, so $R_j \equiv^{\Delta}_{CS} R_m$. By (*) above $Th \vdash R_j = R_m$. Take $h(j) := m$.

By the same reasoning there is a function $e : M \to J$ with $Th \vdash R_m = R_{e(m)}$ for all $m \in M$. Using Axioms **I1**–**3** one obtains $Th \vdash P = Q$. □

11 Conclusion

This paper contributed a new model of CSP, presented as a semantic equivalence on labelled transition systems that is a congruence for the operators of CSP. It is the finest I could find that allows a complete equational axiomatisation for closed recursion-free CSP processes that fits within the existing syntax of the language. For τ-free system, my model coincides with strong bisimilarity, but in matching internal transitions it is less pedantic than weak bisimilarity.

It is left for future work to show that recursion is treated well in this model, and also to extend my complete axiomatisation with the interrupt operator of Roscoe [15,17].

An annoying feature of my complete axiomatisation is the enormous collections of heavy-duty axioms needed to bring parallel compositions of CSP processes in head normal form. These are based on the expansion law of Milner [12], but a multitude of them is needed due to the optional presence of

divergence-summands and sliding choices in head normal forms. In the process algebra ACP the expansion law could be avoided through the addition of two auxiliary operators: the left merge and the communication merge [3]. Unfortunately, failures-divergences equivalence fails to be a congruence for the left-merge, and the same problems exists for any other models of CSP [8, Sect. 3.2.1]. In [1] an alternative left-merge is proposed, for which failures-divergences equivalence, and also $\equiv^{\Delta}_{CS}$, is a congruence. It might be used to eliminate the expansion law **P4** from the axiomatisation of Table 2. Unfortunately, the axiom that splits a parallel composition between a left-, right- and communication merge (Axiom CM1 in [3]), although valid in the failures-divergences model, is not valid for $\equiv^{\Delta}_{CS}$. This leaves the question of how to better manage the axiomatisation of parallel composition entirely open.

References

1. Aceto, L., Ingólfsdóttir, A.: A theory of testing for ACP. In: Baeten, J.C.M., Groote, J.F. (eds.) CONCUR 1991. LNCS, vol. 527, pp. 78–95. Springer, Heidelberg (1991). doi:10.1007/3-540-54430-5_82
2. Bergstra, J.A., Klop, J.W., Olderog, E.-R.: Failures withoutchaos: a new process semantics for fair abstraction. In: Wirsing, M. (ed.) Formal Description of Programming Concepts - III, Proceedings of the 3th IFIP WG 2.2 working conference, Ebberup 1986, North-Holland, Amsterdam, pp. 77–103 (1987)
3. Bergstra, J.A., Klop, J.W.: Process algebra for synchronous communication. Inform. Control **60**, 109–137 (1984). doi:10.1016/S0019-9958(84)80025-X
4. Brookes, S.D., Hoare, C.A.R., Roscoe, A.W.: A theory of communicating sequential processes. J. ACM **31**(3), 560–599 (1984). doi:10.1145/828.833
5. Brookes, S.D., Roscoe, A.W.: An improved failures model for communicating processes. In: Brookes, S.D., Roscoe, A.W., Winskel, G. (eds.) CONCURRENCY 1984. LNCS, vol. 197, pp. 281–305. Springer, Heidelberg (1985). doi:10.1007/3-540-15670-4_14
6. De Nicola, R.: Two complete Axiom systems for a theory of communicating sequential processes. Inf. Control **64**(1–3), 136–172 (1985). doi:10.1016/S0019-9958(85)80048-6
7. Glabbeek, R.J.: The linear time — branching time spectrum II. In: Best, E. (ed.) CONCUR 1993. LNCS, vol. 715, pp. 66–81. Springer, Heidelberg (1993). doi:10.1007/3-540-57208-2_6
8. van Glabbeek, R.J., Vaandrager, F.W.: Modular specification of process algebras. Theor. Comput. Sci. **113**(2), 293–348 (1993). doi:10.1016/0304-3975(93)90006-F
9. Groote, J.F., Vaandrager, F.W.: Structured operational semantics and bisimulation as a congruence. Inf. Comput. **100**(2), 202–260 (1992). doi:10.1016/0890-5401(92)90013-6
10. Hoare, C.A.R.: Communicating sequential processes. Commun. ACM **21**(8), 666–677 (1978). doi:10.1145/359576.359585
11. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall, Upper Saddle River (1985)

12. Milner, R.: Operational and algebraic semantics of concurrent processes. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, Chap. 19, pp. 1201–1242. Elsevier Science Publishers B.V., North-Holland (1990). Alternatively see Communication and Concurrency. Prentice-Hall (1989), of which an earlier version appeared as A Calculus of Communicating Systems. LNCS, vol. 92. Springer (1980)
13. Olderog, E.-R., Hoare, C.A.R.: Specification-oriented semantics for communicating processes. Acta Informatica **23**, 9–66 (1986). doi:10.1007/BF00268075
14. Parrow, J., Sjödin, P.: Multiway synchronization verified with coupled simulation. In: Cleaveland, W.R. (ed.) CONCUR 1992. LNCS, vol. 630, pp. 518–533. Springer Berlin Heidelberg, Berlin, Heidelberg (1992). doi:10.1007/BFb0084813
15. Roscoe, A.W.: The Theory and Practice of Concurrency. Prentice-Hall, Upper Saddle River (1997). http://www.comlab.ox.ac.uk/bill.roscoe/publications/68b.pdf.
16. Roscoe, A.W.: Revivals, stuckness and the hierarchy of CSP models. J. Logic Algebraic Program. **78**(3), 163–190 (2009). doi:10.1016/j.jlap.2008.10.002
17. Roscoe, A.W.: Understanding Concurrent Systems. Springer, London (2010). doi:10.1007/978-1-84882-258-0