The Complete Axiomatization of Cs-congruence

Joachim Parrow[1, 2, 3] and Peter Sjödin[1]

[1] Swedish Institute of Computer Science, Box 1263, S-16428 Kista, Sweden
[2] Dept. of Teleinformatics, Royal Institute of Technology, Stockholm, Sweden
[3] Dept. of Computer Systems, Uppsala University, Sweden

Abstract. The coupled simulation equivalence is slightly larger than observation equivalence. Where observation equivalence is based on weak bisimulations, coupled simulation equivalence is based on pairs of simulations which coincide at stable states. We establish the corresponding congruence and provide a complete axiomatization by adding a new τ-law, $\tau.(\tau.P + Q) = \tau.P + Q$, to the axiomatization of observation congruence. We further indicate how the definition of the equivalence can be extended to divergent transition systems.

1 Introduction

A problem that naturally arises when working with transition system models of computation is to determine when two given transition systems should be regarded as "equivalent" in an abstract sense. Clearly, identity is too strong an equivalence since it does not equate transition systems that intuitively represent the same behaviour. Language equivalence (or trace equivalence), on the other hand, is often too weak since it equates transition systems that represent different deadlock properties.

We here treat a particular equivalence, *cs-equivalence*, that was introduced in our companion paper [6]. There we argued that cs-equivalence is suitable for implementation-oriented applications, and that it is practical even for large transition systems. The use of cs-equivalence was illustrated through a realistic example—a distributed algorithm for multi-process synchronization. The distributed algorithm was defined as a *distributed synchronizer*, which is a set of processes that cooperate to establish multi-process synchronizations. A *central* synchronizer was defined as an abstraction of the distributed synchronizer. The idea was that the central synchronizer should be simple and intuitively "correct," so the distributed algorithm could be verified by showing the central and distributed synchronizers to be equivalent. However, the two synchronizers select processes for synchronization in slightly different ways: the central synchronizer does this in a single internal step, while the distributed synchronizer gradually narrows down the possible combinations until only one remains.

In order to verify the distributed algorithm, we need an equivalence relation with two features: it should have a practical proof method, and it should not require an exact correspondence between all internal choices in equivalent transition systems. To our knowledge, cs-equivalence is unique in possessing these two desirable properties. The first property is important in any large-scale application, and the second is important in for example verification that a distributed system (which may resolve internal choices through negotiations between components) is equivalent to a non-distributed system (where internal choices are resolved atomically). We briefly explain these points below in comparison with two other popular equivalences: observation equivalence and testing equivalence.

Observation equivalence (originally suggested by Milner [4]) is based on the idea that two equivalent transition systems should be able to mimic each other indefinitely. Formally, it can be defined through the notion of *bisimulation*, which is a binary relation on the states of the systems. Two states are bisimilar if any transition from one of them can be mimicked by an appropriate sequence of transitions from the other, leading again to bisimilar states.

Not surprisingly, observation equivalence is a formally undecidable relation for most process formalisms that can define infinite-state transition systems. In spite of this there is a useful method for checking that a *particular* relation is a bisimulation, namely to go through all transitions in the systems, and for each find a simulating sequence. Although there are many transitions in non-trivial applications, it is often possible to find a structure to the transitions and perform a case analysis over them. The practical importance of this method is that the two systems to be compared for equivalence may be accompanied by some intuition—perhaps supplied by the designer of the systems—for *why* they are equivalent. A bisimulation is a convenient way to make this intuition explicit, and a case analysis over transitions will determine if the intuition is correct.

However, the bisimulation requirement is sometimes unnecessarily strong. If at one state a certain set of internal choices is resolved, at any bisimilar state exactly the same choices must be resolved. If we compare two transition systems which do not resolve internal choices in "lock step", we find that no bisimulation relation between their states exists, and that the systems consequently are not observation equivalent. Thus, although there is a nice method to establish observation equivalence, based on case analysis over transitions, it is inapplicable in some problem areas.

Testing equivalence [2] is a weaker equivalence which does not suffer from this problem. This equivalence presupposes some precisely defined class of *tests*, and two transition systems are equivalent if they can pass exactly the same tests; different classes of tests yield different equivalences. By including traces and deadlock potentials among the tests an interesting equivalence akin to the "failures" equivalence of CSP [3] is obtained. This equivalence does not require bisimilarity, so it can equate transition systems which differ in the way internal choices are resolved. However, to establish that testing equivalence holds it is necessary to perform a case analysis over all *sequences* of transitions, a much more difficult task than analyzing single transitions. Hence, testing equivalence is difficult to apply to large-scale problems.

Cs-equivalence is to our knowledge the only equivalence to combine the advantages of these equivalences. It is based on simulations, so a case analysis over single transitions suffices to verify a cs-equivalence. However, where observation equivalence requires bisimilarity at every state, cs-equivalence only requires bisimilarity at *stable* states (states where no further internal transitions are possible). So cs-equivalence equates more systems than observation equivalence (though less than testing equivalence) and is consequently applicable in areas where observation equivalence is inappropriate.

In the present paper we derive a complete axiomatization for cs-congruence. In brief, this means that we establish a small set of syntactic rewrite laws through which cs-equivalent transition systems can be transformed into each other. Such an axiomatization makes it possible to treat transition systems algebraically, and gives a better picture of how the equivalence relates to other equivalences.

The rest of the paper is structured as follows. In Section 2 we recapitulate the usual definitions of transition systems and observation equivalence, and in Section 3 we give the definition of cs-equivalence and establish its associated congruence. This is given a complete axiomatization in Section 4. Finally, in Section 5 we sum up the results, mention

related work and indicate possible future work; in particular we demonstrate how the equivalence can be generalized to systems with divergences.

2 Preliminaries

We assume a set of *observable actions* Act ranged over by a, b. The *silent action* τ is not in Act, and we use α to range over $Act \cup \{\tau\}$. A silent action represents an internal computation step of an entity.

We use a simple language, in the lines of Milner's CCS [5], to describe transition systems. Thus, our *agents*, ranged over by P,Q,...are given by

$$P \quad ::= \quad \mathbf{0} \quad | \quad \alpha.P \quad | \quad P + Q$$

Sometimes in examples we omit a trailing $\mathbf{0}$, so e.g. $a.\mathbf{0} + \tau.\mathbf{0}$ is written $a + \tau$.

The family of transition relations $\xrightarrow{\alpha}$ on agents is defined inductively as follows. $\mathbf{0}$ is *inaction* and has no transitions. An *action prefix* expression $\alpha.P$ has the one transition $\alpha.P \xrightarrow{\alpha} P$. The *choice* expression $P+Q$ has all transitions from P and Q; i.e. if $P \xrightarrow{\alpha} R$ or $Q \xrightarrow{\alpha} R$ then $P + Q \xrightarrow{\alpha} R$. The *depth* of an agent is the maximal number of nested prefix operators; for example the depth of $a+\tau.b$ is 2. We do not include a parallel operator in our investigation. Although of undisputed practical value, parallel composition is easy to axiomatize with an expansion law as in [5].

We say that P is *unstable* if, for some P', $P \xrightarrow{\tau} P'$, otherwise P is *stable*. If P' is reachable by zero or more transitions from P we call P' a *state* of P.

Given a the transition relations $\xrightarrow{\alpha}$ we construct, in the standard way, the relations $\stackrel{\alpha}{\Longrightarrow}$ meaning that α can be preceded and followed by internal steps. Thus, $P \stackrel{\alpha}{\Longrightarrow} P'$ means $P(\xrightarrow{\tau})^* \xrightarrow{\alpha} (\xrightarrow{\tau})^* P'$. Following Milner we write $\stackrel{\widehat{\alpha}}{\Longrightarrow}$ to mean $\stackrel{\alpha}{\Longrightarrow}$ if $\alpha \neq \tau$ and $\stackrel{\alpha}{\Longrightarrow} \cup \mathbf{Id}$ (where $\mathbf{Id}$ is the identity relation on agents) if $\alpha = \tau$.

We formulate equivalence and congruence relations through *simulation* relations between agents.

Definition 1 (simulation). A *(weak) simulation* is a binary relation $\mathcal{S}$ satisfying

$$\text{whenever } P \mathcal{S} Q \text{ and } P \xrightarrow{\alpha} P' \text{ then } \exists Q' : Q \stackrel{\widehat{\alpha}}{\Longrightarrow} Q' \text{ and } P' \mathcal{S} Q' \ .$$

Following Milner [5] we then define observation equivalence and congruence:

Definition 2 (observation equivalence). A binary relation $\mathcal{S}$ on agents is called a *(weak) bisimulation* if both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are weak simulations. Two agents P and Q are *observation equivalent*, written $P \approx Q$, if they are related by a weak bisimulation.

Definition 3 (observation congruence). Two agents P and Q are *observation congruent*, written $P \approx^c Q$, if

$$\begin{aligned} &\text{whenever } P \xrightarrow{\alpha} P' \text{ then } \exists Q' : Q \stackrel{\alpha}{\Longrightarrow} Q' \text{ and } P' \approx Q' \ , \\ &\text{whenever } Q \xrightarrow{\alpha} Q' \text{ then } \exists P' : P \stackrel{\alpha}{\Longrightarrow} P' \text{ and } P' \approx Q' \ . \end{aligned}$$

Milner writes "=" for observation congruence but we prefer to use "=" as identity. It is a standard result that observation congruence is the largest congruence included in observation equivalence.

3 Coupled Simulations

3.1 Cs-equivalence

The coupled simulations from our companion paper we here call *S-coupled* (coupled by *stability*). The intuition is that two agents can simulate each other, and also bisimulate each other in the stable states (in contrast, observation equivalence requires bisimilarity in every state).

Definition 4 (S-coupled simulation). An *S-coupled* simulation is a pair $(\mathcal{S}_1, \mathcal{S}_2)$, where $\mathcal{S}_1$ and $\mathcal{S}_2^{-1}$ are weak simulations, that satisfies both of the following:

1. If $P\mathcal{S}_1 Q$ and P is stable then $P\mathcal{S}_2 Q$.
2. If $P\mathcal{S}_2 Q$ and Q is stable then $P\mathcal{S}_1 Q$.

Two agents are *cs-equivalent*, written $=_{\rm cs}$, if they are related by both components of an S-coupled simulation.

Definition 5. P is *simulated* by Q, written $P \preceq Q$, if $P\mathcal{S}_1 Q$ for some S-coupled simulation $(\mathcal{S}_1, \mathcal{S}_2)$. P is *co-simulated* by Q, written $P \succeq Q$, if $P\mathcal{S}_2 Q$ for some S-coupled simulation $(\mathcal{S}_1, \mathcal{S}_2)$.

Lemma 6. 1. $(\preceq, \succeq)$ is the largest S-coupled simulation,
2. $\preceq \;=\; \succeq^{-1}$,
3. $=_{\rm cs} \;=\; \preceq \cap \succeq$.

Proof

1. We prove that $(\preceq, \succeq)$ is an S-coupled simulation directly from the definitions. If $P \preceq Q$ then $P\mathcal{S}_1 Q$ for some S-coupled simulation $(\mathcal{S}_1, \mathcal{S}_2)$. Then if $P \xrightarrow{\alpha} P'$ it holds $Q \stackrel{\widehat{\alpha}}{\Longrightarrow} Q'$ for some Q' such that $P'\mathcal{S}_1 Q'$, which implies $P' \preceq Q'$. Also if P is stable then $P\mathcal{S}_2 Q$ whence $P \succeq Q$. With the symmetric argument for $\succeq$ this establishes that $(\preceq, \succeq)$ is an S-coupled simulation; since it includes all S-coupled simulations it must be the largest.
2. $\preceq \;=\; \succeq^{-1}$ follows from the fact that if $(\mathcal{S}_1, \mathcal{S}_2)$ is an S-coupled simulation, then $(\mathcal{S}_2^{-1}, \mathcal{S}_1^{-1})$ is also an S-coupled simulation.
3. If $P =_{\rm cs} Q$ then $P(\mathcal{S}_1 \cap \mathcal{S}_2)Q$ for some S-coupled simulation $(\mathcal{S}_1, \mathcal{S}_2)$; this implies $P(\preceq \cap \succeq)Q$. Conversely, if $P(\preceq \cap \succeq)Q$ then $P =_{\rm cs} Q$ since $(\preceq, \succeq)$ is an S-coupled simulation. □

The relation $\preceq$ is not transitive, as evidenced by $\mathbf{0} \preceq \tau \preceq \tau + a$ and $\mathbf{0} \not\preceq \tau + a$. In spite of this $=_{\rm cs}$ is transitive:

Proposition 7. $=_{\rm cs}$ is an equivalence relation.

Symmetry and reflexivity are easy to establish but transitivity is more difficult. Rather than giving a direct proof here we refer to our technical report [7].

As a simple example, consider a system P consisting of one master process and one slave. The master can either perform an action a, or tell the slave to start execution. The slave, when started, performs some internal computation to determine whether it should perform b or c. The system thus has an externally observable behaviour described by

$$P = a + \tau.(\tau.b + \tau.c)$$

where the first τ represents the communication between master and slave, and the second and third τ represent the internal computation in the slave resolving the choice between b and c. A more abstract description of the system, Q, may employ only one process which either does an a or internally chooses between b and c:

$$Q = a + \tau.b + \tau.c \ .$$

So P has a state $\tau.b + \tau.c$ representing the fact that a has been preempted but b and c are still both possible. There is no bisimilar state in Q, so P and Q are not observation equivalent. But they are cs-equivalent; consider the transition $a+\tau.(\tau.b+\tau.c) \stackrel{\tau}{\longrightarrow} \tau.b+\tau.c$. This is simulated by Q remaining in the initial state, and $\tau.b + \tau.c \preceq a + \tau.b + \tau.c$ as required. Note that $\tau.b + \tau.c \succeq a + \tau.b + \tau.c$ does not hold.

3.2 Cs-congruence

Since we aim at a complete axiomatization we next consider the congruence induced by $=_{\mathrm{cs}}$. It turns out to have an attractive characterization:

Definition 8. P and Q are *cs-congruent*, written $P =^{c}_{\mathrm{cs}} Q$, if

1. $P =_{\mathrm{cs}} Q$,
2. P is stable iff Q is stable.

It may be surprising that this defines a congruence since the corresponding property of observation equivalence fails: there are unstable observation equivalent agents which are not observation congruent, for example $\tau.(\tau + a)$ and $\tau + a$. We proceed to show that $=^{c}_{\mathrm{cs}}$ is indeed a congruence.

Lemma 9. If $P =_{\mathrm{cs}} Q$, then $\alpha.P =^{c}_{\mathrm{cs}} \alpha.Q$.

Proof Directly from the definitions. □

Lemma 10. $P =^{c}_{\mathrm{cs}} Q$ iff for all R, $P + R =_{\mathrm{cs}} Q + R$.

Proof For the direction $\Longrightarrow$, assume $P =^{c}_{\mathrm{cs}} Q$ and choose an arbitrary R. By symmetry it suffices to prove $P + R \preceq Q + R$. So assume $P + R \stackrel{\alpha}{\longrightarrow} P'$. We must establish that $Q + R \stackrel{\widehat{\alpha}}{\Longrightarrow} Q'$ with $P' \preceq Q'$. If the transition from $P + R$ emanates from R, then also $Q + R \stackrel{\alpha}{\longrightarrow} P'$ and $P' \preceq P'$. So assume it emanates from P. If $\alpha \neq \tau$ then from $P \preceq Q$ we get $Q \stackrel{\alpha}{\Longrightarrow} Q'$ and thus also $Q + R \stackrel{\alpha}{\Longrightarrow} Q'$ with $P' \preceq Q'$. So assume that P and thus also Q are unstable, and that $P \stackrel{\tau}{\longrightarrow} P'$.

It is convenient to establish the more general result here that if $P \stackrel{\tau}{\Longrightarrow} P'$ there is a simulating sequence of transitions $Q + R \stackrel{\widehat{\tau}}{\Longrightarrow} Q'$ with $P' \preceq Q'$; the proof goes by induction of the depth of P'. First, from $P \preceq Q$ we get that Q can simulate $P \stackrel{\tau}{\Longrightarrow} P'$ with $Q \stackrel{\widehat{\tau}}{\Longrightarrow} Q'$ such that $P' \preceq Q'$. If this involves at least one τ-transition from Q, i.e. $Q \stackrel{\tau}{\Longrightarrow} Q'$, then also $Q + R \stackrel{\tau}{\Longrightarrow} Q'$ and we are done. So assume that $Q = Q'$, and thus $P' \preceq Q$.

If P' is stable, as in the base case of the induction, then also $P' \succeq Q$. But Q is unstable and thus has at least one τ-transition, and since Q is convergent $Q \stackrel{\tau}{\Longrightarrow} Q''$ for some stable Q''. So $P' \stackrel{\widehat{\tau}}{\Longrightarrow} P''$ with $P'' \succeq Q''$. But P' is stable so $P' = P''$, and Q'' is stable so $P' \preceq Q''$. In conclusion $Q + R \stackrel{\tau}{\Longrightarrow} Q''$ with $P' \preceq Q''$.

There remains the case where also P' is unstable; we will argue that then $P' \preceq Q+R$ and thus $Q+R \stackrel{\widehat{\tau}}{\Longrightarrow} Q+R$ is the required simulating transition. Let $P' \stackrel{\beta}{\longrightarrow} P''$ be any transition from P'. Thus $P \stackrel{\beta}{\Longrightarrow} P''$. If $\beta = \tau$ then by induction we get a simulating transition from $Q+R$. If $\beta \neq \tau$ then by $P' \preceq Q$ we get $Q \stackrel{\beta}{\Longrightarrow} Q'$ and thus also $Q+R \stackrel{\beta}{\Longrightarrow} Q'$ with $P'' \preceq Q'$.

For the direction $\Longleftarrow$, assume $P+R =_{\mathrm{cs}} Q+R$ for all R. By choosing $R = \mathbf{0}$ we get that $P =_{\mathrm{cs}} Q$. Assume that P is unstable and Q is stable; we will derive a contradiction and thus (by symmetry) $P =^{c}_{\mathrm{cs}} Q$. Let the τ-transition from P be $P \stackrel{\tau}{\longrightarrow} P'$, and choose $R = \ell.\mathbf{0}$, such that neither P nor Q has any ℓ-transitions. Since $P+R \stackrel{\tau}{\longrightarrow} P'$ there must be a simulating transition $Q+R \stackrel{\widehat{\tau}}{\Longrightarrow} Q'$ with $P' \preceq Q'$. But Q and R are stable, so $Q' = Q+R$. Since P' is convergent there is a P'' such that $P' \stackrel{\widehat{\tau}}{\Longrightarrow} P''$ where P'' is stable; again this must be simulated such that $P'' \preceq Q+R$. But P'' is stable so also $P'' \succeq Q+R$ must hold; this is impossible since $Q+R \stackrel{\ell}{\longrightarrow} \mathbf{0}$ and P'' has no ℓ-transition. □

Lemma 11. If $P =^{c}_{\mathrm{cs}} Q$ then for any R, $P+R =^{c}_{\mathrm{cs}} Q+R$.

Proof Use the associativity of $+$ and Lemma 10 twice: $P =^{c}_{\mathrm{cs}} Q$ implies that for all R' and R, $P+(R+R') =_{\mathrm{cs}} Q+(R+R')$, i.e. $(P+R)+R' =_{\mathrm{cs}} (Q+R)+R'$, which implies $P+R =^{c}_{\mathrm{cs}} Q+R$. □

We also get that $=^{c}_{\mathrm{cs}}$ is an equivalence (from Lemma 10 and the fact that $=_{\mathrm{cs}}$ is an equivalence), so collecting the results in this section we have:

Corollary 12. $=^{c}_{\mathrm{cs}}$ is the largest congruence included in $=_{\mathrm{cs}}$.

It should be mentioned that this result holds even if a parallel composition operator is added to the language. The relation between observation and cs-equivalence (congruence) is captured by the following two lemmas:

Lemma 13. If $P \approx Q$ then $P =_{\mathrm{cs}} Q$

Proof Directly from the definitions, observing that $(\approx, \approx)$ is an S-coupled simulation. □

Lemma 14. If $P \approx^{c} Q$ then $P =^{c}_{\mathrm{cs}} Q$

Proof Directly from Lemma 10 and the corresponding result for observation congruence ([5, p. 153]), using Lemma 13: $P \approx^{c} Q$ implies $\forall R : P+R \approx Q+R$ implies $\forall R : P+R =_{\mathrm{cs}} Q+R$ which implies $P =^{c}_{\mathrm{cs}} Q$. □

These implications are strict as seen by the following example:

$$a + \tau.(\tau.b + c) \stackrel{=^{c}_{\mathrm{cs}}}{\not\approx} a + \tau.b + c$$

However, for a special class of agents the converses of the last two lemmas hold, and we use this fact in the completeness proof in the next section.

4 Axiomatization

We base our axiomatization of cs-congruence on Milner's results on observation congruence; the following axiom system is sound and complete for observation congruence [5, p. 160]:

	Axioms $\mathcal{A}_2$
A1	$P + Q = Q + P$
A2	$P + (Q + R) = (P + Q) + R$
A3	$P + P = P$
A4	$P + \mathbf{0} = P$
A5	$\alpha.\tau.P = \alpha.P$
A6	$P + \tau.P = \tau.P$
A7	$\alpha.(P + \tau.Q) + \alpha.Q = \alpha.(P + \tau.Q)$

For cs-congruence we add a τ law and get axiom system $\mathcal{A}_{\mathrm{cs}}$:

	Axioms $\mathcal{A}_{\mathrm{cs}}$
A1–A7	
A8	$\tau.(P + \tau.Q) = P + \tau.Q$

A8 says that when a τ occurs immediately under another τ, then this outer τ can be removed. Thus, through this axiom agents can be reduced so that P is stable in any subagent of type $\tau.P$ (this is one of the key steps in the completeness proof below). Of course **A8** is sound for weaker congruences such as testing congruence, but these additionally satisfy further axioms such as $\alpha.P + \alpha.Q = \alpha.(\tau.P + \tau.Q)$, which do not hold for cs-equivalence in general.

Theorem 15. $\mathcal{A}_{\mathrm{cs}}$ is sound for cs-congruence.

Proof All axioms are true if $=$ is replaced by $=^c_{\mathrm{cs}}$. In view of Lemma 14 we only have to check **A8**. Note that $\tau.(P + \tau.Q) \xrightarrow{\tau} P + \tau.Q$ is simulated by $P + \tau.Q \stackrel{\widehat{\tau}}{\Longrightarrow} P + \tau.Q$ and that both sides of this axiom are unstable. □

The rest of this section is devoted to the proof that $\mathcal{A}_{\mathrm{cs}}$ is complete. The proof contains several intricate points and it is substantially more complex than the corresponding result for observation congruence. The main idea is to show that each agent can be reduced to a normal form through $\mathcal{A}_{\mathrm{cs}}$, and that equivalent normal forms are provably equal. In this respect the structure of our proof is similar to that for observation congruence, where Milner defines [5, p. 162] that P is a *full standard form* if

1. $P = \sum_{i=1}^{m} \alpha_i.P_i$, where each P_i is in full standard form,
2. Whenever $P \stackrel{\alpha}{\Longrightarrow} P'$ then $P \xrightarrow{\alpha} P'$,

and shows that for any P, there is a full standard form P' such that $\mathcal{A}_2 \vdash P = P'$ [5, pp. 163]. We have to introduce more requirements on full standard forms to obtain the normal forms suitable for our completeness proof.

Definition 16. $\widehat{P}$ is a *partial sum* of P if $P = \sum_{i \in I} \alpha_i.P_i$ and $\widehat{P} = \sum_{i \in J} \alpha_i.P_i$ and $J \subseteq I$.

Definition 17. P is a *normal form* if

1. $P = \sum_{i=1}^{m} \alpha_i.P_i$ where each P_i is a normal form,

2. Whenever $P \stackrel{\alpha}{\Longrightarrow} P'$ then $P \stackrel{\alpha}{\longrightarrow} P'$,
3. If $\tau.P'$ is a summand of P then P' is stable,
4. Whenever P has a summand of the form $\alpha.(\tau.P' + Q)$, then for any partial sum $\widehat{Q}$ of Q there is a summand $\alpha.R$ of P such that $R =^{c}_{\mathrm{cs}} \tau.P' + \widehat{Q}$.

Below follows an example of an agent (left) and a normal form for it (right).

$$\begin{aligned} a.\tau.(\tau.b + \tau.c) =^{c}_{\mathrm{cs}} \; & a.b + a.c + \\ & a.(\tau.b + b) + a.(\tau.c + c) + \\ & a.(\tau.b + b + c) + a.(\tau.c + b + c) + \\ & a.(\tau.b + \tau.c + b + c) \end{aligned}$$

Lemma 18. For any P there is a normal form N of equal or less depth such that $\mathcal{A}_{\mathrm{cs}} \vdash N = P$.

Proof By induction on the depth of P. For the base case P has depth 0, then it is just a sum of $\mathbf{0}$, and **A3** suffices to rewrite P to $\mathbf{0}$, which is a normal form.

For the inductive step, Milner shows [5, pp. 163] how to rewrite any agent to a sum $\sum_{i=1}^{m} \alpha_i.P_i$ satisfying clause 2. (For this, axiom system $\mathcal{A}_2$ suffices.) By induction all P_i can be assumed to be normal forms. We next demonstrate that with the use of **A8**, P can be rewritten to satisfy clauses 3 and 4 and still satisfy clauses 1 and 2.

For clause 3, we show that any summand $\tau.P'$ of P where P' is unstable can be rewritten to P'. For such a summand $P' = \tau.P'' + R$ where $R = \sum_i \alpha_i.R_i$. Thus $\tau.P' = \tau.(\tau.P'' + R) =^{c}_{\mathrm{cs}} \tau.P'' + R = P'$ by **A8**. So $\tau.P'$ can be replaced by the summands of P'; all these satisfy clause 3 since by induction P' is a normal form. Clearly this rewrite does not invalidate clause 2 since it does not introduce any new transitions $P \stackrel{\alpha}{\Longrightarrow} Q$. Also, it does not invalidate clause 1 since P'' and all R_i are, by induction, normal forms.

For clause 4, let a summand of P be $\alpha.(\tau.P' + Q + R)$. Then, by **A8** and **A7**,

$$\begin{aligned} \alpha.(\tau.P' + Q + R) &=^{c}_{\mathrm{cs}} \alpha.(\tau.(\tau.P' + Q) + R) \\ &=^{c}_{\mathrm{cs}} \alpha.(\tau.(\tau.P' + Q) + R) + \alpha.(\tau.P' + Q) \\ &=^{c}_{\mathrm{cs}} \alpha.(\tau.P' + Q + R) + \alpha.(\tau.P' + Q) \end{aligned}$$

Thus the summand $\alpha.(\tau.P' + Q)$ can be added to P; in this way all required partial sums can be introduced. This operation does not invalidate clause 3 since it does not introduce new summands $\tau.P'$ where P' is unstable. Also, it does not invalidate clause 2 since any new transition $P \stackrel{\alpha}{\Longrightarrow} \widehat{Q}$ that it generates corresponds to one of the new summands $\alpha.\widehat{Q}$. However, it may invalidate clause 1 since in a new summand $\alpha.\widehat{Q}$ (where $\widehat{Q}$ is a partial sum), $\widehat{Q}$ is not necessarily a normal form—$\widehat{Q}$ may not satisfy clause 2 and 4. But by induction $\widehat{Q}$ is provably equivalent to a normal form. Rewriting all such summands to normal form produces the required normal form for P. This concludes the inductive step and the proof. □

The reason for introducing clause 4 in the normal forms is to make possible the following lemma:

Lemma 19. Let P, Q be normal forms, $P \stackrel{\alpha}{\longrightarrow} P'$ and $Q \stackrel{\alpha}{\longrightarrow} Q'$, and $P' \preceq Q'$. Then there exists a U such that $Q \stackrel{\alpha}{\longrightarrow} U$ and $P' =_{\mathrm{cs}} U$.

Proof The proof is by induction on the sum of the depths of P' and Q'. If P' is stable, as in the base case where $P' = Q' = \mathbf{0}$, then immediately $P' \succeq Q'$ whence $P' =_{\mathrm{cs}} Q'$. So suppose P' is unstable. There are two cases:

1. For some P'', R it holds that $P' = \tau.P'' + R$ and $P'' \preceq Q'$. But P' is a normal form so P'' is stable; this implies $P'' \succeq Q'$, so $P'' =_{\mathrm{cs}} Q'$, and we can write

$$P' =_{\mathrm{cs}} \tau.Q' + R \ . \qquad (*)$$

The idea in the following proof is to show that each summand $\beta.R'$ in R can be eliminated while preserving $(*)$. Put $R = \beta.R' + R''$. Because of $P' \preceq Q'$, there must exist a Q'' such that $Q' \stackrel{\widehat{\beta}}{\Longrightarrow} Q''$ and $R' \preceq Q''$. There are two sub-cases for this.
The first sub-case is when $\beta = \tau$ and $R' \preceq Q'$. Then $P' =_{\mathrm{cs}} \tau.Q' + \tau.R' + R''$. But P is in normal form so R' must be stable, and therefore $R' \succeq Q'$ and $R' =_{\mathrm{cs}} Q'$ so

$$P' =_{\mathrm{cs}} \tau.Q' + \tau.Q' + R'' =_{\mathrm{cs}} \tau.Q' + R'' \ .$$

The second sub-case is when $Q' \stackrel{\beta}{\Longrightarrow} Q''$ such that $R' \preceq Q''$, and since Q is a normal form $Q' \stackrel{\beta}{\longrightarrow} Q''$. The depths of R' and Q'' are less than the depths of P' and Q', so by induction there exists a Q''' such that $Q' \stackrel{\beta}{\longrightarrow} Q'''$ and $R' =_{\mathrm{cs}} Q'''$. Let $Q' = \beta.Q''' + T$. Then

$$P' =_{\mathrm{cs}} \tau.(\beta.Q''' + T) + \beta.Q''' + R'' =_{\mathrm{cs}} \tau.(\beta.Q''' + T) + R'' =_{\mathrm{cs}} \tau.Q' + R''$$

where the middle equality follows from **A6** and **A3**.
In conclusion, we have shown that if $R = \beta.R' + R''$ then $P' =_{\mathrm{cs}} \tau.Q' + R''$. By doing this repeatedly we find that $(*)$ holds if we replace R by $\mathbf{0}$, so $P' =_{\mathrm{cs}} \tau.Q' =_{\mathrm{cs}} Q'$, so we fulfil the lemma by $U = Q'$.
2. For no summand $\tau.P''$ of P' does it hold that $P'' \preceq Q'$. Then, because of $P' \preceq Q'$, for each $P' \stackrel{\beta}{\longrightarrow} P''$ there is a corresponding $Q' \stackrel{\beta}{\Longrightarrow} Q''$ (and since Q' is a normal form $Q' \stackrel{\beta}{\longrightarrow} Q''$) such that $P'' \preceq Q''$. Both P'' and Q'' are of less depth than P' and Q', respectively, so by induction there exists a Q''' such that $Q' \stackrel{\beta}{\longrightarrow} Q'''$ and $P'' =_{\mathrm{cs}} Q'''$.
Let $\widehat{Q'}$ be the partial sum of Q' consisting of only such summands $\beta.Q'''$. That is, $\beta.Q'''$ is a summand of $\widehat{Q'}$ if and only if there exists a P'' such that $P' \stackrel{\beta}{\longrightarrow} P''$ and $P'' =_{\mathrm{cs}} Q'''$. Then $P' \preceq \widehat{Q'}$, since for each transition of P' there is a simulating transition in $\widehat{Q'}$, and conversely $P' \succeq \widehat{Q'}$ since only summands which have simulating counterparts in P' are included in $\widehat{Q'}$. In other words, $P' =_{\mathrm{cs}} \widehat{Q'}$.
Now P' is unstable, so Q' must also be unstable (since each transition $\stackrel{\beta}{\longrightarrow}$ from P' is simulated by Q'). Then also $\widehat{Q}$ is unstable (there is a $P' \stackrel{\tau}{\longrightarrow} P''$ so there must be a $Q' \stackrel{\tau}{\longrightarrow} Q''$ such that $P'' =_{\mathrm{cs}} Q''$, and therefore $\tau.Q''$ is a summand of $\widehat{Q}$). Furthermore, $\widehat{Q}$ is a partial sum of Q' so, since Q is a normal form, there is a transition $Q \stackrel{\alpha}{\longrightarrow} R =_{\mathrm{cs}} \widehat{Q} =_{\mathrm{cs}} P'$. Thus we fulfil the lemma with $U = R$. □

The lemma above enables the key result that observation congruence and cs-congruence coincide on normal forms. So our completeness proof goes through by a reduction to the completeness of $\mathcal{A}_2$ for observation congruence.

Lemma 20. Let P, Q be normal forms. Then $P =_{\mathrm{cs}} Q$ iff $P \approx Q$, and $P =^{c}_{\mathrm{cs}} Q$ iff $P \approx^{c} Q$.

Proof In view of Lemmas 13 and 14 it suffices to establish the directions $\Longrightarrow$. For the equivalences, we show that

$$\{(P,Q) : P =_{\mathrm{cs}} Q \text{ and } P,Q \text{ normal forms}\}$$

is a weak bisimulation. So assume that P,Q are normal forms, $P =_{\mathrm{cs}} Q$, and $P \stackrel{\alpha}{\longrightarrow} P'$. Then $Q \stackrel{\widehat{\alpha}}{\Longrightarrow} Q'$ with $P' \preceq Q'$. If $\alpha = \tau$ then P' must be stable, so we immediately get $P' \succeq Q'$, which implies $P' =_{\mathrm{cs}} Q'$. Otherwise $Q \stackrel{\alpha}{\Longrightarrow} Q'$, and since Q is a normal form this means $Q \stackrel{\alpha}{\longrightarrow} Q'$. Then Lemma 19 says that for some Q'', $Q \stackrel{\alpha}{\longrightarrow} Q''$ with $P' =_{\mathrm{cs}} Q''$ as required (P and Q are both normal forms, so P', Q' and Q'' must also be normal forms).

For the congruences, assume $P =^c_{\mathrm{cs}} Q$ and $P \stackrel{\alpha}{\longrightarrow} P'$. Then $Q \stackrel{\widehat{\alpha}}{\Longrightarrow} Q'$ with $P' \preceq Q'$. If the transition involves at least one action, i.e. $Q \stackrel{\alpha}{\Longrightarrow} Q'$, then as above we get $Q \stackrel{\alpha}{\longrightarrow} Q''$ with $P' =_{\mathrm{cs}} Q''$ which, with the first part of this lemma, implies $P' \approx Q''$ as required. If not, then $\alpha = \tau$ and $Q' = Q$. Since P is a normal form P' must be stable, so $P' =_{\mathrm{cs}} Q$. But P is unstable, so (by $=^c_{\mathrm{cs}}$) Q must be unstable, i.e. $Q \stackrel{\tau}{\longrightarrow} Q''$ for some Q''. Thus $P' \stackrel{\widehat{\tau}}{\Longrightarrow} P''$ with $P'' \succeq Q''$; since P' is stable $P' = P''$, i.e. $P' \succeq Q''$; further Q'' is stable (because Q is a normal form) so $P' =_{\mathrm{cs}} Q''$, and since these are normal forms $P' \approx Q''$ by the first part of the lemma. □

Theorem 21. $\mathcal{A}_{\mathrm{cs}}$ is complete for cs-congruence.

Proof Assume $P =^c_{\mathrm{cs}} Q$. By Lemma 18 there are normal forms P' and Q' such that $\mathcal{A}_{\mathrm{cs}} \vdash P = P'$ and $\mathcal{A}_{\mathrm{cs}} \vdash Q = Q'$. Since $\mathcal{A}_{\mathrm{cs}}$ is sound, $P' =^c_{\mathrm{cs}} Q'$, which with Lemma 20 gives that $P' \approx^c Q'$. Then by [5, p. 164] (Proposition 18) $\mathcal{A}_2 \vdash P' = Q'$. Since $\mathcal{A}_{\mathrm{cs}}$ includes $\mathcal{A}_2$ we get $\mathcal{A}_{\mathrm{cs}} \vdash P = Q$. □

5 Conclusion

We have provided a complete axiomatization of cs-congruence ($=^c_{\mathrm{cs}}$) by adding a new τ-law,

$$\tau.(\tau.P + Q) = \tau.P + Q$$

to the axiomatization of observation congruence ($\approx^c$). Compared with the corresponding result for $\approx^c$ [5], our normal forms make more requirements on agents: No two τ may occur after each other (if so the first τ can be removed with the new law), and if $\alpha.(\tau.P + Q + R)$ is a summand then so is $\alpha.(\tau.P + Q)$ (this summand can be added with the new and old τ laws). We have shown that any agent can be rewritten to normal form, and that $\approx^c$ and $=^c_{\mathrm{cs}}$ coincide on normal forms.

Our proofs are substantially different from any completeness result we know, and more complicated than the corresponding result for $\approx^c$. One reason is that induction arguments are harder in the absence of bisimulations. Completeness proofs for e.g. testing equivalence [2] are also complicated but their understanding are facilitated by the intuitive denotational models of processes as acceptance trees. We know of no such model for $=^c_{\mathrm{cs}}$.

A natural question is whether our axiomatization can be extended to the *finite-state* agents obtained by adding recursion to the simple language with "." and "+". We conjecture that adding the laws **R1**–**R5** of [5] will provide a complete inference system,

but the details of the proof remain to be worked out. In the presence of recursion a subtlety arises: Proposition 7 (that $=_{\mathrm{cs}}$ is an equivalence relation) breaks down when we consider *divergent* agents (agents with infinite sequences of τ-transitions).

This can be demonstrated by the following small example, where Ω is a divergent agent (e.g. $\Omega \stackrel{\mathrm{def}}{=} \tau.\Omega$):

$$a.b =_{\mathrm{cs}} a.b + a.\Omega =_{\mathrm{cs}} a.b + a$$

whereas

$$a.b \neq_{\mathrm{cs}} a.b + a \ !$$

If we want to generalize $=_{\mathrm{cs}}$ to divergent agents it appears that we lose one of its main advantages, namely a characterization admitting case analysis over single transitions. A possible generalization is the following: A *weakly coupled* (*W-coupled*) simulation is a pair $(\mathcal{S}_1, \mathcal{S}_2)$, where S_1 and S_2^{-1} are simulations, that satisfies both of the following:

1. If $P \mathcal{S}_1 Q$ then $\exists Q' : Q \stackrel{\widehat{\tau}}{\Longrightarrow} Q'$ and $P \mathcal{S}_2 Q'$.
2. If $P \mathcal{S}_2 Q$ then $\exists P' : P \stackrel{\widehat{\tau}}{\Longrightarrow} P'$ and $P' \mathcal{S}_1 Q$.

Two agents are *cw-equivalent* if they are related by both components of a W-coupled simulation. As can be seen the notion of stability does not enter the definition. Where the S-coupled simulations were coupled at stable states, the weakly coupled simulations are coupled through the relation $\stackrel{\widehat{\tau}}{\Longrightarrow}$. This entails an equivalence which coincides with cs-equivalence for convergent agents. The proof is standard; in particular transitivity follows from the fact that the component-wise composition of two W-coupled simulations is a W-coupled simulation. For convergent agents, an S-coupled simulation can be built from a W-coupled, and vice versa. Essentially, to build a W-coupled simulation it is necessary to add pairs to the S-coupled simulation in order to achieve weak coupling (broadly speaking, the relation must be "closed" under $\stackrel{\widehat{\tau}}{\Longrightarrow}$). Conversely, to build an S-coupled simulation from a W-coupled it is necessary to remove some pairs which might invalidate the S-coupling. Note that these results imply Proposition 7. Full proof details can be found in [7].

A considerable number of alternative equivalences has been axiomatized by van Glabbeek [10]. These are however based on a framework without unobservable actions, and are thus not directly comparable to our work. Some of the equivalences have been generalized to cater for unobservables, the *stable bisimulation* equivalence [8] is one that is close to cs-equivalence[4]. A stable bisimulation is a binary relation $\mathcal{S}$ such that $P \mathcal{S} Q$ implies

$$\text{if } P \stackrel{s}{\Longrightarrow} P' \text{ and } P' \text{ is stable, then } \exists Q' : Q \stackrel{s}{\Longrightarrow} Q' \text{ and } P' \mathcal{S} Q'$$

and vice versa. Here s is a *sequence* of actions, so in order to verify that a relation is a stable bisimulation it is necessary to analyze all possible sequences of transitions from all states. This means that stable bisimulation equivalence can be difficult to establish for large examples. It lies between $=_{\mathrm{cs}}$ and testing equivalence (the intersection of the two components of a coupled simulation is a stable bisimulation), and its distinguishing axiom (which does not hold for $=_{\mathrm{cs}}$) is $\alpha.P + \alpha.Q = \alpha.(\tau.P + \tau.Q)$. For divergent processes stable bisimulation equivalence suffers from the same problem as $=_{\mathrm{cs}}$: it is not transitive.

[4] This reference is hard to access; we base our exposition on personal communications from Rob van Glabbeek.

An alternative characterization in terms of *contrasimulations* remedies this: a relation $\mathcal{S}$ is a contrasimulation if $P\mathcal{S}Q$ implies

$$P \stackrel{s}{\Longrightarrow} P' \text{ implies } Q \stackrel{s}{\Longrightarrow} Q' \text{ and } Q'\mathcal{S}P'$$

(note the reversed order in the consequent!), and P and Q are contrasimilar if $P\mathcal{S}Q$ and $Q\mathcal{S}P$ for some contrasimulation $\mathcal{S}$.

There are several other possible avenues of further work. Many other equivalences between observation equivalence and testing equivalence have recently been surveyed by van Glabbeek [9], and it remains to be seen if any of them is practically useful in the same way as cs-equivalence. Another idea is to implement a decision procedure for finite-state systems; an easy way would be to implement the normalization procedure, as sketched in the proof of Lemma 18, within the Concurrency Workbench [1] and then use the algorithm for observation equivalence. The algorithmic complexity of this is unclear—the normalization may generate an exponential blowup in the size of terms—and it would be interesting to determine whether there is a polynomial algorithm for cs-equivalence (observation equivalence has a polynomial algorithm but testing equivalence is PSPACE hard). However, the main practical motivation for cs-equivalence is in the method to verify that a pair of relations constitute a coupled simulation. We believe that tool development for support of this would be more rewarding (but also more time consuming).

Acknowledgement

We are grateful for extended discussions with Rob van Glabbeek on the axiomatization of stable bisimulation and similar equivalences. Rob also suggested the definition of weakly coupled simulations (in analogy with contrasimulations) to cater for divergent agents.

References

1. R. Cleaveland, J. Parrow, and B. Steffen. A semantics-based verification tool for finite-state systems. In E. Brinksma, G. Scollo, and C. A. Vissers, editors, *Protocol Specification, Testing, and Verification, IX*, pages 287–302. North-Holland, 1989.
2. M. Hennessy. *Algebraic Theory of Processes.* The MIT Press, 1988.
3. C. A. R. Hoare. *Communicating Sequential Processes.* Prentice-Hall International, 1985.
4. R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1980.
5. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.
6. J. Parrow and P. Sjödin. Multiway synchronization verified with coupled simulation. In W. R. Cleaveland, editor, *Proceedings of CONCUR '92*, volume 630 of *Lecture Notes in Computer Science*, pages 518–533. Springer-Verlag, 1992.
7. J. Parrow and P. Sjödin. The complete axiomatization of cs-congruence. Research report, Swedish Institute of Computer Science, 1994. In preparation.
8. R. J. van Glaabeek. De semantiek van eidige, sequentiële processes met interne acties. Syllabus processemantieken, deel 2. Handwritten manuscript, in Dutch, 1988.
9. R. J. van Glaabeek. The linear time – branching time spectrum II. In E. Best, editor, *Proceedings of CONCUR '93*, volume 715 of *Lecture Notes in Computer Science*, pages 66–80. Springer-Verlag, 1993.
10. R. J. van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions.* PhD thesis, Free University of Amsterdam, The Netherlands, 1990.

This article was processed using the LaTeX macro package with LLNCS style