ELSEVIER

Theoretical Computer Science 139 (1995) 315–354

Theoretical Computer Science

Deciding observational congruence of finite-state CCS expressions by rewriting☆

P. Inverardi[a,*], M. Nesi[a,b]

[a] *Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, via S. Maria 46, I-56126 Pisa, Italy*

[b] *Computer Laboratory, University of Cambridge, New Museum Site, Pembroke Street, Cambridge CB2 3QG, UK*

Received March 1991; revised January 1994
Communicated by M. Nivat

Abstract

We propose a term rewriting approach to verify observational congruence between guarded recursive (finite-state) CCS expressions. Starting from the complete axiomatization of observational congruence for this subset of CCS, a non-terminating rewriting relation has been defined. This rewriting relation is ω-canonical over a subclass of infinite derivations, *structured fair derivations*, which compute all the ω-normal forms. The rewriting relation is shown to be complete with respect to the axiomatization by proving that every structured fair derivation computes a term that denotes an $r\tau$-normal process graph. The existence of a finite representation for ω-normal forms allows the definition of a *rewriting strategy* that, in a finite number of rewriting steps, decides observational congruence of guarded recursive (finite-state) CCS expressions.

1. Introduction

The calculus of communicating systems (CCS) [15, 18] is a formalism for describing and reasoning about concurrent systems. One of the most interesting features of CCS is the algebraic characterization of its semantics, besides the usual operational one that is based on the labelled transitions interpretation of the language. As it is well known, it is possible to equip CCS with several different semantics [3, 8, 21] that define which processes can be considered to be equivalent with respect to a certain *behaviour*. Often, verification of properties of concurrent systems is to prove the behavioural equivalence of different specifications of the same system. In the past few

☆ Work partially supported by Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo.

* Corresponding author. Present address: Dipartimento di Matematica Pura ed Applicata, Università di L'Aquila, via Vetoio, Coppito, 67010 L'Aquila, Italy.

SSDI 0304-3975(94)00074-S

years there has been a growing interest in the field of the analysis and verification of properties for CCS-like languages and a number of tools and approaches have been proposed and realized (for a survey see [12]).

In this framework we have undertaken a project [4] whose main goal is to develop a verification system for CCS-like languages entirely based on equational reasoning. Rewriting methods appear to be the more suitable techniques to be used. In fact, a term rewriting approach can be adopted both to execute the operational semantics of these languages, as advocated in a general framework in [9, 14], and to verify behavioural equivalences defined over CCS expressions.

In particular, the axiomatic presentation of behavioural equivalences can be used by executing an equivalent term rewriting system obtained, if it exists, by means of a completion process [5]. In [4] this approach has been applied to the axiomatic presentation for observational congruence over finite CCS as given in [8, 16]. When trying to derive an equivalent term rewriting system from the axiomatization for observational congruence, it results that the completion process diverges, i.e. the term rewriting system has an infinite number of rules. We have coped with this divergence by defining a *rewriting strategy* [10] that is able to compute the normal form of a finite CCS term and verify the observational congruence of two finite terms without performing any completion. In doing that, we have been supported by a notion of *normal form* for a finite term with respect to observational congruence (OBS-normal form).

In this paper we extend our rewriting strategy to deal with guarded recursive (finite-state) CCS terms. A correct and complete axiomatization for observational congruence over such a subset of CCS has been given in [17], but, unlike finite CCS, the completeness of such an axiomatization has not been proved by resorting to a notion of recursive OBS-normal form. Thus, no explicit information about the existence and the structure of the normal form of a recursive CCS term has been provided. Nevertheless, CCS terms can be characterized as *process graphs*. In [1] the notion of *unique* normal process graph with respect to observational congruence is defined. This has influenced the definition of our rewriting relation, $\rightarrow_{r\_obs}$, over recursive terms.

The presence of an unfolding rule for recursion makes $\rightarrow_{r\_obs}$ non-terminating. This has led to the use of the theory of *infinite rewritings* developed in [7], where some conditions on infinite relations, namely *left linearity*, and on infinite derivations, namely *fairness*, are required in order to compute the ω-normal form of a term as the limit of an infinite derivation. Our relation $\rightarrow_{r\_obs}$ does not satisfy the left linearity requirement, but we are still able to obtain ω-normal forms as limits of derivations by restricting to a particular subclass of infinite derivations, *structured fair derivations*, and by applying $\rightarrow_{r\_obs}$ modulo a congruence relation, which identifies those terms that, although syntactically different, have equivalent unfolding semantics. The congruence relation can be decided through a *canonical transformation* that reduces any recursive term to the equivalent canonical one [2].

Moreover, the ω-canonicity of $\rightarrow_{r\_obs}$ can be proved with respect to structured fair derivations and for any term a finite representation of its ω-normal form,

recursive normal form, can be defined and computed in a finite number of derivation steps. Given these results, the completeness of $\rightarrow_{r\_obs}$ with respect to the axiomatization of observational congruence, i.e. any two observational congruent recursive terms admit the same ω-normal form, is proved by showing that a recursive normal form denotes a normal process graph. Finally, we define a rewriting strategy to compute a recursive normal form with respect to $\rightarrow_{r\_obs}$, thus obtaining a decision procedure for observational congruence of guarded recursive (finite-state) CCS expressions.

2. Basic ingredients

2.1. Term rewriting systems

We assume that the reader is familiar with the basic concepts of term rewriting systems. We summarize the most relevant definitions below, while we refer to [5–7] for more details.

Let $\mathscr{F} = \bigcup_n \mathscr{F}_n$ be a set of function symbols, where $\mathscr{F}_n$ is the set of symbols of *arity* n. Let $\mathscr{T}$ denote the set $\mathscr{T}(\mathscr{F}, \mathscr{X})$ of (finite, first-order) terms with function symbols $\mathscr{F}$ and variables $\mathscr{X}$. A binary relation $\succ$ is a *partial ordering* if it is irreflexive and transitive. A partial ordering $\succ$ on $\mathscr{T}$ is *well-founded* if there is no infinite descending sequence $t_1 \succ t_2 \succ \cdots$ of terms in $\mathscr{T}$. A relation $\succ$ on $\mathscr{T}$ is *monotonic* if $s \succ t$ implies $f(\cdots s \cdots) \succ f(\cdots t \cdots)$ for all f in $\mathscr{F}$ and for all terms in $\mathscr{T}$ (*replacement property*). A partial ordering $\succ$ on $\mathscr{T}$ is a *simplification ordering* if it is monotonic and $f(\cdots t \cdots) \succ t$ for all f in $\mathscr{F}$ and for all terms in $\mathscr{T}$ (*subterm property*). For any partial ordering $\succ$ on $\mathscr{T}$, the *multiset ordering* $\succ\!\!\succ$ is the smallest partial ordering containing the following relation between multisets: $S \cup \{s\} \succ\!\!\succ S \cup \{t_1, \ldots, t_n\}$ for $s \succ t_1, \ldots, t_n$ $(n \geqslant 0)$. If $\succ$ is well-founded so is $\succ\!\!\succ$.

Let $\succ$ be a partial ordering on $\mathscr{F}$. The (*generalized*) *recursive path ordering* (rpo) on $\mathscr{T}$ is defined recursively as follows:

$s = f(s_1, \ldots, s_n) \succ_{\text{rpo}} g(t_1, \ldots, t_m) = t$ if and only if

$f = g$ and $\{s_1, \ldots, s_n\} \succ\!\!\succ_{\text{rpo}} \{t_1, \ldots, t_m\}$ or

$f \succ g$ and $s \succ_{\text{rpo}} t_i$ for $i = 1, \ldots, m$ or

$f \not\succcurlyeq g$ and $s_i \succcurlyeq_{\text{rpo}} t$ for some i, $i = 1, \cdots, n$

and $s \succ_{\text{rpo}} x$ if and only if $x \in \mathscr{V}ar(s)$, where $\succ\!\!\succ_{\text{rpo}}$ is the extension of $\succ_{\text{rpo}}$ to multisets and $\succcurlyeq_{\text{rpo}}$ is $\succ_{\text{rpo}}$ or the permutation equivalence of subterms. It holds that any rpo is a simplification ordering.

An *equational theory* is any set $E = \{(s, t) \mid s, t \in \mathscr{T}\}$. Elements (s, t) are called *equations* and written $s = t$. Let $\sim_E$ be the smallest symmetric relation that contains E and is closed under monotonicity and substitution. Let $=_E$ be the reflexive-transitive closure of $\sim_E$.

Given an equational theory E over $\mathscr{T}$, we define that $f \in \mathscr{F}$ is an *AC operator* if E contains the associative and commutative laws for f, i.e. $f(f(x,y),z)=f(x,f(y,z))$ and $f(x,y)=f(y,x)$. An *AC term* is a term which contains AC operators.

A *term rewriting system* (TRS) R is any set $\{(l_i, r_i) \mid l_i, r_i \in \mathscr{T}, \mathscr{V}ar(r_i) \subseteq \mathscr{V}ar(l_i)\}$. The pairs (l_i, r_i) are called *rewriting rules* and written $l_i \to r_i$. The *rewriting relation* $\to_R$ over $\mathscr{T}$ is defined as the smallest relation containing R that is closed under monotonicity and substitution. A term t *rewrites* to a term s, written $t \to_R s$, if there exist $l \to r$ in R, a substitution σ and a subterm $t|_u$ at the position u, called *redex*, such that $t|_u = \sigma l$ and $s = t[\sigma r]_u$. A term t is said to *overlap* a term t' if t unifies with a non-variable subterm of t' (after renaming the variables in t so as not to conflict with those in t'). If $l \to r$ and $s \to t$ are two rewriting rules (with distinct variables), u is a position of a non-variable subterm of s, and σ is a most general unifier for $s|_u$ and l, then the equation $\sigma t = \sigma s[\sigma r]_u$ is a *critical pair* formed from those rules. A TRS R is *left linear* if the left-hand side l of each rule $l \to r$ in R has at most one occurrence of any variable. We use $|R|$ to denote the maximum depth of a left-hand side of a TRS R.

Let $\overset{+}{\to}$ and $\overset{*}{\to}$ denote the transitive and reflexive-transitive closure of $\to$, respectively. A TRS R is *terminating* if there is no infinite sequence $t_1 \to_R t_2 \to_R \cdots$ of rewriting steps in R. A TRS R is *confluent* if whenever $s \,{}_R\overset{*}{\leftarrow} t \overset{*}{\to}_R q$, there exists a term t' such that $s \overset{*}{\to}_R t' \,{}_R\overset{*}{\leftarrow} q$, and R is *locally confluent* if whenever $s \,{}_R\leftarrow t \to_R q$, there exists a term t' such that $s \overset{*}{\to}_R t' \,{}_R\overset{*}{\leftarrow} q$. A term t is in *R-normal form* if there is no term s such that $t \to_R s$. A term s is an *R-normal form* of t if $t \overset{*}{\to}_R s$ and s is in R-normal form; in this case we write $t \to !_R s$. A TRS R is *canonical* if it is terminating and confluent.

The notion of ordering is used to correctly direct the rules of a TRS so that it is terminating. In presence of AC operators, an rpo is able to handle commutative operators, but it cannot handle the associative ones. The notion of rpo is then extended by defining the *associative path ordering* $>_{\text{apo}}$. In the simplified case of the theory we will deal with, where only the operator "+" is AC, in order to define an apo it is enough to consider any rpo, provided that (i) the precedence ordering on $\mathscr{F}$ assigns minimal precedence to the AC operator "+"; (ii) when ordering terms, "+" becomes varyadic and any AC term t is transformed into its "flattened" version *flat*(t) i.e. any deeper summand becomes a top level summand, e.g. the term $+(a, +(b,c))$ is treated as $+(a,b,c)$.

An *equational* TRS is a tuple (R, E), where R is a TRS and E is an equational theory. The rewriting relation modulo E, written $\to_{R/E}$, is defined as $=_E \bullet \to_R \bullet =_E$, where $\bullet$ denotes composition of relations.

Let $\mathscr{T}^\infty$ denote the set $\mathscr{T}^\infty(\mathscr{F}, \mathscr{X})$ of finite and infinite terms with function symbols $\mathscr{F}$ and variables $\mathscr{X}$. It is possible to form a complete ultra-metric space on $\mathscr{T}^\infty$ by defining a notion of *distance* d between two terms s, t such that $d(s,t) = 1/2^{v(s,t)}$, where $v(s,t)$ is the smallest depth of a symbol occurrence at which terms s and t differ, with the convention that $d(t,t)=0$.

Given a TRS R, it is straightforward to extend $\to_R$ over $\mathscr{T}^\infty$. Let $\to$ be a (possibly non-terminating) rewriting relation. A term t (ω-)*rewrites* to t', written $t \to^\omega t'$, if $t \overset{*}{\to} t'$

or if there exists an infinite *derivation* $t=t_0\to t_1\to\cdots\to t_n\to\cdots$ such that $\lim_{n\to\infty} t_n=t'$. The relation $\to$ is *ω-converging* if for any infinite derivation $t_0\to t_1\to\cdots\to t_n\to\cdots$ of terms, the limit $\lim_{n\to\infty} t_n$ exists. The relation $\to$ is *top-terminating* if there are no infinite derivations $t_0\to t_1\to\cdots\to t_n\to\cdots$ with infinitely many rewrites at the topmost position. The relation $\to$ is *ω-confluent* if whenever $s\,{}^{\omega}\!\leftarrow t\to{}^{\omega} q$, there exists a term t' such that $s\to{}^{\omega} t'\,{}^{\omega}\!\leftarrow q$. The relation $\to$ is *ω-canonical* if it is ω-converging and ω-confluent. A term t' is an *ω-normal form* of t if $t\to{}^{\omega} t'$ and t' is minimal for $\to$, i.e. if $t'\to t''$, then $t''=t'$. Thus, an ω-normal form need not be irreducible. The relation $\to$ is *ω-normalizing* if every finite term in $\mathscr{T}$ admits an ω-normal form in $\mathscr{T}^{\infty}$.

A derivation $t_0\to t_1\to\cdots\to t_n\to\cdots$ is *fair* if whenever there is a rule $l\to r$ and a position u such that, for all n past some N, the subterm $t_n|_u$ is a redex for $l\to r$, then (at least) one of the rewriting steps $t_n\to t_{n+1}$ $(n\geqslant N)$ is an application of $l\to r$ at u.

Thus, a fair derivation guarantees that a redex does not persist forever. Note that this definition does not prevent the fact that the same rewriting rule is applicable infinitely many times at different positions.

Theorem 2.1 (Dershowitz et al. [7, Proposition 5.1]). *If R is a top-terminating* TRS, *then it is ω-converging.*

For left linear TRS's, fair derivations compute ω-normal forms at the limit.

Theorem 2.2 (Dershowitz et al. [7, Theorem 4.3]). *Let R be a left linear* TRS. *If a term* $t_0\in\mathscr{T}$ *admits an ω-normal form* $t_\infty\in\mathscr{T}^{\infty}$, *then there exists a fair derivation* $t_0\to t_1\to\cdots\to t_n\to\cdots\to{}^{\omega} t_\infty$ *with limit* t_∞.

Theorem 2.3 (Dershowitz et al. [7, Theorem 4.4]). *Let R be a left linear* TRS. *For any fair derivation* $t_0\to t_1\to\cdots\to t_n\to\cdots\to{}^{\omega} t_\infty$, *the limit* t_∞, *if it exists, is an ω-normal form of* t_0.

2.2. Observational congruence over CCS expressions

Let $\mathscr{T}(\mathscr{P},\mathscr{X})$ with $\mathscr{P}_0=\{\text{nil}\}$, $\mathscr{P}_1=\{\tau.,a.,b.,c.,\ldots\}$ and $\mathscr{P}_2=\{+\}$, be the class of CCS expressions representing finite processes, which from now on we refer to as *finitary* CCS expressions. The set $\mathscr{P}_1$ of action prefix operators is ranged over by "$\mu.$", τ is the so-called *internal action* and $\mathscr{X}$ is the set of process variables $\{E, E1, E2, \ldots, F, G, \ldots\}$.

The operational semantics of the above operators is given by the following inference rules:

$$\mu.E \xrightarrow{\mu} E$$

$$E1\xrightarrow{\mu} E2 \quad \text{implies} \quad E1+E\xrightarrow{\mu} E2 \text{ and } E+E1\xrightarrow{\mu} E2$$

The following axiomatization OBS for observational congruence over $\mathscr{T}(\mathscr{P},\mathscr{X})$ has been proved correct and complete with respect to bisimulation in [8, 16]:

S1. $E+(F+G)=(E+F)+G$ T1. $\mu.\tau.E=\mu.E$

S2. $E+F=F+E$ T2. $\tau.E+E=\tau.E$

S3. $E+\text{nil}=E$ T3. $\mu.(E+\tau.F)+\mu.F=\mu.(E+\tau.F)$

S4. $E+E=E$

The completeness of OBS has been shown by resorting to a notion of unique (modulo associativity and commutativity of the "+" operator) *OBS-normal form* of a term with respect to observational congruence. Two finitary expressions E and F can be proved observationally congruent by reducing them to their OBS-normal forms and then checking these normal forms for equivalence modulo the AC axioms S1, S2.

The intuition behind this axiomatization is that, in order to compute the OBS-normal form of a term, those summands which are "semantically contained" in others through the operational notion of μ-derivative have to be deleted. A term E' is a *μ-derivative* of E, written $E \overset{\mu}{\Rightarrow} E'$, if $E \overset{\tau^*}{\to} \overset{\mu}{\to} \overset{\tau^*}{\to} E'$, where $\overset{\tau^*}{\to}$ is the reflexive-transitive closure of the transition relation $\overset{\tau}{\to}$. The notion of semantic redundancy of a term is stated in the so-called absorption lemma [8, 16].

Absorption Lemma. If E' *is a* μ*-derivative of* E *and* OBS $\vdash E'=F$, *then* OBS $\vdash E+\mu.F=E$.

The OBS-normal form of a term is defined as follows. A term $\sum\mu_i.E_i$ is a *proper normal form* if (i) it does not take the form $\tau.E'$ for some term E'; (ii) each E_i is a proper normal form; (iii) for $k\neq j$ no μ_k-derivative of $\mu_j.E_j$ is equivalent to E_k modulo the AC axioms. An OBS-normal form is either E or $\tau.E$, where E is a proper normal form.

In [10] a rewriting relation $\to_{\text{f\_obs}}$ (there called $\to_{\text{strat}}$) has been defined and proved correct and complete with respect to OBS. This relation computes the OBS-normal form of a finitary CCS expression by implementing the absorption lemma. The main feature of $\to_{\text{f\_obs}}$ is that it makes use of control strategies and selection criteria in order to keep some of the equations as equations, i.e. allowing expansions besides reductions, at the same time remaining a deterministic and complete strategy.

The starting point of $\to_{\text{f\_obs}}$ is the TRS R_{OBS} obtained by directing the axioms S3, S4, T1, T2 and T3 according to a chosen apo $\succ$.

R_{OBS} r1. $E+\text{nil}\to E$

r2. $E+E\to E$

r3. $\mu.\tau.E\to\mu.E$

r4. $\tau.E+E\to\tau.E$

r5. $\mu.(E+\tau.F)+\mu.F\to\mu.(E+\tau.F)$

R_{OBS} is terminating but is not confluent modulo AC: during the AC-completion process infinitely many critical pairs are generated from the overlapping of r2, r4 and r5 and they do not reduce to identity [10]. Note that these rules, and those derived from critical pairs, rewrite terms by deleting one of the summands of their left-hand side.

In order to define a rewriting strategy which is complete with respect to the axiomatic presentation, we have to cope with all the *critical peak* situations, i.e. when a term can be rewritten by means of two (or more) rules.

In the critical peak above, let us suppose that t can be rewritten into $t1$ and $t2$ by applying the rules r and r', respectively, and $t1 \succ\!\!- t2$. This means that $t1$ can be rewritten into $t2$ by applying the rule derived from the critical pair associated to the overlapping of r with r'.

The definition of $\rightarrow_{f\_obs}$ is based on the idea that all critical peaks have to be recognized and the application of the rule derived from the associated critical pair has to be simulated. The strategy can be seen as composed of two phases. The first phase, R_{OBS}-*normalization*, normalizes the input term with respect to R_{OBS}. The second phase, *absorption*, works on the resulting term by looking for critical peak situations and summands to be deleted according to observational congruence. This is done by rewriting the term with T2 and T3 as expansion rules (*expansion* process which, roughly speaking, corresponds to moving up along the peak, on the left) and, as soon as possible, by deleting the redundant summands by means of R_{OBS} (*reduction* process which, roughly speaking, corresponds to deleting the top-level summand which would be deleted by applying the rule derived from the associated critical pair). When applying such reductions, a specific redex selection criterion is used that prevents those reductions which are exactly opposite to the previous expansions by T2 and T3. Another criterion is then needed to stop the expansion and reduction steps, which are applied as long as there exist summands to be deleted. Finally, to obtain the OBS-normal form, the current term is rewritten by applying the reductions opposite to the previous expansions (*contraction* process which, roughly speaking, corresponds to moving down along the peak) by using a redex selection criterion that selects the smallest redexes with respect to the fixed term ordering.

This strategy can be defined as the following regular expression (r^* means repetition of the rule r as long as its applicability conditions are satisfied, and ";" means sequencing of rules):

$$\rightarrow_{f\_obs} =_{def} R_{OBS}\text{-normalization; absorption}$$

where absorption $=_{def}$ (expansion; reduction)*; contraction*.

Example 2.4. Let us illustrate how the strategy works, and consider the term $t = \tau.(E1 + E2 + E3) + E2$. This term is in normal form with respect to R_{OBS} but can

still be reduced in the equational theory to a smaller (with respect to the chosen ordering) term. If we give t as input to $\rightarrow_{\text{f\_obs}}$, we will obtain:

$$\tau.(E1+E2+E3)+(E1+E2+E3)+E2$$

$$\uparrow \text{ expansion } \tau.E+E\leftarrow\tau.E \qquad \downarrow \text{ reduction in } R_{\text{OBS}} \text{ modulo } AC \; E+E\rightarrow E$$

$$\tau.(E1+E2+E3)+E2 \qquad \tau.(E1+E2+E3)+(E1+E2+E3)$$

$$\downarrow \text{ contraction } \tau.E+E\rightarrow\tau.E$$

$$\tau.(E1+E2+E3)$$

It is easy to see that what we have done by applying the strategy is to go along the critical peak which would generate, during the completion process, one of the new infinitely many rewriting rules. In particular, the rule $\tau.(E1+E2+E3)+E2\rightarrow\tau.(E1+E2+E3)$ is the one whose application we need to simulate in order to reduce the term t.

The rewriting strategy $\rightarrow_{\text{f\_obs}}$ is *sound*: every rewriting step applies an axiom of OBS, thus preserving the observational congruence among terms. In [10], $\rightarrow_{\text{f\_obs}}$ has been shown to be *correct*: if $\rightarrow_{\text{f\_obs}}$ with input E returns E', then E' is an OBS-normal form of E. *Completeness* is a corollary of correctness: if E has an OBS-normal form E', then $\rightarrow_{\text{f\_obs}}$ with input E returns E' or a term which is equivalent to E' modulo associativity and commutativity. The detailed definition of $\rightarrow_{\text{f\_obs}}$ is reported in Appendix A.

Let us now introduce the language $\mathcal{T}(\mathcal{R},\mathcal{X})$ of *recursive* CCS expressions which is obtained by properly extending $\mathcal{T}(\mathcal{P},\mathcal{X})$ to deal with the recursion operator rec: $\mathcal{R}=\mathcal{P}\cup\mathcal{R}_0=\{x,y,z,\ldots\}\cup\mathcal{R}_1=\{\text{rec } x.,\text{rec } y.,\text{rec } z.,\ldots\}$. In $\mathcal{X}$ the set of process constant $\mathcal{R}_0$ is ranged over by $\{X,Y,Z,\ldots\}$, $\{\text{rec } X.,\text{rec } Y.,\text{rec } Z.,\ldots\}$ range over $\mathcal{R}_1$ and $\{E,E1,E2,\ldots,F,G,\ldots\}$ is the set of variables denoting recursive CCS expressions.

The set $\{x,y,z,\ldots\}$ identifies what in the CCS terminology are called *variables* but they are actually place holders. From now on we will stick to this CCS notation by referring to process constants identifiers as variables.

The operational semantics for the rec operator is:

$$E\{\text{rec } X.E/X\}\xrightarrow{\mu}E' \text{ implies } \text{rec } X.E\xrightarrow{\mu}E'$$

where $E\{F/X\}$ denotes the result of substituting F for each *free* occurrence (i.e. not bound by rec) of X in E, renaming bound variables as necessary. For any expression E, FreeVar(E) denotes the set of free variables in E.

A *free occurrence* of X in E is *guarded* if it occurs within some subexpression $\mu.F$ with $\mu\neq\tau$ of E. The *variable* X is *guarded* in E if every free occurrence of X in E is guarded, otherwise X is *unguarded* in E. A recursive expression rec $X.E$ is *guarded* if X is guarded in E. An expression E is *guarded* if every recursive subexpression of E is guarded.

In the following, we deal with the subclass $\mathscr{E}_g \subset \mathscr{T}(\mathscr{R}, \mathscr{X})$ of guarded recursive *closed* (i.e. every variable is bound to a rec operator) CCS expressions.

A correct and complete axiomatization OBSREC_g for observational congruence over $\mathscr{E}_g$ has been given in [17] by adding the following axioms for recursion to OBS:

U1. $\text{rec}\, X \,.\, E = E\{\text{rec}\, X \,.\, E/X\}$

U2. $F = \text{rec}\, X \,.\, E$ if $F = E\{F/X\}$, provided X is guarded in E.

Note that U1 and U2 are actually schematizations of infinitely many first-order equations.

Differently from OBS, the proof of the completeness of OBSREC_g with respect to observational congruence does not resort to an explicit definition of "recursive normal form" over $\mathscr{E}_g$. In our study for a notion of normal form for terms in $\mathscr{E}_g$ with respect to a rewriting relation equivalent to OBSREC_g we need different characterizations for recursive terms, such as sets of recursive equations and process graphs.

2.3. *A canonical transformation over recursive expressions*

In this section we address the problem of deciding if two CCS recursive terms in $\mathscr{E}_g$ can be rewritten into the same infinite term. As we will see in the next sections, this problem turns out to be crucial in our rewriting framework.

Let the rule R1 be obtained by orienting the axiom U1 in the following way:

$$\text{R1} =_{\text{def}} \text{rec}\, X \,.\, E \to E\{\text{rec}\, X \,.\, E/X\}$$

Definition 2.5 (*unfolding*). A term $t' \in \mathscr{T}^{\infty}(\mathscr{P}, \mathscr{X})$ is the *unfolding* of a term $t \in \mathscr{E}_g$ if $t \to_{\text{R1}}^{\omega} t'$.

Thus, the unfolding of a recursive term is the term that can be reached by applying an infinite number of rewriting steps by $\to_{\text{R1}}$ and does not contain any further redex for $\to_{\text{R1}}$.

In general, there are a number of syntactically different terms that admit the same unfolding. Nevertheless, we can restrict our attention to a *canonical* term in the class of those terms having the same unfolding. Any CCS recursive term can be equivalently seen as a set of recursive equations; in [2] the existence of a canonical representative for the class of systems which admit the same solution in the canonical interpretation, is shown. Their notion of solution in the canonical interpretation corresponds to our notion of unfolding. Actually, we will not explicitly define the notion of canonical term, but we will take the term corresponding to the canonical system as the canonical term.

In the following, we provide an algorithm to determine the canonical system; our algorithm is derived from the one presented in [2] by extending it to deal with AC terms. The transformations from a term to a system of equations and from a system of equations to a term are informally introduced, while we refer to [20] for more details.

From a recursive CCS *term to a system of recursive equations*

Given an expression $E_i \equiv \mathrm{op}_i(E_{i1}, \ldots, E_{ik}) \in \mathscr{E}_{\mathscr{G}}$, the associated system of equations $S(X_i, E_i)$ where X_i is called the *main variable* of the system, is recursively defined as follows:

(i) $\mathrm{op}_i \neq \mathrm{rec}$

$S(X_i, E_i) =_{\mathrm{def}} \{X_i = \mathrm{op}_i(X_{i1}, \ldots, X_{ik})\} \cup \bigcup_j \{S(X_{ij}, E_{ij}) \mid E_{ij}$ is not a variable$\}$ where $X_{ij} \equiv E_{ij}$ if E_{ij} is a variable, otherwise X_{ij} is a new fresh variable, main variable of $S(X_{ij}, E_{ij})$.

(ii) $\mathrm{op}_i = \mathrm{rec}$, i.e. $E_i \equiv \mathrm{rec}\, Z \,.\, \mathrm{op}'(E_{i1}, \ldots, E_{ik})$ for some op′ in $\mathscr{R}$ of arity k.

$S(X_i, E_i) =_{\mathrm{def}} \{X_i = \mathrm{op}'(X_{i1}, \ldots, X_{ik})\} \cup \bigcup_j \{S(X_{ij}, E_{ij}\{X_i/Z\}) \mid E_{ij}$ is not a variable$\}$ where $X_{ij} \equiv E_{ij}$ if E_{ij} is a variable, otherwise X_{ij} is a new fresh variable, main variable of the system $S(X_{ij}, E_{ij}\{X_i/Z\})$.

We now informally recall the basic ideas the algorithm to determine a canonical system of recursive equations is based on. In order to obtain the canonical system $\mathrm{CS}(X, E)$ for an expression E, the system $S(X, E)$ is normalized by means of a normalization algorithm that identifies equivalent equations. Since we deal with AC operators, we have extended the original algorithm in order to cope with commutativity, while associativity is dealt with by considering the associative operators as varyadic. To this respect, we will make use of a flattening procedure to transform any term $t \in \mathscr{E}_{\mathscr{G}}$ into a term $\mathrm{flat}(t)$ as defined in Section 2.1.

The algorithm presented in [2] works on *uniform* systems, i.e. systems in which the right-hand side E_i of each equation has the form $E_i \equiv \mathrm{op}_i(X_1, \ldots, X_{k(i)})$ for some $\mathrm{op}_i \in \mathscr{R}$ of arity $k(i)$ and variables $X_1, \ldots, X_{k(i)}$. Note that our transformation from an expression to the system yields a uniform system by construction. From now on, depending on the context, we equivalently use either $\mathrm{CS}(X, E)$ or $\mathrm{CS}(E)$ or CS to denote the canonical system of an expression E.

From a system of recursive equations to the canonical system

Given a uniform system of n recursive equations $S \equiv \{X_i = \mathrm{op}_i(X_{i1}, \cdots, X_{ik(i)}) \mid 1 \leqslant i \leqslant n\}$ where $k(i)$ denotes the arity of op_i, we define an equivalence relation R on its variables such that $X_i R X_j$ if and only if the terms E_i and E_j corresponding to the two subsystems whose main variables are X_i and X_j respectively, have the same unfolding.

In order to constructively characterize the relation R, let us now inductively define an increasing sequence of partitions on $V_S \times V_S$, where $V_S = \{X_i \mid 1 \leqslant i \leqslant n\}$:

1. $D_0 = \{(X_i, X_j) \in V_S \times V_S \mid \mathrm{op}_i \neq \mathrm{op}_j \vee (\mathrm{op}_i = \mathrm{op}_j \wedge \mathrm{op}_i$ is associative $\wedge\ k(i) \neq k(j)\}$
2. $D_{n+1} = D_n \cup \{(X_i, X_j) \in V_S \times V_S \mid \mathrm{op}_i = \mathrm{op}_j \wedge \mathrm{op}_i$ is commutative $\wedge\ k(i) = k(j) \wedge$ for every permutation Π of $\{1, \ldots, k(j)\}\ \exists m \in [1, k(j)]$ s.t. $(X_{im}, X_{j\Pi(m)}) \in D_n\} \cup \{(X_i, X_j) \in V_S \times V_S \mid \mathrm{op}_i = \mathrm{op}_j \wedge \mathrm{op}_i$ is not commutative $\wedge\ \exists m \in [1, k(i)]$ s.t. $(X_{im}, X_{jm}) \in D_n\}$.

In this way any partition of the sequence contains pairs of variables (X_i, X_j) such that $X_j \not R X_j$.

We can now extend the result given in [2] to our AC version of their algorithm.

Lemma 2.6. (i) *There exists an index r such that* $D_r = \bigcup_{n=0}^{\infty} D_n$; (ii) $X_i R X_j$ *if and only if* $(X_i, X_j) \notin D_r$.

Proof. (i) It trivially derives from the fact that V_S is finite.

(ii) First half: if $X_i R X_j$ then $(X_i, X_j) \notin D_r$. By contradiction, let us assume $(X_i, X_j) \in D_r$. This means that either $\mathrm{op}_i \neq \mathrm{op}_j \vee (\mathrm{op}_i = \mathrm{op}_j \wedge \mathrm{op}_i$ is associative $\wedge\ k(i) \neq k(j))$, or X_i and X_j refer to subterms which are structurally different, thus contradicting the hypothesis $X_i R X_j$ that the associated terms have the same unfolding.

Second half: if $(X_i, X_j) \notin D_r$ then $X_i R X_j$. By contradiction, let us assume that $X_i \not R X_j$. This means that the two corresponding subterms are structurally different, i.e. they are different with respect to (at least) a subterm. Let t and t' be the two different subterms in E_i and E_j, respectively. By case analysis it is easy to see that any structural difference, apart from those related to the associativity and commutativity of the "+" operator, leads to a contradiction of the hypothesis. □

Let us now state the following.

Theorem 2.7. *Given* $E1, E2 \in \mathscr{E}_g$, *it is decidable if* $E1$ *and* $E2$ *admit the same unfolding modulo associativity and commutativity of the* "+" *operator.*

Proof. Let S1, S2 be the systems of equations built from flat($E1$), flat($E2$), respectively. The algorithm above can be applied to compute the canonical systems CS1, CS2 corresponding to S1 and S2, respectively. Since it is always possible to assume the two sets of variables V_{CS1} and V_{CS2} to be disjoint, we can consider the system $\mathrm{CS} = \mathrm{CS1} \cup \mathrm{CS2}$ and apply the algorithm above in order to decide if $X\ R\ Y$, where X and Y are the main variables of CS1 and CS2, respectively. □

From a system of recursive equations to the CCS *term*

Given a system of recursive equations $S \equiv \{X_i = \mathrm{op}_i(X_{i1}, \ldots, X_{ik(i)}) \mid 1 \leqslant i \leqslant n\}$, let $E'(S)$ be the expression resulting from the following transformation:

$E'(S) = \mathrm{Expr}(X_1, \emptyset)$ where X_1 is the main variable of S and

$$\mathrm{Expr}(X_i, \mathrm{env}) = \begin{cases} \mathrm{rec}\, X_i . \mathrm{op}_i(\mathrm{Expr}(X_{i1}, \mathrm{env} \cup \{X_i\}), \ldots, \mathrm{Expr}(X_{ik(i)}, \mathrm{env} \cup \{X_i\})) \\ \quad \text{if } k(i) \neq 0 \wedge X_i \notin \mathrm{env}, \\ X_i \quad \text{if } k(i) \neq 0 \wedge X_i \in \mathrm{env}, \\ \mathrm{op}_i \quad \text{if } k(i) = 0. \end{cases}$$

The expression $E(S)$ denoted by a system S can be obtained by eliminating the superfluos rec operators from $E'(S)$ using the equivalence:

$\mathrm{rec}\, X . E = E$ if E does not contain any free occurrence of X.

It is easy to show that the above axiom can be derived from OBSREC_g by applying U1.

Example 2.8. For simplicity the usual infix notation for CCS terms is used in this example. Given the expression $E \equiv \text{a.rec}X.((a.X+b.\text{nil})+c.\text{nil})$, the flattened version is $\text{flat}(E)=a.\text{rec}X.(a.X+b.\text{nil}+c.\text{nil})$. The system $S(X_1,E)$ is the following:

$$\{X_1=a.X_2,\ X_2=X_3+X_4+X_5,\ X_3=a.X_2,\ X_4=b.X_6,\ X_5=c.X_7,\ X_6=\text{nil},\ X_7=\text{nil}\}.$$

The normalization algorithm finds out that the equations for X_1 and X_3 are equal, besides the equality between the equations for X_6 and X_7. The resulting canonical system $\text{CS}(X_1,E)$ is:

$$\{X_1=a.X_2,\ X_2=X_1+X_4+X_5,\ X_4=b.X_6,\ X_5=c.X_6,\ X_6=\text{nil}\}.$$

The corresponding expression is $\text{rec}X_1.(a.(\text{rec}X_2.X_1+\text{rec}X_4.b.\text{nil}+\text{rec}X_5.c.\text{nil}))$ from which those "$\text{rec}X_i$." operators whose body is constant with respect to the variable X_i can be eliminated. The resulting expression $\text{rec}X_1.a.(X_1+b.\text{nil}+c.\text{nil})$ is the canonical representative for E.

Thus, we can decide whether two recursive expressions admit the same unfolding. Two terms $t1$, $t2$ will be equivalent modulo CT (from *canonical transformation*) if and only if they admit the same unfolding. In the following, we refer to this congruence relation as $=_{\text{CT}}$ and the application of a rewriting relation $\rightarrow_R$ modulo $=_{\text{CT}}$ means that $t \rightarrow_{R,\text{CT}} s$ if there exist a rule $l \rightarrow r$ in R, a substitution σ and a subterm $t|_u$ at the position u, such that $t|_u =_{\text{CT}} \sigma l$ and $s=t[\sigma r]_u$. Note that $\rightarrow_{R,\text{CT}}$ is defined as an extended rewrite relation, see for example [5].

2.4. *Normal process graphs*

The definitions and results reported in this section will not be used in the definition of the rewriting relation for OBSREC_g, but will be necessary when proving its completeness with respect to OBSREC_g.

CCS terms can always be represented by means of graphs and behavioural equivalences can also be defined on such graphs. In [1], a characterization of the kind of transformations necessary to obtain the unique normal graph with respect to observational congruence is defined. We assume that the reader is familiar with graph theory and only recall some relevant notions and results on graphs taken from [1].

The considered graphs are *connected, rooted multidigraphs*: any graph has a *root* (starting node), the edges between the nodes are directed and between two nodes there may be several edges, every node is accessible from the root. A *path* π in a graph g is an alternating sequence of nodes and edges, π: $s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_n$ for $n \geqslant 0$. The *length* of the path is n; if $n \geqslant 1$ and s_0 and s_n coincide, π is a *cycle*. If $n=1$ and s_0 and s_n coincide, π is a *loop*. If s is lying on a cycle, it is called *cyclic*, otherwise *acyclic*. If s is a node of g, the *subgraph* $(g)_s$ of g is the graph with root s and all the nodes and edges accessible from s. Graphs differing only in their naming of the nodes are considered to be identical.

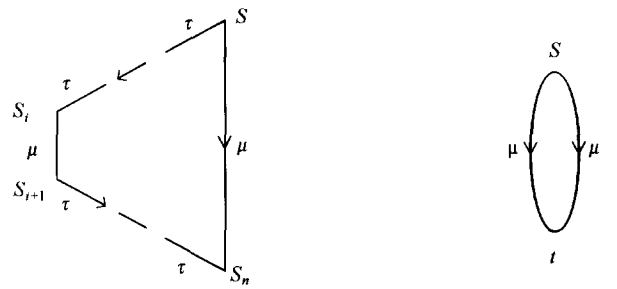


Fig. 1.

A *process graph* is a graph whose edges are labelled with actions from a set $\Lambda \cup \{\tau\}$ ranged over by μ. Given a graph g, let Root(g) and Nodes(g) denote the root and the set of nodes of g, respectively. We recall from Section 2.2 that $s \stackrel{\mu}{\Rightarrow} t$ if $s \stackrel{\tau^*}{\rightarrow} \stackrel{\mu}{\rightarrow} \stackrel{\tau^*}{\rightarrow} t$. Let g, h be any process graphs with acyclic root. It is possible to define a notion of bisimulation on process graphs in the following way. The relation R on Nodes(g) × Nodes(h) is a *τ-bisimulation* from g to h, and g, h are *τ-bisimilar*, if:

(i) Domain(R) = Nodes(g) and Range(R) = Nodes(h);

(ii) (Root(g), Root(h))∈R;

(iii) if $(s,t)\in R$ and $s \stackrel{\mu}{\Rightarrow} s'$, then there exists t' such that $t \stackrel{\mu}{\Rightarrow} t'$ and $(s',t')\in R$;

(iv) if $(s,t)\in R$ and $t \stackrel{\mu}{\Rightarrow} t'$, then there exists s' such that $s \stackrel{\mu}{\Rightarrow} s'$ and $(s',t')\in R$.

R is an *rτ-bisimulation* if $(s,t)\in R$ implies s = Root(g) and t = Root(h), or $s \neq$ Root(g) and $t \neq$ Root(h). R is called *(r)τ-autobisimulation* of g if it is a (r)τ-bisimulation from g to itself. For finite process graphs the *rτ-bisimulation* coincides with observational congruence.

A *τ-loop* is a loop $s \stackrel{\tau}{\rightarrow} s$. Given a process graph g, an *arc* in g is a subgraph $(g)_s$ such that there exist two paths starting from s: $s \stackrel{\tau}{\rightarrow} s_1 \stackrel{\tau}{\rightarrow} \cdots \stackrel{\tau}{\rightarrow} s_i \stackrel{\mu}{\rightarrow} s_{i+1} \stackrel{\tau}{\rightarrow} \cdots \stackrel{\tau}{\rightarrow} s_n$ and $s \stackrel{\mu}{\rightarrow} s_n$. A *double edge* is a particular arc where the two paths are the same of length 1, i.e. $s \stackrel{\mu}{\rightarrow} t$. Fig. 1 exemplifies the two situations.

Note that the notions of arc and double edge characterize, at graph level, the same situation that is captured by the absorption lemma at term level. That is, they identify those portions of a graph which are redundant with respect to observational congruence. In addition to this, when considering finite-state process graphs we have to cope with another source of redundancy. It is in fact possible for a process graph to contain a bisimilar subgraph. In order to identify this situation the notion of *rτ*-rigidity is introduced.

A process graph g with acyclic root is *rτ-rigid* if it has only the trivial *rτ*-autobisimulation, i.e. the identity relation. A process graph g with acyclic root is *minimal* if g contains no double edges, no τ-loops and no arcs. A process graph g is *rτ-normal* if it is *rτ*-rigid and minimal. The following two theorems guarantee that it is possible to rely on the above notion of normality.

Theorem 2.9 (Bergstra and Klop [1, Theorem 3.2.2]). *Let g, h be $r\tau$-normal and $r\tau$-bisimilar process graphs. Then g and h are identical.*

Corollary 2.10 (Bergstra and Klop [1, Corollary 3.2.3]). *Let g be a process graph with acyclic root. There is then a unique process graph g' with acyclic root such that g' is $r\tau$-normal and g, g' are $r\tau$-bisimilar.*

Note that when talking about the process graph associated to a term t, this graph will be derived from the canonical system CS(t). In this way, it is possible to define a transformation between terms and process graphs, which yields the graph with the minimum number of states (apart from the nil nodes), among all the graphs representing the same term. On the contrary, if the graph is built from the term representation of a canonical system, it may well be that the resulting graph is not the smallest one. This is due to the inadequacy of the μ-calculus in expressing *horizontal sharing* (see, e.g., [22, 23]).

From a system of recursive equations to the process graph

The transformation *process_graph* from a system $S \equiv \{X_i = \mathrm{op}_i(X_{i1}, \ldots, X_{ik(i)}) \mid 1 \leqslant i \leqslant n\}$ of equations to a process graph g can be defined as a function which builds the graph starting from the first equation. An environment *env* allows the nodes already built in the current graph to be taken into account:

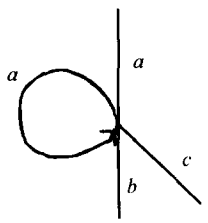
$$\text{process\_graph}(S) =_{\text{def}} \text{graph}(X_1 = E_1, \emptyset)$$

where X_1 is the main variable of S and the function *graph* is defined as follows:

```
graph (X_i = E_i, env) =_def
  if X_i = nil then create node N_i;
  if X_i = μ.X_j
    then begin
      if X_i ∉ env then create node N_i;
      if X_j ∉ env
        then N_i --μ--> graph (X_j = E_j, env ∪ {X_i})
        else N_i --μ--> N_j
      end;
  if X_i = X_i1 + ··· X_ir
    then begin
      create node N_i;
      graph (X_i = E_i1, env ∪ {X_i});
      ...
      graph (X_i = E_ir, env ∪ {X_i});
      end;
```

Note that $X_i = E_i$ need not be an equation in S. Summarizing we can say that
- to each variable X_i in S, which denotes a prefix ($X_i = \mu . X_j$), corresponds a node N_i in g;
- an edge labelled μ from a node N_i to a node N_j exists in g if $X_i = \mu . X_j$ is an equation in S, for some action prefix operator μ.;
- a node N_i in g is the common root of r subgraphs $g_1, \ldots, g_r$ (N_i has r successors) if $X_i = X_1 + \cdots + X_r$ is an equation in S and $g_1, \ldots, g_r$ are the process graphs associated to $X_1, \ldots, X_r$, respectively.

Example 2.11. Let us consider the canonical system CS(E) in Example 2.8 and build the corresponding process graph according to the transformation above. We obtain the following process graph:



3. The rewriting relation $\rightarrow_{\text{r\_obs}}$ over $\mathscr{E}_g$

Given $\text{OBSREC}_g = \text{OBS} \cup \{\text{U1}, \text{U2}\}$, let us consider the two axioms U1 and U2 and how the rewriting relation $\rightarrow_{\text{f\_obs}}$ for finitary CCS can be extended to decide the observational congruence over $\mathscr{E}_g$. Note that in the following, when working on recursive expressions, the equivalence $=$ is meant to be modulo renaming of the variables (for example, $\text{rec}\, X . a . X + \tau . \text{rec}\, Y . a . Y \rightarrow_{\text{f\_obs}} \tau . \text{rec}\, Y . a . Y$).

Let us first consider the axiom U1. The rule R1 as defined in Section 2.3 leads to a nonterminating rewriting relation, and we cope with the problem of non-terminating rewritings in the framework of ω-rewriting and ω-normal forms.

Let us now consider the axiom U2: $F = \text{rec}\, X . E$ if $F = E\{F/X\}$, provided X is guarded in E. In our rewriting framework, we replace it with a more convenient rule by specializing its application patterns. The axiom U2 says that an expression F has to be observational congruent to an expression E containing F itself as a subexpression, and obviously this cannot be the case for finite trees. We replace U2 with the following axiom CE (*collapsing equivalence*):

$\text{CE} =_{\text{def}} \text{rec}\, X . E = \text{rec}\, X . E'\{X/F\}$
if $\text{rec}\, X . E'\{X/F\} = F\{Y/X\}$, $\text{FreeVar}(E) = \text{FreeVar}(F\{Y/X\})$, where $E|_u = \text{rec}\, Y . F'$ for some u, Y, F', $E \xrightarrow{*}_{\text{R1}} E'$ by applying R1 on $\text{rec}\, Y . F'$, $E'|_v = F$ for some v, such that $\text{rec}\, Y . F' \xrightarrow{*}_{\text{R1}} F$.

The collapsing equivalence restricts the range of application of U2 and removes bisimilar nodes other than those obtained by unfolding.

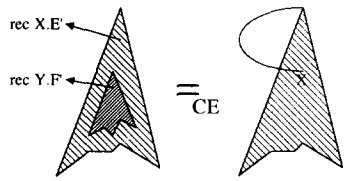


Fig. 2. The collapsing equivalence

Note that it is necessary to check the equivalence $\mathrm{rec}\,X.E'\{X/F\}=F(Y/X\}$ in the applicability condition of CE, in order to capture all the cases in which CE is applicable. Let us consider, for example, the term $\mathrm{rec}\,X.(a.X+\mathrm{rec}\,Y.a.Y)$. In order to apply CE it is necessary to consider as $\mathrm{rec}\,X.E'\{X/F\}$ the expression $\mathrm{rec}\,X.(a.X+a.\mathrm{rec}\,Y.a.Y)\{X/\mathrm{rec}\,Y.a.Y\}$, in which an unfolding of the internal rec expression has been performed. Instead, for $\mathrm{rec}\,X.a1.(a2.X+a2.a1.a2.\mathrm{rec}\,Y.a1.a2.Y)$, $\mathrm{rec}\,X.E'\{X/F\}$ is the expression $\mathrm{rec}\,X.a1.(a2.X+a2.a1.a2.\mathrm{rec}\,Y.a1.a2.Y)\{X/a1.a2.\mathrm{rec}\,Y.a1.a2.Y\}$ where F is the expression $a1.a2.\mathrm{rec}\,Y.a1.a2.Y$, which is obtained by unfolding the recursive subexpression $\mathrm{rec}\,Y.a1.a2.Y$. Note that the application of the CE axiom can always be decided, given that for every term t, there exist a maximum number of unfolding steps with $\rightarrow_{R1}$ to be performed in order to check for the existence of the F expression to be folded.

Proposition 3.1 (correctness of CE with respect to OBSREC_g). $\mathrm{OBSREC}_g \vdash \mathrm{CE}$.

Proof. We only consider the case of $E'\equiv E$ and $F\equiv \mathrm{rec}\,Y.F'$, the correctness of the other cases trivially follows from the correctness of R1. In order to prove the correctness of CE in OBSREC_g it is more convenient to reformulate CE as follows:

$\mathrm{CE}=_{\mathrm{def}} F\{Y/X\}=\mathrm{rec}\,X.E[F]$

if $\mathrm{rec}\,X.E[F]\{X/F\}=F\{Y/X\}$ where $F\equiv\mathrm{rec}\,Y.F'$ is a subexpression of E and $\mathrm{FreeVar}(E)=\mathrm{FreeVar}(F\{Y/X\})$.

Let us first prove the case in which the recursive expression F does not contain any free occurrences of the bound variable X of the recursive expression E. In this case CE becomes the following axiom:

$\mathrm{CE1}=_{\mathrm{def}}\mathrm{rec}\,Y.F'=\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F']$

if $\mathrm{rec}\,Y.F'=\mathrm{rec}\,X.E\{X/\mathrm{rec}\,Y.F'\}$ $\mathrm{FreeVar}(\mathrm{rec}\,X.E)=\mathrm{FreeVar}(\mathrm{rec}\,Y.F')$.

The correctness of CE1 can be proved starting from its hypothesis, by using U1 and U2:

$\mathrm{rec}\,Y.F'=\mathrm{rec}\,X.E\{X/\mathrm{rec}\,Y.F'\}$

$=_{\mathrm{U1}} E'\{X/\mathrm{rec}\,Y.F'\}\{\mathrm{rec}\,X.E\{X/\mathrm{rec}\,Y.F'\}/X\}$

$=E\{X/\mathrm{rec}\,Y.F'\}\{\mathrm{rec}\,Y.F'/X\}$ (by hypothesis of CE1)

$=E\{\mathrm{rec}\,Y.F'/X\}$ (by composing the substitutions)

$=\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F']$ (by applying U2.)

Let us now prove the general case in which F may contain free occurrences of the bound variable X. The correctness is shown by the following proof steps as above:

$$\begin{aligned}
&\operatorname{rec} Y.F'\{Y/X\}\\
&\quad = \operatorname{rec} X.E\{X/\operatorname{rec} Y.F'\} \quad \text{(by hypothesis of CE)}\\
&\quad =_{U1} E\{X/\operatorname{rec} Y.F'\}\{\operatorname{rec} X.E\{X/\operatorname{rec} Y.F'\}/X\}\\
&\quad = E\{X/\operatorname{rec} Y.F'\}\{\operatorname{rec} Y.F'\{Y/X\}/X\} \text{ (by hypothesis of CE)}\\
&\quad = E\{\operatorname{rec} Y.F'\{Y/X\}/\operatorname{rec} Y.F', \operatorname{rec} Y.F'\{Y/X\}/X\}\\
&\quad \text{(by composing the substitutions)}\\
&\quad =_{U2} \operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}].
\end{aligned}$$

Thus we have obtained:

$$\operatorname{rec} Y.F'\{Y/X\} = \operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}] \qquad (*)$$

At this point, we have to show that $\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}] = \operatorname{rec} X.E[\operatorname{rec} Y.F']$. We prove this by applying U2, thus we have to show that its hypothesis holds:

$$\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}] \triangleq E[\operatorname{rec} Y.F']\{\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]/X\} \qquad (**)$$

where the U2-F is $\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]$, and the U2-E is $E[\operatorname{rec} Y.F']$.

Since the occurrence of X inside F' is involved in the substitution, we can rewrite the right-hand side of (**) as follows:

$$\begin{aligned}
&\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]\\
&\quad \triangleq E[\operatorname{rec} Y.F'[\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]]]\{\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]/X\}
\end{aligned}$$

and by applying U1 on the left hand side we get:

$$\begin{aligned}
&E[\operatorname{rec} Y.F'\{Y/X\}]\{\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]/X\}\\
&\quad \triangleq E[\operatorname{rec} Y.F'[\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]]]\{\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]/X\}
\end{aligned}$$

which reduces to prove

$$\operatorname{rec} Y.F'\{Y/X\} \triangleq \operatorname{rec} Y.F'[\operatorname{rec} X.E[\operatorname{rec} Y.F'\{Y/X\}]] \qquad \text{(i)}$$

We rewrite the left-hand side as follows:

$$\begin{aligned}
\operatorname{rec} Y.F'\{Y/X\} &=_{U1} F'\{Y/X\}\{\operatorname{rec} Y.F'\{Y/X\}/Y\}\\
&= F'\{\operatorname{rec} Y.F'\{Y/X\}/X, \operatorname{rec} Y.F'\{Y/X\}/Y\}\\
&\quad \text{(by composing the substitutions)}\\
&=_{U2} \operatorname{rec} Y.F'[\operatorname{rec} Y.F'\{Y/X\}]
\end{aligned}$$

Then, since the right-hand side of (i) is equivalent to $\operatorname{rec} Y.F'[\operatorname{rec} Y.F'\{Y/X\}]$ by (*), the thesis is proved.

The axiom CE is turned into the following rule CR (*collapsing rule*):

$\mathrm{CR} =_{\mathrm{def}} \mathrm{rec}\,X\,.\,E = \mathrm{rec}\,X\,.\,E'\{X/F\}$
if $\mathrm{rec}\,X\,.\,E'\{X/F\} = F\{Y/X\}$, $\mathrm{FreeVar}(E) = \mathrm{FreeVar}(F\{Y/X\})$, where $E|_u = \mathrm{rec}\,Y\,.\,F'$ for some u, Y, F', $E \xrightarrow{*}_{\mathrm{R1}} E'$ by applying R1 on $\mathrm{rec}\,Y\,.\,F'$, $E'|_v = F$ for some v, such that $\mathrm{rec}\,Y\,.\,F' \xrightarrow{*}_{\mathrm{R1}} F$.

Example 3.2. The expression $E \equiv \mathrm{rec}\,X\,.\,a\,.\,((X + b\,.\,\mathrm{nil}) + \mathrm{rec}\,Y\,.\,a\,.\,(Y + b\,.\,\mathrm{nil}))$ can be rewritten as follows: $\mathrm{rec}\,X\,.\,a\,.\,((X + b\,.\,\mathrm{nil}) + \mathrm{rec}\,Y\,.\,a\,.\,(Y + b\,.\,\mathrm{nil})) \rightarrow_{\mathrm{CR}} \mathrm{rec}\,X\,.\,a\,.\,((X + b\,.\,\mathrm{nil}) + X) \rightarrow_{\mathrm{f\_obs}} \mathrm{rec}\,X\,.\,a\,.\,(X + b\,.\,\mathrm{nil})$.

Given the framework of ω-rewritings, our aim is to characterize infinite rewritings in such a way that their limit exists, are ω-normal form and are obtained by applying R1 infinitely many times after a finite number of reductions. This implies that any redex other than those for R1 has to be reduced along the derivation and an infinite generation of new redexes has to be avoided. Let us consider simple terms like $t \equiv \mathrm{rec}\,X\,.\,\tau\,.\,(a\,.\,X + t')$ for some term t', i.e. generic terms $\mathrm{rec}\,X\,.\,\tau\,.\,E$, where E contains directly prefixed occurrences of X. After the first rewriting step by means of R1, any further unfolding step generates a new redex for $\rightarrow_{\mathrm{f\_obs}}$ (in particular, for the rule $\mu\,.\,\tau\,.\,\mathrm{E} \rightarrow \mu\,.\,E$). In order to cope with this situation we introduce the following axiom, *action prefix equivalence*, which we will refer to as $\mathrm{A_pE}$:

$$\mathrm{A_p}E =_{\mathrm{def}} \mathrm{rec}\,X\,.\,\tau\,.\,E = \tau\,.\,\mathrm{rec}\,X\,.\,E\{\tau\,.\,X/X\}$$

Proposition 3.3 (correctness of $\mathrm{A_pE}$ with respect to $\mathrm{OBSREC_g}$). $\mathrm{OBSREC_g} \vdash \mathrm{A_p}E$.

Proof. The correctness of $\mathrm{A_pE}$ with respect to the axiomatization $\mathrm{OBSREC_g}$ can be proved by first applying U1,

$$\begin{aligned}\tau\,.\,\mathrm{rec}\,X\,.\,E\{\tau\,.\,X/X\} &=_{\mathrm{U1}} \tau\,.\,E\{\tau\,.\,X/X\}\{(\mathrm{rec}\,X\,.\,E\{\tau\,.\,X/X\})/X\} \\ &= \tau\,.\,E\{(\tau\,.\,\mathrm{rec}\,X\,.\,E\{\tau\,.\,X/X\})/X\}\end{aligned}$$

and then using U2,

$$\tau\,.\,\mathrm{rec}\,X\,.\,E\{\tau\,.\,X/X\} = \mathrm{rec}\,X\,.\,\tau\,.\,E. \qquad \square$$

The rule $\mathrm{A_p}$ is defined by orienting the axiom $\mathrm{A_p}E$ from left to right:

$$\mathrm{A_p} =_{\mathrm{def}} \mathrm{rec}\,X\,.\,\tau\,.\,E \rightarrow \tau\,.\,\mathrm{rec}\,X\,.\,E\{\tau\,.\,X/X\}$$

Note that further infinitely reducible combinations between a rec body and its external context do not exist, since the guardedness hypothesis implies that a redex for $\rightarrow_{\mathrm{f\_obs}} \cup \rightarrow_{\mathrm{CR}} \cup \rightarrow_{\mathrm{A_p}}$ can arise, as a result of such a combination, only after a finite number of rewritings by $\rightarrow_{\mathrm{R1}}$. The guardedness hypothesis guarantees that possible redexes for the absorption lemma can only occur after a finite number of unfoldings and cannot be produced infinitely many times.

Let us now introduce the rewriting relation $\rightarrow_{r\_obs}$:

$$\rightarrow_{r\_obs} =_{def} \rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p} \cup \rightarrow_{R1}$$

The rewriting rules in $\rightarrow_{r\_obs}$ are characterized by the following properties:

- $\rightarrow_{f\_obs,CT}$ reduces either inside the rec body or by considering the rec term as a whole; this means that redexes for $\rightarrow_{f\_obs,CT}$ cannot involve subterms of a rec term *and* its external context;
- $\rightarrow_{CR}$ reduces a recursive expression $\mathrm{rec}\,X.E$ by replacing an internal recursive term with X. To be applied, $\rightarrow_{CR}$ checks subterms for equivalence using $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$, but $\rightarrow_{CR}$ does not apply such possible reductions. Thus, the expression resulting from the application of $\rightarrow_{CR}$ can still be reducible according to $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$;
- after a rewriting step by $\rightarrow_{A_p}$, redexes for $\rightarrow_{f\_obs,CT}$ can occur in $\tau.\mathrm{rec}\,X.E\{\tau.X/X\}$. This is the case when directly prefixed occurrences of the variable X occur in the body E.

The following proposition sheds light on the interactions between R1, the rule that leads to the limit of a derivation, and redexes for $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$.

Proposition 3.4. *Given $E \in \mathcal{E}_g$, let the subexpression $G = E|_u$ be a redex for $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$. If $E \rightarrow_{R1} E'$ at the redex $\mathrm{rec}\,X.F$ in E, then:*

(i) *if G and $\mathrm{rec}\,X.F$ occur at independent positions or $\mathrm{rec}\,X.F$ is a subterm of G, then G still occurs in E'* (modulo $=_{CT}$);

(ii) *if G occurs in F, $\rightarrow_{R1}$ produces as many new redexes $E'|_{u_j}$ for $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$ in E' as the number of the occurrences of X in F. Moreover, $G\{\mathrm{rec}\,X.F/X\}|_{u'}$ is a redex in E' for some position u' which is prefix of u_j for each j;*

(iii) *if $G = \mathrm{rec}\,X.F$, i.e. it is a redex for $\rightarrow_{CR}$ or $\rightarrow_{A_p}$, then $\rightarrow_{R1}$ produces as many new redexes $E'|_{u_j}$ in E' as the number of the occurrences of X in F.*

Proof. It follows from the definition of the rule R1 and the above properties of $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$. □

In the following, we will first show that $\rightarrow_{r\_obs}$ is an ω-canonical rewriting relation over $\mathcal{E}_g$ and then that it is complete with respect to OBSREC_g.

4. ω-canonicity of $\rightarrow_{r\_obs}$

In order to show that the rewriting relation $\rightarrow_{r\_obs}$ is ω-canonical over $\mathcal{E}_g$, we have to prove that $\rightarrow_{r\_obs}$ is ω-converging and ω-confluent. Let us first consider the termination issues.

4.1. Top-termination and ω-convergence of $\rightarrow_{r\_obs}$

Proposition 4.1. *The rewriting relation $\rightarrow_{r\_obs}$ is top-terminating over $\mathcal{E}_g$.*

Proof. The relation $\rightarrow_{\text{f\_obs,CT}} \cup \rightarrow_{\text{CR}} \cup \rightarrow_{\text{A}_\text{p}}$ is terminating and top-terminating over $\mathscr{E}\mathscr{g}$: given any term $t \in \mathscr{E}\mathscr{g}$, there exists a finite number of rewriting steps by $\rightarrow_{\text{f\_obs,CT}} \cup \rightarrow_{\text{CR}} \cup \rightarrow_{\text{A}_\text{p}}$ for any position, included the topmost one. Since terms are finite, R1 can be applied only a finite number of times at the topmost position, e.g. $\text{rec}\, X_1 . \text{rec}\, X_2 \cdots \text{rec}\, X_n . E$. Moreover, since terms are guarded, infinitely many applications of R1 cannot generate infinitely many redexes for $\rightarrow_{\text{r\_obs}}$ at the topmost position. It follows that $\rightarrow_{\text{r\_obs}}$ is top-terminating over $\mathscr{E}\mathscr{g}$. □

Proposition 4.2. *The rewriting relation* $\rightarrow_{\text{r\_obs}}$ *is* ω*-converging over* $\mathscr{E}\mathscr{g}$.

Proof. It follows from Theorem 2.1, since $\rightarrow_{\text{r\_obs}}$ is top-terminating over $\mathscr{E}\mathscr{g}$ by Proposition 4.1. □

4.2. Structured fair derivations

We now show that we can restrict our considerations to (structured) fair derivations as they compute ω-normal forms at the limit. We will prove a result analogous to Theorem 2.2, even if $\rightarrow_{\text{r\_obs}}$ is not left linear. This is possible because reductions are applied modulo $=_{\text{CT}}$.

Proposition 4.3. *Given* $\rightarrow_{\text{r\_obs}}$, *if a term* $t_0 \in \mathscr{E}\mathscr{g}$ *admits an* ω*-normal form* $t' \in \mathscr{T}^\infty(\mathscr{P}, \mathscr{X})$, *then there exists a fair derivation* $t_0 \rightarrow_{\text{r\_obs}} t_1 \rightarrow_{\text{r\_obs}} \cdots \rightarrow_{\text{r\_obs}} t_n \rightarrow_{\text{r\_obs}} \cdots$ *with* $\lim_{n\to\infty} t_n = t'$.

Proof. The proof is similar to the one for Theorem 4.3 in Dershowitz et al. [6]. Given a non-fair derivation with an ω-normal form as the limit, by definition of $\rightarrow_{\text{r\_obs}}$ it is possible to build a fair derivation with the same limit. Suppose that D: $t_0 \rightarrow_{\text{r\_obs}} t_1 \rightarrow_{\text{r\_obs}} \cdots \rightarrow_{\text{r\_obs}} t_n \rightarrow_{\text{r\_obs}} \cdots \rightarrow^{\omega}_{\text{r\_obs}} t'$ and t' is an ω-normal form. If the derivation is not fair, then for some index N', position u and rule r in $\rightarrow_{\text{r\_obs}}$, the rule r must be continually applicable at u in the subderivation $(t_n)_{n \geqslant N'}$, though not actually applied. Let $N \geqslant N'$ be an index such that for all $n \geqslant N$, we have $d(t_n, t') \leqslant 1/2^{|u|+|R|+|R|}$. Let t'_n denote the result of applying r to t_n at u. On account of the low positions of reductions, any changes incurred by the steps past N take place in the variable part of r. The situation is the following:

$$
\begin{array}{lccccccc}
D: & t_0 \rightarrow_{\text{r\_obs}} t_1 & \rightarrow_{\text{r\_obs}} \cdots & \rightarrow_{\text{r\_obs}} t_N & \rightarrow_{\text{r\_obs}} t_{N+1} & \rightarrow_{\text{r\_obs}} \cdots & \rightarrow^{\omega}_{\text{r\_obs}} t' \\
 & \downarrow r & & \downarrow r & & & \downarrow r \\
 & t'_N & & t'_{N+1} & & & t'
\end{array}
$$

The rule r is continually applicable at u on t_n for $n \geqslant N$, though not actually applied. The same rule also applies to t', but since t' is an ω-normal form, it must be that the result of rewriting t' is t' itself. In order to build a fair derivation from D, we have to

mimic D by applying the rule r and then linking the terms in the subderivation $(t'_n)_{n \geqslant N}$ with rewriting steps $t'_n \overset{*}{\to}_{\text{r\_obs}} t'_{n+1}$, $n \geqslant N$. In this way we build a derivation in $\to_{\text{r\_obs}}$ which is ω-converging by Proposition 4.2. Therefore we have only to guarantee that the limit is reached in a fair way. Let r' be the rule in $\to_{\text{r\_obs}}$ such that $t_N \to_{r'} t_{N+1}$ in D. Let us consider the following cases based on the (non-)left linearity of the rules r, r'.

(i) If r, r' are both left linear, see Theorem 4.3 in [7].

(ii) If r is non-left linear and r' is left linear, it is always possible to rewrite $t'_n \overset{*}{\to}_{\text{r\_obs}} t'_{n+1}$ and close the diagram, since r' is left linear and independent of the changes due to the application of r.

(iii) The interesting cases are when r' is non-left linear, independently of the (non-)left linearity of r. Non-left linearity means that the application of a rule requires the equality of (at least) two subterms. The application of r from t_N to t'_N may destroy the redex for r', by rewriting the equal subterms into different ones and thus resulting in the impossibility of rewriting t'_N into t'_{N+1}. In the rewriting relation $\to_{\text{r\_obs}}$, the rule r can only be R1 which rewrites subterms denoting infinite structures (otherwise the result of rewriting t' with r would not be t' itself) and the possible changes introduced by r in t'_n are taken into account when rewriting in $\to_{\text{r\_obs}}$ modulo $=_{\text{CT}}$. Thus, it is always possible to close the diagram from t'_n to t'_{n+1} with $\overset{*}{\to}_{\text{r\_obs}} (n \geqslant N)$ and the derivation from t_N can be mimicked by a derivation issuing from t'_N as follows:

$$
\begin{array}{lllll}
D: t_0 \to_{\text{r\_obs}} t_1 \to_{\text{r\_obs}} \cdots \to_{\text{r\_obs}} t_N \to_{\text{r\_obs}} t_{N+1} \to_{\text{r\_obs}} \cdots \to^{\omega}_{\text{r\_obs}} t' \\
\qquad\qquad\qquad\qquad\quad \downarrow r \qquad \downarrow r \qquad\qquad\qquad \downarrow r \\
\qquad\qquad\qquad\qquad\quad t'_N \overset{*}{\to}_{\text{r\_obs}} t'_{N+1} \overset{*}{\to}_{\text{r\_obs}} \cdots \to^{\omega}_{\text{r\_obs}} t'
\end{array}
$$

Since the same reductions are essentially applied to the terms on the subderivation $(t'_n)_{n \geqslant N}$, the distance $d(t'_n, t') \leqslant 1/2^{|u|}$ for all $n \geqslant N$ and, moreover, $\lim_{n \to \infty} t'_n = t'$. This process may be repeated starting from some $t'_{n'} (n' > N)$ such that $d(t'_{n'}, t') \leqslant 1/2^{|u|+1}$ to obtain a fair derivation with t' as the limit. □

In general, the limit of a fair derivation in $\to_{\text{r\_obs}}$ need not be an ω-normal form.

Example 4.4. Let us consider the term $t = (\text{rec } X \,.\, a \,.\, X) + \text{rec } X \,.\, (a \,.\, X + \text{nil})$. It is easy to check that t admits a fair derivation with the limit given by $a^{\omega} + a^{\omega}$ which is not an ω-normal form. □

Actually, we are able to identify a particular subclass of fair derivations which have a peculiar structure.

Definition 4.5 (structured derivation). A derivation $t_0 \to_{\text{r\_obs}} t_1 \to_{\text{r\_obs}} \cdots \to_{\text{r\_obs}} t_n \to_{\text{r\_obs}} \cdots$ is *structured* if there exists an index N such that, for all $n \geqslant N$, it can only be $t_n \to_{\text{R1}} t_{n+1}$.

Thus, for any structured derivation it is possible to single out an index N that splits the infinite derivation into a finite subderivation of terms $(t_n)_{n<N}$, in which $\rightarrow_{\text{r\_obs}}$ is applied, and an infinite subderivation of terms $(t_n)_{n\geqslant N}$, in which only $\rightarrow_{\mathbf{R1}}$ can be applied. The fair derivation in Example 4.4. is not structured because at each step the rule $E+\text{nil}\rightarrow E$ in $\rightarrow_{\text{f\_obs,CT}}$ can be applied at deeper and deeper positions.

The limit of a structured fair derivation is an ω-normal form.

Proposition 4.6. *Given* $\rightarrow_{\text{r\_obs}}$ *and a term* $t_0\in\mathscr{E}_\mathscr{G}$, *then for any structured fair derivation* $t_0\rightarrow_{\text{r\_obs}}t_1\rightarrow_{\text{r\_obs}}\cdots\rightarrow_{\text{r\_obs}}t_N\rightarrow_{\mathbf{R1}}\cdots$ *for some* $N\geqslant 0$ *with* $\lim_{n\rightarrow\infty}t_n=t'$, t' *is an* ω-*normal form of* t_0.

Proof. Let D: $t_0\rightarrow_{\text{r\_obs}}t_1\rightarrow_{\text{r\_obs}}\cdots\rightarrow_{\text{r\_obs}}t_N\rightarrow_{\mathbf{R1}}\cdots$ be a structured fair derivation for some $N\geqslant 0$ with $\lim_{n\rightarrow\infty}t_n=t'$. Suppose that t' is not an ω-normal form of t_0. Since D is a structured fair derivation, it is not possible to generate new redexes for the same rule at deeper and deeper positions infinitely many times. We have only to consider the case in which t' can be rewritten at an infinite redex by a non-left linear rule in $\rightarrow_{\text{r\_obs}}$, whose application was never possible on any of the finite terms in the subderivation $(t_n)_{n\geqslant N}$. In particular, this situation concerns $\rightarrow_{\text{f\_obs,CT}}$ and $\rightarrow_{\mathbf{CR}}$, because their application requires the equivalence of possibly syntactically different subexpressions s',s'' which denote the same infinite term, i.e. $s'=_{\text{CT}}s''$. Since the applicability of $\rightarrow_{\text{f\_obs,CT}}$ and $\rightarrow_{\mathbf{CR}}$ is checked modulo $=_{\text{CT}}$, these rules would be continually applicable in the subderivation $(t_n)_{n\geqslant N}$, thus contradicting the structured fairness of D. □

We now prove that every term in $\mathscr{E}_\mathscr{G}$ admits a structured fair derivation and that only structured fair derivations need to be considered to compute ω-normal forms.

Proposition 4.7. *Any term* $t_0\in\mathscr{E}_\mathscr{G}$ *admits a structured fair derivation in* $\rightarrow_{\text{r\_obs}}$.

Proof. Starting from t_0, a structured fair derivation D is obtained by applying $\rightarrow_{\mathbf{R1}}$ only when no redexes for $\rightarrow_{\text{f\_obs,CT}}\cup\rightarrow_{\mathbf{CR}}\cup\rightarrow_{\mathbf{A_p}}$ exist. This derivation is structured fair for some $N\geqslant 0$. In fact, $\rightarrow_{\mathbf{A_p}}$ and the guardedness hypothesis guarantee that infinitely many redexes for $\rightarrow_{\text{f\_obs,CT}}$ cannot be generated. □

Proposition 4.8. *Given* $\rightarrow_{\text{r\_obs}}$, *if a term* $t_0\in\mathscr{E}_\mathscr{G}$ *admits an* ω-*normal form* $t_\infty\in\mathscr{T}^\infty(\mathscr{P},\mathscr{X})$, *then there exists a structured fair derivation*

$$D': t_0=t'_0\rightarrow_{\text{r\_obs}}t'_1\rightarrow_{\text{r\_obs}}\cdots\rightarrow_{\text{r\_obs}}t'_N\rightarrow_{\mathbf{R1}}\cdots \text{ for some } N\geqslant 0 \text{ with } \lim_{n\rightarrow\infty}t'_n=t_\infty.$$

Proof. Since t_0 admits an ω-normal form t_∞, by Proposition 4.3 there exists a fair derivation D: $t_0\rightarrow_{\text{r\_obs}}t_1\rightarrow_{\text{r\_obs}}\cdots\rightarrow_{\text{r\_obs}}t_n\rightarrow_{\text{r\_obs}}\cdots\rightarrow^{\omega}_{\text{r\_obs}}t_\infty$ with $\lim_{n\rightarrow\infty}t_n=t_\infty$. If D is not structured, it follows from the hypotheses and Proposition 3.4 that the only

way to produce new redexes for $\rightarrow_{\text{r\_obs,CT}} \cup \rightarrow_{\text{CR}} \cup \rightarrow_{\text{A}_\text{p}}$ infinitely many times is when $\rightarrow_{\text{R1}}$ is applied to subterms $\text{rec}\,X.E$ such that:
(i) the body E is not in normal form with respect to $\rightarrow_{\text{f\_obs,CT}} \cup \rightarrow_{\text{CR}} \cup \rightarrow_{\text{A}_\text{p}}$ or
(ii) $\text{rec}\,X.E$ is also a redex for $\rightarrow_{\text{A}_\text{p}} \cup \rightarrow_{\text{CR}}$.

Let D' be the structured fair derivation from t_0 built as shown in the proof of Proposition 4.7. Since D is fair, there exists an index j such that t_j is a term $t[t''_j]$ where the context t is in normal form with respect to $\rightarrow_{\text{r\_obs}}$ and t''_j only contains redexes of the kind (i) and/or (ii) or derived from them. Then, for every t_n in D ($n \geqslant j$), there exist a term t'_n and a term t'_k in D' ($k \geqslant N$) such that $t_n\,(\rightarrow_{\text{f\_obs,CT}} \cup \rightarrow_{\text{CR}} \cup \rightarrow_{\text{A}_\text{p}})^*\, t'_n \overset{*}{\rightarrow}_{\text{R1}} t'_k$ and this holds for the limit t_∞ of D as well, i.e. $t_\infty\,(\rightarrow_{\text{f\_obs,CT}} \cup \rightarrow_{\text{CR}} \cup \rightarrow_{\text{A}_\text{p}})^*\, t'_\infty$ where t'_∞ can only be the limit t'_∞ of D'.

$$D\colon\ t_0 \rightarrow_{\text{r\_obs}} t_1 \rightarrow_{\text{r\_obs}} \cdots \rightarrow_{\text{r\_obs}} t_j \rightarrow_{\text{r\_obs}} \cdots \rightarrow_{\text{r\_obs}} t_n \rightarrow_{\text{r\_obs}} \cdots \overset{\omega}{\rightarrow}_{\text{r\_obs}} t_\infty$$

$$D'\colon\ t_0 = t'_0 \rightarrow_{\text{r\_obs}} t'_1 \rightarrow_{\text{r\_obs}} \cdots \rightarrow_{\text{r\_obs}} t'_N \rightarrow_{\text{R1}} \cdots \overset{*}{\rightarrow}_{\text{R1}} t'_k \rightarrow_{\text{R1}} \cdots \overset{\omega}{\rightarrow}_{\text{R1}} t'_\infty$$

Since t_∞ is an ω-normal form by hypothesis, this means that $t'_\infty = t_\infty$, i.e. D' is a structured fair derivation from t_0, for some $N \geqslant 0$, with limit t_∞. □

Proposition 4.9. *The rewriting relation $\rightarrow_{\text{r\_obs}}$ is ω-normalizing over $\mathscr{E}\mathscr{g}$.*

Proof. By Proposition 4.7 any term $t_0 \in \mathscr{E}\mathscr{g}$ has a structured fair derivation D for some $N \geqslant 0$. By Proposition 4.2 $\rightarrow_{\text{r\_obs}}$ is ω-converging over $\mathscr{E}\mathscr{g}$, hence D has a limit t' which is an ω-normal form of t_0 by Proposition 4.6. □

4.3. ω-confluence of $\rightarrow_{\text{r\_obs}}$

To show the ω-confluence of $\rightarrow_{\text{r\_obs}}$ with respect to structured fair derivations, we have to prove the uniqueness of ω-normal forms, i.e. every structured fair derivation from a term t computes the *same* ω-normal form. By Proposition 4.6, this means proving that any two structured fair derivations from t have the same limit (modulo AC).

Actually, provided that $\rightarrow_{\text{R1}}$ is ω-confluent, we can restrict our considerations only to the finite subderivations of structured fair derivations. That is, given a term t_0, for any two structured fair derivations D_1, D_2 with indexes $N1$, $N2$ respectively, it is sufficient to prove the confluence of their finite subderivations

$$D'_1\colon t_0 \rightarrow_{\text{r\_obs}} t_1 \rightarrow_{\text{r\_obs}} \cdots \rightarrow_{\text{r\_obs}} t_{N1} \quad \text{and}$$

$$D'_2\colon\ t_0 = t'_0 \rightarrow_{\text{r\_obs}} t'_1 \rightarrow_{\text{r\_obs}} \cdots \rightarrow_{\text{r\_obs}} t'_{N2}.$$

It follows that $\rightarrow_{\text{r\_obs}}$ can be treated as a terminating relation, thus we can prove its confluence by resorting to local confluence due to the Newman Lemma [5]. Local confluence of $\rightarrow_{\text{r\_obs}}$ is shown by analysing all the possible situations in which a term can be rewritten by two (or more) rules in $\rightarrow_{\text{r\_obs}}$ on non-independent redexes.

Proposition 4.10. *The rewriting relation* $\rightarrow_{R1}$ *is* ω*-confluent over* $\mathscr{E}_{\mathscr{G}}$.

Proof. $\rightarrow_{R1}$ is ω-confluent if whenever $s \overset{\omega}{{}_{R1}\leftarrow} t \overset{\omega}{\rightarrow_{R1}} q$, there exists t' such that $s \overset{\omega}{\rightarrow_{R1}} t' \overset{\omega}{{}_{R1}\leftarrow} q$. $\rightarrow_{R1}$ is left linear, ω-convergence of $\rightarrow_{R1}$ follows from Propositions 4.1 and 4.2, thus by Theorem 2.3 s and q are ω-normal forms. Moreover, s and q are unfoldings of t, hence by the uniqueness of solution in systems of recursive equations $s =_{AC} q$ [17]. □

Proposition 4.11. *The rewriting relation* $\rightarrow_{r\_obs}$ *is locally confluent over* $\mathscr{E}_{\mathscr{G}}$.

Proof. See Appendix B. □

Proposition 4.12. *Given* $\rightarrow_{r\_obs}$, *a term* $t_0 \in \mathscr{E}_{\mathscr{G}}$ *and the finite subderivations* D_1 *and* D_2 *of any two structured fair derivations from* t_0 *for some* N1, $N2 \geqslant 0$,

$$D_1 : t_0 \rightarrow_{r\_obs} t_1 \rightarrow_{r\_obs} \cdots \rightarrow_{r\_obs} t_{N1} \quad \text{and} \quad D_2 : t_0 = t'_0 \rightarrow_{r\_obs} t'_1 \rightarrow_{r\_obs} \cdots \rightarrow_{r\_obs} t'_{N2}$$

then $t_{N1} =_{CT} t'_{N2}$.

Proof. The thesis follows from the structured fairness of the derivations, Proposition 4.11 and the Newman Lemma. □

Proposition 4.13. *Given* $\rightarrow_{r\_obs}$, *a term* $t_0 \in \mathscr{E}_{\mathscr{G}}$ *and any two structured fair derivations* D_1: $t_0 \rightarrow_{r\_obs} t_1 \rightarrow_{r\_obs} \cdots \rightarrow_{r\_obs} t_k \cdots \rightarrow_{r\_obs} t_{N1} \rightarrow_{R1} \cdots$ *with* $\lim_{n\rightarrow\infty} t_n = t_\infty$ *and* D_2: $t_0 = t'_0 \rightarrow_{r\_obs} t'_1 \rightarrow_{r\_obs} \cdots \rightarrow_{r\_obs} t'_k \cdots \rightarrow_{r\_obs} t'_{N2} \rightarrow_{R1} \cdots$ *with* $\lim_{n\rightarrow\infty} t'_n = t'_\infty$, *then* $t_\infty =_{AC} t'_\infty$ *if and only if* $t_{N1} =_{CT} t'_{N2}$.

Proof. D_1 and D_2 are structured fair derivations, thus t_∞ and t'_∞ are ω-normal forms by Proposition 4.6. Moreover, since the derivations are structured, every possible reduction by $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$ has been applied before $N1$ in D_1 and before $N2$ in D_2, and *only* rewritings by $\rightarrow_{R1}$ can be applied after $N1$ in D_1 and after $N2$ in D_2, respectively. Therefore, t_∞ and t'_∞ are unfoldings of t_{N1} and t'_{N2}, respectively. This means that $t_\infty =_{AC} t'_\infty$ if and only if $t_{N1} =_{CT} t'_{N2}$. □

Corollary 4.14. *The rewriting relation* $\rightarrow_{r\_obs}$ *is* ω*-confluent modulo* AC.

We have shown that the rewriting relation $\rightarrow_{r\_obs}$ is ω-canonical, i.e. the ω-normal form of any term $t \in \mathscr{E}_{\mathscr{G}}$ exists and is unique (modulo AC). We are now in the position to define our notion of recursive normal form.

Definition 4.15. A term $t \in \mathscr{E}_{\mathscr{G}}$ is in *recursive normal form* if every term t' such that $t \overset{*}{\rightarrow_{R1}} t'$ is in normal form with respect to $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$.

Thus, given a term $t_0 \in \mathscr{E}_{\mathscr{G}}$ and any structured fair derivation $t_0 \rightarrow_{r\_obs} t_1 \rightarrow_{r\_obs} \cdots \rightarrow_{r\_obs} t_N \rightarrow_{R1} \cdots$ for some $N \geqslant 0$, then t_N is in recursive normal form.

Corollary 4.16. *Any term* $t \in \mathscr{E}_{\mathscr{G}}$ *admits a (unique modulo* $=_{CT}$*) recursive normal form.*

5. Completeness of $\rightarrow_{r\_obs}$

We now prove the completeness of $\rightarrow_{r\_obs}$ with respect to the axiomatization OBSREC_g for observational congruence, i.e. any two observational congruent recursive terms admit the same ω-normal form with respect to $\rightarrow_{r\_obs}$. This is proved by showing that the canonical system of recursive equations corresponding to a recursive normal form denotes an $r\tau$-normal process graph. In order to perform the proof we need some auxiliary definitions which set a suitable structure on process graphs.

Let us recall that an $r\tau$-normal process graph is (see Section 2.4):

(i) $r\tau$-rigid, i.e. it has no $r\tau$-bisimilar nodes;

(ii) minimal, i.e. it has no arcs, no double edges and no τ-loops.

Definition 5.1. (partition Π on the equations of a system S). Given a system $S=\{X_1=E_1,\ldots,X_n=E_n\}$ associated to a term $t\in\mathcal{E}_g$, the *partition* $\Pi=\{S_1,\ldots,S_k\}$ on S ($k\leqslant n$) is induced by the following relation $\mathcal{S}$: $\{X_i=E_i\}\,\mathcal{S}\,\{X_j=E_j\}$ if $X_j\in\mathcal{V}ar(E_i)$ and $X_i\in\mathcal{V}ar(E_j)$ $j\neq i$ where $\mathcal{V}ar(E)=\{X_1,\ldots,X_r\}\cup\bigcup_{j=1,\ldots,r}\mathcal{V}ar(E_j)$ if $E\equiv \text{op}(X_1,\ldots,X_r)$, $\text{op}\in\mathcal{R}$ of arity $r\geqslant 0$.

Definition 5.2 (level of an element S_i in the partition Π). Given a partition $\Pi=\{S_1,\ldots,S_k\}$ on a system $S=\{X_1=E_1,\ldots,X_n=E_n\}$, *level*$(S_i)$ is a mapping from Π to the set of integers such that:

$$\text{level}(S_i)=\begin{cases}0 & S_i=\{X_i=c_k \mid c_k \text{ constant}\}\\ n+1 & n=\max\{\text{level}(S_j)\mid S_i \text{ uses } S_j,\ j\neq i\}\end{cases}$$

where, given $S_i=\{X_{i1}=E_{i1},\ldots,X_{ik}=E_{ik}\}$, we say that S_i *uses* S_j if $\exists X_{ir}=E_{ir}$ in S_i for $1\leqslant r\leqslant k$ such that $\exists X_{jq}\in\mathcal{V}ar(E_{ir})$ and $X_{jq}=E_{jq}\in S_j$.

Definition 5.3 (distinct elements of Π). Given a partition $\Pi=\{S_1,\ldots,S_k\}$ on a system $S=\{X_1=E_1,\ldots,X_n=E_n\}$, S_i, S_j are *distinct* if S_i and S_j do not use each other.

In the following, with abuse of notation, the above notions will be used freely also when referring to graphs and main variables of a system. We will also refer to the variables of a system as the nodes of the corresponding graph.

Proposition 5.4. *Given a term $t_0\in\mathcal{E}_g$, let D: $t_0\rightarrow_{r\_obs}t_1\rightarrow_{r\_obs}\cdots\rightarrow_{r\_obs}t_N\rightarrow_{R1}\cdots$ be a structured fair derivation for some $N\geqslant 0$. Let S be the canonical system associated to t_N, and $g(t_N)=\text{process\_graph}(S)$. Then $g(t_N)$ is an $r\tau$-normal process graph.*

Proof. The proof is by induction on the level n of the elements S_i of the partition Π on the system S.

$n=0$: Level$(S_i)=0$ implies $S_i\equiv\{X_i=\text{nil}\}$ by definition. In this case, the subterm "nil" is trivially a recursive normal form and denotes an $r\tau$-normal process graph.

$n=k+1$: Assume the result for level k (inductive hypothesis) and consider $\text{level}(S_i)=k+1$. The proof is by contradiction: let $g(t_N)$ be a graph which is not $r\tau$-normal. This means that at least one of the two conditions above is not satisfied.

$r\tau$-rigidity. If $g(t_N)$ is not $r\tau$-rigid, then there exists an $r\tau$-autobisimulation R of $g(t_N)$ such that there exist two nodes s_i, s_j, $i \neq j$ and $(s_i, s_j) \in R$. It follows that the subgraphs $(g)_{s_i}$, $(g)_{s_j}$ with root s_i, s_j respectively, are τ-bisimilar. Let X_i, X_j be the main variables of the subsystems of S associated to the subgraphs $(g)_{s_i}$, $(g)_{s_j}$ respectively.

Let us consider the following cases:

1. $\text{level}(X_i)$, $\text{level}(X_j) \leqslant k$

1.1. $(g)_{s_i}$, $(g)_{s_j}$ are distinct. Since they are $r\tau$-normal by inductive hypothesis, they must be the same graph. Therefore, X_i and X_j are two equivalent variables in S thus contradicting its canonicity.

1.2. $(g)_{s_j}$ is a subgraph of $(g)_{s_i}$. Since they are $r\tau$-normal by inductive hypothesis, it can only happen that $(g)_{s_i}$ denotes a term $\tau . P$ and $(g)_{s_j}$ denotes P for some term P, and in S there exist two equations $X'=\mu . X_i$ for some action prefix operator $\mu.$, where $\text{level}(X')=k+1$. This contradicts the hypothesis that t_N is in recursive normal form since $\rightarrow_{\text{f\_obs}}$ can be applied on the subterm $\mu . \tau . P$ denoted by X'.

2. $\text{level}(x_i)=k+1$ and $\text{level}(X_j) \leqslant k$. Since $(g)_{s_i}$ and $(g)_{s_j}$ are bisimilar and $(g)_{s_j}$ is $r\tau$-normal by inductive hypothesis, it follows that $(g)_{s_j}$ is a subgraph of $(g)_{s_i}$. Note that this situation can only happen if the two subgraphs denote infinite trees, X_i may denote a nonrecursive term but the tree corresponding to the subgraph $(g)_{s_i}$ is infinite because it contains $(g)_{s_j}$ that denotes a recursive term t_j, namely $X_j \in \mathcal{V}ar(E_j)$.

Let us consider all the possible contexts in which X_i uses X_j.

Prefix context: let $X_i=E_i$ such that $E_i \equiv \mu . X_2$, for some action operator $\mu.$, $X_2=E_2$ in S and $X_j \in \mathcal{V}ar(E_2)$.

(a) If $X_i \notin \mathcal{V}ar(E_i)$ then $\text{level}(X_2)=k$ and the term associated to E_2 is in recursive normal form. X_i and X_j are τ-bisimilar and this means that X_i denotes a subterm which can only be the unfolding of the subterm associated to X_j, thus contradicting the canonicity hypothesis of S.

(b) If $X_i \in \mathcal{V}ar(E_i)$, then $\text{level}(X_2)=k+1$ and there exists an equation $X_r=E_r$ such that $X_r \in \mathcal{V}ar(E_2)$, $\text{level}(X_r)=k+1$, and $E_r \equiv \text{op}(X_{r1}, \ldots, X_i, \ldots, X_{rq})$.

X_r and X_j are τ-bisimilar, the subterm t_i, associated to the subsystem with X_i as the main variable, denotes a recursive term and, by inductive hypothesis, X_j must denote a recursive term as well. At term level the pattern for $\rightarrow_{\text{CR}}$ could have not been applied because it was not possible to prove that either $t_i\{X_i/X_j\}$, i.e. the term t_i in which every reference to X_j has been replaced with a reference to X_i, or one of its unfoldings with respect to R1, is bisimilar to t_j, i.e. the term associated to the subsystem with X_j as the main variable. Since $(g)_{s_i}$ and $(g)_{s_j}$ are τ-bisimilar, every move from one has to be done in the other and viceversa. This means that in the subsystem associated to X_i there exists an equation $X_{ih}=E_{ih} \equiv X_{ih_1}+\cdots+X_{ih_n}$, in correspondence of which an equation $X_{jk}=E_{jk}$ exists in S_j such that either $E_{jk} \equiv X_{jk_1}+\cdots+X_{jk_m}$ with $m \leqslant n$ or $E_{jk} \equiv \mu . X_{jk_2}$, for some action operator $\mu.$, $X_{jk_2}=E_2$. Thus, in general, in E_{ih} there are some variables that denote equivalent equations, i.e. they are τ-bisimilar to the same

equation in $S_{jk}.E_{ih}$ contains at most a summand, whose corresponding variable is of level less or equal to k, and at least a summand, whose corresponding variable is of level $k+1$, which denote bisimilar distinct graphs because one of them uses X_i and the other uses X_j. This means that the term $t_i\{X_i/X_j\}$ could be reduced by $\rightarrow_{\text{f\_obs,CT}}$, as long as it becomes equal to t_j, thus allowing the application of $\rightarrow_{\mathbf{CR}}$ and contradicting the hypothesis that t_N is a recursive normal form.

Summation context: let $X_i=E_i$ in S such that $E_i \equiv X_{i1}+\cdots+X_{iq}$, $X_{i1}=E_{i1},\ldots,X_{iq}=E_{iq}$ and $X_j \in \mathscr{V}ar(E_i)$.

(a) If $X_i \notin \mathscr{V}ar(E_i)$, the terms associated to $E_{i1},\ldots,E_{iq}$ are in recursive normal form and denote $r\tau$-normal process graphs. Furthermore, X_i is τ-bisimilar to X_j, which denotes a recursive term. We can only have that X_i represents an unfolding of X_j, thus contradicting the canonicity hypothesis of S.

(b) If $X_i \in \mathscr{V}ar(E_i)$ with $E_i \equiv X_{i1}+\cdots+X_{iq}$, then there exists an equation $X_{ir}=E_{ir}$ such that $X_i \in \mathscr{V}ar(E_{ir})$, i.e. $\text{level}(X_{ir})=k+1$.

The proof carries on analogously as above in the second case of prefix context.

3. $\text{level}(X_i)=\text{level}(X_j)=k+1$. We have two cases by considering a prefix or summation context, respectively.

Prefix context: it can only happen that $(g)_{s_i}$ denotes a term $\tau.E$ and $(g)_{s_j}$ denotes E for some term E, and in S there exist two equations $X'=\mu.X_i$ for some action prefix operator $\mu.$, where $\text{level}(X')=\text{level}(X_i)=\text{level}(X_j)=k+1$. This contradicts the hypothesis that t_N is in recursive normal form since $\rightarrow_{\text{f\_obs,CT}}$ can be applied on the subterm $\mu.\tau.E$ denoted by X'.

Summation context: we proceed by analysing the two outermost τ-bisimilar nodes, if more than two τ-bisimilar nodes exist. The proof carries on analogously as in case 2 by considering that, when trying to apply $\rightarrow_{\mathbf{CR}}$, the τ-bisimilarity of the terms denoted by X_i and X_j is proved by checking that $t_j\{X_j/X_i\}$ is τ-bisimilar to $t_i\{X_i/X_j\}$.

Minimality: If $g(t_N)$ is not minimal, then it contains τ-loops, arc and/or double edges. As regards τ-loops, since any guarded term is built from its canonical system and the reductions performed on the term maintain its guardedness, τ-loops cannot occur. Furthermore, double edges are a particular instance of arcs, thus we have only to consider the occurrence of arcs in $g(t_N)$. The situation is as follows: as long as levels less or equal to k are considered, no arcs occur, but as soon as level $k+1$ is considered, (at least) an arc occurs. This means that, at term level, there exist two summands like $\tau.(\ldots\mu.(\ldots\tau.\ldots(E+\ldots)\ldots)\ldots)$ and $\mu.F$, where $E=F$ since they are bisimilar recursive normal forms. This corresponds to a pattern of the absorption lemma, which can be reduced by $\rightarrow_{\text{f\_obs,CT}}$, thus contradicting the hypothesis that t_N is a recursive normal form. □

The following proposition guarantees that the use of the canonical system does not introduce any relevant reduction with respect to the notion of normal form.

Proposition 5.5. *Given a term* $t_0 \in \mathscr{E}g$, *let* $D: t_0 \rightarrow_{\text{r\_obs}} t_1 \rightarrow_{\text{r\_obs}} \cdots \rightarrow_{\text{r\_obs}} t_N \rightarrow_{\mathbf{R1}} \cdots$ *be a structured fair derivation for some* $N \geqslant 0$. *Let* S *be the system associated to* t_N, *and* CS *be the canonical one. Then* $S \neq \text{CS}$ *if and only if* $t_N =_{\mathbf{R1}} E(\text{CS})$.

Proof. The implication $\Leftarrow$ is trivial: t_N is an unfolding of $E(\mathrm{CS})$, then S is not the canonical system. The other direction, instead, assures that the only reductions performed when constructing the canonical system for a recursive normal form are unfoldings with R1. Let us assume $S \neq \mathrm{CS}$. Since the two systems are built from the same term t_N and CS is canonical, by construction of S and CS, it can only be that S has more variables than CS, and some of them are equivalent. In terms of the graph $(g)_S$ associated to S this means that there are equivalent nodes, that is $(g)_S$ is not $r\tau$-rigid. Let s_i, s_j be nodes in $(g)_S$ such that the subgraphs $(g)_{s_i}$, $(g)_{s_j}$ with root s_i, s_j respectively, are τ-bisimilar. Let X_i, X_j be the main variables of the subsystems of S associated to the subgraphs $(g)_{s_i}$, $(g)_{s_j}$ respectively. The only relevant cases with respect to the term structure are when X_i refers X_j, directly or indirectly, i.e. $(g)_{s_j}$ is a subgraph of $(g)_{s_i}$, or they refer each other. The proof is by contradiction. If $t_N \neq_{\mathbf{R1}} E(\mathrm{CS})$, this means that some other kind of redundancy exists in t_N. Since $(g)_{s_i}$ denotes an infinite tree and the node s_i is not an unfolding node, it must be a recursive node, that is X_j is a recursive variable. Therefore it follows that $E(S_i)$ is a recursive expression which contains $E(S_i)$ with $(g)_{s_i}$ and $(g)_{s_j}$ τ-bisimilar, that is every move from one has to be done in the other and viceversa. By the same arguments of the proof above, case 2), this means that a reduction pattern for $\rightarrow_{\mathbf{CR}}$ can be found that t_N is not a recursive normal form, thus contradicting the hypothesis. □

Corollary 5.6 (Completeness of $\rightarrow_{\mathrm{r\_obs}}$ with respect to OBSREC_g). *For any $t_1, t_2 \in \mathscr{E}_g$ such that $\mathrm{OBSREC}_g \vdash t_1 = t_2$, then $t_{N1} =_{\mathrm{CT}} t_{N2}$ where t_{N1}, t_{N2} are the recursive normal forms of t_1, t_2, respectively.*

6. A rewriting strategy for $\rightarrow_{\mathrm{r\_obs}}$

We can now define a rewriting strategy in order to compute a specific structured fair derivation $t_0 \rightarrow_{\mathrm{r\_obs}} t_1 \rightarrow_{\mathrm{r\_obs}} \cdots \rightarrow_{\mathrm{r\_obs}} t_N \rightarrow_{\mathbf{R1}} \cdots$ from any term $t_0 \in \mathscr{E}_g$, such that an upper bound for the index N can be determined.

Let us first show that, given a term in normal form with respect to $\rightarrow_{\mathrm{f\_obs,CT}} \cup \rightarrow_{\mathbf{CR}} \cup \rightarrow_{\mathbf{A_p}}$, rewriting by $\rightarrow_{\mathbf{R1}}$ cannot generate any redexes for $\rightarrow_{\mathbf{CR}}$ and $\rightarrow_{\mathbf{A_p}}$. Moreover, rewriting by $\rightarrow_{\mathrm{f\_obs,CT}}$ preserves normality with respect to $\rightarrow_{\mathbf{CR}} \cup \rightarrow_{\mathbf{A_p}}$.

Lemma 6.1. *Let $t \in \mathscr{E}_g$ be a term in normal form with respect to $\rightarrow_{\mathrm{f\_obs,CT}} \cup \rightarrow_{\mathbf{CR}} \cup \rightarrow_{\mathbf{A_p}}$. If $t \rightarrow_{\mathbf{R1}} t'$, then t' is in normal form with respect to $\rightarrow_{\mathbf{CR}} \cup \rightarrow_{\mathbf{A_p}}$. Moreover, if $t' \rightarrow_{\mathrm{f\_obs,CT}} t''$, then t'' is still in normal form with respect to $\rightarrow_{\mathbf{CR}} \cup \rightarrow_{\mathbf{A_p}}$.*

Proof. If $t \rightarrow_{\mathbf{R1}} t'$, then t contains a recursive subterm which is rewritten by R1:

$$t[\mathrm{rec}\, X \,.\, E] \rightarrow_{\mathbf{R1}} t[E\{\mathrm{rec}\, X \,.\, E/X\}] \equiv t'.$$

A redex for $\rightarrow_{A_p}$ is an instance of $\mathrm{rec}\,X.\tau.E$. This means that t' contains redexes for $\rightarrow_{A_p}$ if and only if they already occurred in t, but t is in normal form with respect to $\rightarrow_{A_p}$ by hypothesis.

A redex for $\rightarrow_{CR}$ is a term $\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F]$ which $\rightarrow_{R1}$ can rewrite on both recursive subterms.

(i) Unfolding on the outermost recursive subterm: $t[\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F]] \rightarrow_{R1} t[E[\mathrm{rec}\,Y.F]\{\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F]/X\}] \equiv t'$. It t' contains a redex for $\rightarrow_{CR}$ given by the subterm $\mathrm{rec}\,Y.F\{\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F]/X\}$, then the subterm $\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F]$ of t is necessarily a redex for $\rightarrow_{CR}$, but t is in normal form with respect to $\rightarrow_{CR}$.

(ii) Unfolding on the innermost recursive subterm: $t[\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F]] \rightarrow_{R1} t[\mathrm{rec}\,X.E[F\{\mathrm{rec}\,Y.F/Y\}]] \equiv t'$.

If t' contains a redex for $\rightarrow_{CR}$ given by the subterm $\mathrm{rec}\,X.E[F\{\mathrm{rec}\,Y.F/Y\}]$, then the subterm $\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F]$ of t is necessarily a redex for $\rightarrow_{CR}$, but t is in normal form with respect to $\rightarrow_{CR}$. Hence, t' is in normal form with respect to $\rightarrow_{CR} \cup \rightarrow_{A_p}$.

Let us now consider $t' \rightarrow_{f\_obs,CT} t''$. Since $\rightarrow_{CR}$ is applied modulo $\rightarrow_{f\_obs,CT}$, t'' cannot contain any redexes for $\rightarrow_{CR}$. A redex $\mathrm{rec}\,X.\tau.E$ for $\rightarrow_{A_p}$ can be generated in t'' only if the redex $\tau.E+E$ for $\rightarrow_{f\_obs,CT}$ is the body of "rec X." in t'. This cannot happen since t' is obtained from t by applying $\rightarrow_{R1}$ and t is in normal form with respect to $\rightarrow_{f\_obs,CT}$. □

Example 6.2. Rewriting by $\rightarrow_{R1}$ can generate redexes for $\rightarrow_{f\_obs,CT}$. For example, the term $(\mathrm{rec}\,X.(a.X+b.\mathrm{nil}))+\tau.b.\mathrm{nil}$ is in normal form with respect to $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$, but rewriting by $\rightarrow_{R1}$ generates a redex for $\rightarrow_{f\_obs,CT}$:

$$(\mathrm{rec}\,X.(a.X+b.\mathrm{nil}))+\tau.b.\mathrm{nil} \rightarrow_{R1} (a.(\mathrm{rec}\,X.(a.X+b.\mathrm{nil}))+b.\mathrm{nil})+\tau.b.\mathrm{nil}$$

$$\rightarrow_{f\_obs,CT} a.(\mathrm{rec}\,X.(a.X+b.\mathrm{nil}))+\tau.b.\mathrm{nil}.$$

Definition 6.3 (effective derivation). Given a term $t_0 \in \mathscr{E}\mathscr{g}$, let D be a derivation from t_0 obtained as follows:

1. all possible reductions by $(\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p})$, let us say $k \geqslant 0$, are first applied: $t_0(\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p})^k t_k$;
2. one unfolding step by $\rightarrow_{R1}$ on each recursive subexpression, let us say $r \geqslant 0$, is then applied starting from the deepest ones: $t_k(\rightarrow_{R1})^r t_{k+r}$;
3. all possible new reductions by $\rightarrow_{f\_obs,CT}$, let us say $v \geqslant 0$, are then performed: $t_{k+r}(\rightarrow_{f\_obs,CT})^v t_{k+r+v}$;
4. only rewritings by $\rightarrow_{R1}$ are finally applied: $t_{k+r+v} \rightarrow_{R1}^{\omega} t_\infty$.

The derivation D is called *effective* and denoted as $D(k,r,v)$.

Proposition 6.4. *Given a term $t_0 \in \mathscr{E}\mathscr{g}$, let $D(k,r,v)$ be an effective derivation from t_0. Then $D(k,r,v)$ is a structured fair derivation $t_0 \rightarrow_{r\_obs} t_1 \rightarrow_{r\_obs} \cdots \rightarrow_{r\_obs} t_N \rightarrow_{R1} \cdots$, where the index N has an upper bound $U(N)=k+r+v$.*

Proof. It is sufficient to prove that t_{k+r+v} is a recursive normal form. We have only to show that rewriting t_{k+r+v} by $\rightarrow_{R1}$ cannot generate redexes for $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$, i.e. t_{k+r+v} is in normal form with respect to $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$. By the guardedness hypothesis, rewriting t_{k+r+v} by $\rightarrow_{R1}$ cannot generate any redexes for $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$, unless they already occurred in t_{k+r+v}, but this is not possible because $D(k,r,v)$ is an effective derivation. By Lemma 6.1 only $\rightarrow_{R1}$ can be further applied and this means that t_{k+r+v} is in normal form with respect to $\rightarrow_{f\_obs,CT} \cup \rightarrow_{CR} \cup \rightarrow_{A_p}$. □

7. Conclusions and related works

We have presented a rewriting relation for observational congruence over guarded recursive (finite-state) CCS expressions. On the basis of this relation, a decision procedure has been defined that shows the use of the axiomatization for a behavioural equivalence as a proof procedure and not only as a semantic device.

In this way we extend the approach based on term rewriting presented in [4] to finite-state CCS expressions. In this respect, the fact that the rewriting relation $\rightarrow_{r\_obs}$ is defined over guarded CCS expressions does not represent a limitation. It is, in fact, possible to extend its application to unguarded recursive CCS expressions, for which Milner has given a set of correct and complete axioms to transform an unguarded expression into an equivalent guarded one [17]. Furthermore, it is possible to extend the process algebra through the introduction of parallel, restriction and relabelling operators, for which correct sets of axioms have been defined. If the recursion and parallel operators interact in such a way that expressions are still finite-state, our rewriting strategy remains complete.

As far as related approaches to the verification of observational congruence are concerned, our attempt has to be considered complementary and not opposed to the other ones based on the finite-state representation of a term. Most of the verification systems for process algebras are based on the finite-state automata representation of a process term [12, 13]. This means that in order to perform verification, they first transform the term into an equivalent finite-state automata. This has the advantage of making the application of efficient graph partition algorithms for checking behavioural equivalences possible. On the other hand, these systems exhibit a few limitations mostly due to the fact that they do not provide sufficient control over the verification process. In fact, they suffer from the state explosion problem and can only perform fully automatic proofs. Thus, there is no way either to incrementally control (and prune) the state growth or to accomodate the verification of infinite state processes or to produce a good diagnostics to help identify errors.

All these motivations have recently led to the definition of tools based on equational reasoning [19]. In these systems the idea is to rely on the syntactic representation of processes and to use the various equivalence laws to carry on formal proofs. In general, these systems are less efficient, but they offer a way to cope with the finite-state limitation and allow both interactive and automatic techniques to be

defined. Therefore, their application range is wider and they provide users with a more flexible interface to carry on their specific verification proofs.

The rewriting strategy presented in this paper is a step ahead in the direction of more powerful systems based on equational reasoning. It makes reasoning on recursive terms explicit and allows for a better comprehension of the infinite nature of a term. In particular, our treatment of the axiom U2 by means of the axiom CE, and therefore of the rule CR, allows for a syntactic, even if rather complex, treatment of the "infinite" redundancy of a term. It is worth noting that, in order to turn the axiomatization into rewriting rules, it has been necessary to simplify some of the axioms, namely U2 loses much of its elegance in favour of a more practical and specific collapsing rule. Furthermore, we have introduced a new axiom, the action prefix equivalence, that can be derived from the given axiomatization. The inclusion of this axiom is only motivated in light of the kind of rewritings we want to deal with (structured derivations) and not by semantic considerations. On the contrary, there has been no problem in dealing with the unfolding axiom U1.

The approach we have followed to define the rewriting strategy integrates a number of results and techniques which have been developed in different fields of computer science. In our opinion, the use of the framework of infinite rewritings and ω-normal forms is particularly interesting since, from the rewriting technique point of view, our experience represents both a concrete application example and an extension towards the treatment of infinite derivations for non-left linear term rewriting systems [11].

Acknowledgements

We would like to thank the anonymous referees and Giovanni Mandorino for their very helpful comments on previous versions of this paper.

Appendix A

As mentioned in Section 2.2, the starting point of $\rightarrow_{\text{f\_obs}}$ is the following TRS R_{OBS}:

R_{OBS} r1. $E+\text{nil}\rightarrow E$

r2. $E+E\rightarrow E$

r3. $\mu.\tau.E\rightarrow\mu.E$

r4. $\tau.E+E\rightarrow\tau.E$

r5. $\mu.(E+\tau.F)+\mu.F\rightarrow\mu.(E+\tau.F)$

and the strategy $\rightarrow_{\text{f\_obs}}$ can be seen as composed of two phases. The first phase normalizes the input term with respect to R_{OBS}. The second phase works on the resulting term by looking for summands to be deleted according to the absorption

lemma. It rewrites the term by using T2 and T3 as expansion rules (*expansion* process) and, as soon as possible, it deletes the redundant summands by means of R_{OBS} (*reduction* process). These reductions are performed by using a specific redex selection criterion that prevents those reductions which are exactly opposite to the previous expansions by T2 and T3. The expansion and reduction steps are applied as long as there exist summands to be deleted. Finally, to obtain the OBS-normal form, the current term is rewritten by applying the reductions opposite to the previous expansions (*contraction* process) by using a redex selection criterion that selects the smallest redexes with respect to the chosen term ordering $\succ$. In the following $\lambda.$ ranges over the set of action prefix operators $\mathscr{P}_1 - \{\tau.\}$.

Definition A.1. A term t is *expandible* if it is an AC instance of the right-hand side of T2 or T3. An *expansion step* consists of the application of T2 or T3 as expansion rules in the following way.

$$\tau.x \to_{\text{T2}} \underline{\tau.x} + x \quad \lambda.(x+\tau.y_0) \to_{\text{T3}} \underline{\lambda.(x+\tau.y_0)} + \lambda.y_0 + \lambda.y_1 + \cdots + \lambda.y_r$$

if $\tau.y_j$ is a top level summand of $x, j=1,\ldots,r$ (underlined denotes an already expanded summand).

In the above definition we assume that an expansion step by T3 is performed by applying T3 for all the possible AC instances of its right-hand side: for example, the term $\lambda.(\tau.x+\tau.y+z)$ is expanded into the term $\lambda.(\tau.x+\tau.y+z)+\lambda.x+\lambda.y$ (T3 is applied twice with the two different redexes). In this way, the expansion of all the τ-prefixed subterms is guaranteed: it is necessary because, during the expansion process, we do not know which subterm, if any, will act as μ-derivative. Moreover, in the presence of terms which may be expanded by using both T2 and T3, like $\tau.(x+\tau.y)$, T2 is always applied, since for $\mu. = \tau.$, T3 can be derived in OBS by using T2. Thus, we have replaced the action prefix operator $\mu.$ with $\lambda. \neq \tau.$ in T3.

The rewriting relation $\to_{\text{f\_obs}}$ is supposed to work under the following *flattening hypothesis*: given the input sumform $E = \sum_{1 \leqslant i \leqslant n} \mu_i.E_i$ $(n>1)$, each summand $\mu_i.E_i$ is already in OBS-normal form, while for $n=1$ E_1 is in OBS-normal form. Let us now state the following facts:

F1. Each summand $\mu.F$ to be deleted in order to derive the OBS-normal form of E, is a top level summand $\mu_j.E_j$ for some j.

F2. For each summand $\mu_j.E_j$ to be deleted, there exists a summand $\mu_k.E_k$, $k \neq j$, such that $\mu_k.E_k \succcurlyeq \mu_j.E_j$.

F3. Let $t_i \to_{\text{T2-T3}} t_i + t_{i+1}$ be an expansion step, then $t_i \succ t_{i+1}$. In fact, if $t_i \to_{\text{T2}} t_i + t_{i+1}$ then $t_i \succ t_{i+1}$ follows from the subterm property of the apo. If $t_i \to_{\text{T3}} t_i + t_{i+1}$ where t_{i+1} is the summation of all summands $\lambda.t_{i_j}$ such that t_{i_j} is τ-prefixed in t_i, the fact follows from the definition of multiset ordering and the subterm property.

F4. If E' and F are observationally congruent subterms of E, then E' and F are equivalent modulo the AC axioms since they are subterms in OBS-normal form by the flattening hypothesis. Therefore, $E' =_{OBS} F$ reduces to verify $E' =_{AC} F$.

Let us now describe in details how the OBS-normal form of a sumform E can be derived. The recursive procedure *normal_form* is defined by cases on the sumform structure as follows, thus assuring the application of $\rightarrow_{f\_obs}$ under the flattening hypothesis:

$$\begin{aligned}\text{normal\_form}(E) =_{def}\ & \underline{\text{if}}\ E = \text{nil}\ \underline{\text{then}}\ \text{nil};\\ & \underline{\text{if}}\ E = \mu.E'\ \underline{\text{then}} \rightarrow_{f\_obs}(\mu.\text{normal\_form}(E'));\\ & \underline{\text{if}}\ E = \sum_{1\leqslant i\leqslant n} E_i\ (n>1)\ \underline{\text{then}}\\ & \rightarrow_{f\_obs}\left(\sum_{1\leqslant i\leqslant n} \text{normal\_form}\ (E_i)\right)\end{aligned}$$

The notation $\rightarrow_{f\_obs}(E)$ denotes the normal form of E with respect to $\rightarrow_{f\_obs}$.

The basic steps of $\rightarrow_{f\_obs}$ can be defined by means of inference rules and then $\rightarrow_{f\_obs}$ may be defined as a regular expression built from such inference rules. Normalization with respect to R_{OBS}, expansion, reduction and contraction are the basic steps. Expansion, reduction and contraction are performed according to specific redex selection criteria, which represent the applicability conditions of the corresponding inference rules.

Let t be a sumform $\sum_{1\leqslant i\leqslant n} t_i$. We define the following sets:

$$\text{Summands}(t) = \{t_i \mid i = 1, \ldots, n\}$$

$$\text{Mark}(t) = \{(t_i, m_i) \mid t_i \in \text{Summands}(t),\ m_i \text{ is } t_i\text{'s mark},\ i = 1, \ldots, n\}$$

$$\text{Label}(t) = \{(t_i, l_i) \mid t_i \in \text{Summands}(t),\ l_i \text{ is } t_i\text{'s label},\ i = 1, \ldots, n\}$$

where

$$m_i = \begin{cases} 0 & \text{if } t_i \text{ is a non-expanded summand,}\\ 1 & \text{otherwise,}\end{cases}$$

$$l_i = \begin{cases} \text{Default} & \text{if } t_i \text{ is a summand of the input term } t,\\ l_k \cdot t_k & \text{if the summand } t_i \text{ is generated by an expansion}\\ & \text{step from } t_k \text{ for some } k.\end{cases}$$

A mark m_i and a label l_i are associated to every summand t_i, providing information about its expanded/contracted status and about that summand whose expansion process has generated it. The mark is used to prevent those reductions which are exactly opposite to the previous expansion steps. The label is used both to prevent reductions involving summands derived from different expansion processes and to properly drive the contraction processes. Thus, the structure manipulated by the inference rules is a triple (t, M, L), where t is the current term, M is $\text{Mark}(t)$ and L is $\text{Label}(t)$. The mark and the label of a term are supposed to be inherited by its

summands, i.e. if t'_j is a summation, the notations $(t'_j, m) \in M$ and $(t'_j, l) \in L$ denote $(t'_{j_k}, m) \in M$ and $(t'_{j_k}, l) \in L$ for each $t'_{j_k} \in \text{Summands}(t'_j)$, respectively.

Let us now give the inference rules.

R_{OBS}-Normalization: $(t, \emptyset, \emptyset) \vdash (t', M, L)$ if $t \to^{!}_{R_{\text{OBS}}/\text{AC}} t'$ where $M = \{(t'_i, 0) \mid t'_i \in \text{Summands}(t')\}$ and $\{L = (t'_i, \text{Default}) \mid t'_i \in \text{Summands}(t')\}$.

Expansion: $(t, M, L) \vdash (t', M', L')$ if $\exists t_j \in \text{Summands}(t)$ such that $(t_j, 0) \in M$ and $t_j \to_{\text{T2-T3/AC}} t_j + t'_j$ and $\exists t_k \in \text{Summands}(t)$ such that $(t_k, 0) \in M$, $(t_k, \text{Default}) \in L$ and $t_j \succ t_k$ and $\nexists t_p \in \text{Summands}(t)$ such that t_p is expandible, $(t_p, 0) \in M$ and $t_p \succ t_j$ where $t' = t[t_j \leftarrow t_j + t'_j]$, $M' = (M - \{(t_j, 0)\}) \cup \{(t_j, 1), (t'_j, 0)\}$, $L' = L \cup \{(t'_j, l_j \cdot t_j)\}$.

Reduction: $(t, M, L) \vdash (t', M', L')$ if $\exists t|_u = _{\text{AC}} t_j + t'_j \to_{R_{\text{OBS}}/\text{AC}} t_j, (t|_u, 0) \in M$ and $(t'_j, \text{Default}) \in L$ where $t' = t[t_j]_u$, $M' = M - \{(t'_j, 0)\}$, $L' = L - \{(t'_j, \text{Default})\}$.

Contraction: $(t, M, L) \vdash (t', M', L')$ if $\exists t|_u = _{\text{AC}} t_j + t'_j \to_{R_{\text{OBS}}/\text{AC}} t_j$ and $(t'_j, l_j \cdot t_j) \in L$ and $\nexists t_k \in \text{Summands}(t)$ such that $(t_k, l'_j \cdot t'_j) \in L$ where $t' = t[t_j]_u$, $M' = M - \{(t'_j, m'_j)\}$, $L' = L - \{(t'_j, l_j \cdot t_j)\}$.

The rule Expansion deals with only the summands t_j of t. Deeper expandible subterms are considered in next (iterative) applications of the rule Expansion. This rule selects one of the "greatest and incomparable" summands t_j, whose expansion might allow a reduction, i.e. there must exist a summand t_k smaller than t_j (see fact F2). Thus, if there are two expandible summands t_1, t_2 such that $t_1 \succ t_2$, the rule Expansion first expands t_1 and considers t_2 only at a next expansion step (if t_2 has not been deleted by the rule Reduction).

As far as the contraction process is concerned, note that an expansion step of a term t by T3 may generate a term $t' = \lambda.(x + \tau.y_0) + \lambda.y_0 + \lambda.y_1 + \cdots + \lambda.y_r$. In that case, each $\lambda.y_j, j = 0, \ldots, r$, is labelled $l_t \cdot t$ and the rule contraction has to perform $r+1$ steps in order to rebuild t from t'. At any single step the contraction process reduces $\lambda.(x + \tau.y_0) + \lambda.y_j$, for some j, to $\lambda.(x + \tau.y_0)$ and still allows all the remaining contraction steps to be performed. Thus, the order in which these steps are applied is not significant, independently of the ordering relations among the $\lambda.y_j(s)$.

Finally, $\to_{\text{f\_obs}}$ is defined as the following regular expression:

$$\to_{\text{f\_obs}} =_{\text{def}} R_{\text{OBS}}\text{-normalization; absorption}$$

where absorption $=_{\text{def}}$ (expansion; reduction)*; contraction*.

Due to the flattening hypothesis, absorption cannot produce redexes for R_{OBS}-normalization.

Appendix B

Proposition 4.11. The rewriting relation $\to_{\text{r\_obs}}$ is locally confluent over $\mathscr{E}\mathscr{g}$.

Proof. Let $t \in \mathscr{E}\mathscr{g}$ be reducible by means of the rules in $\to_{\text{r\_obs}}$ on non-independent redexes. By case analysis we consider all the possible ways a term can be rewritten by two rules.

1. The term t can be rewritten by $\rightarrow_{\text{f\_obs,CT}}$. If $t'_{\text{f\_obs}} \leftarrow t \rightarrow_{\text{f\_obs}} t''$ local confluence follows from the canonicity of $\rightarrow_{\text{f\_obs}}$ as proved in [10]. If $t'_{\text{f\_obs,CT}} \leftarrow t \rightarrow_{\text{f\_obs,CT}} t''$ such that one or both rewritings are not possible via $\rightarrow_{\text{f\_obs}}$ only, there exists a term s such that $s =_{\text{CT}} t$, $s'_{\text{f\_obs}} \leftarrow s \rightarrow_{\text{f\_obs}} s''$ for some s', s'', and $s' =_{\text{CT}} t'$, $t'' =_{\text{CT}} s''$. The confluece of s', s'' to a common term follows from the canonicity of $\rightarrow_{\text{f\_obs}}$.

2. The term t can be rewritten by $\rightarrow_{\text{R1}}$ and any other rule in $\rightarrow_{\text{f\_obs}}$. Local confluence follows from Propositions 3.4 and 4.10.

3. The term t can be rewritten by $\rightarrow_{\text{f\_obs,CT}}$ and by $\rightarrow_{\text{CR}}$.

3.1. The redex for $\rightarrow_{\text{CR}}$ is a subterm of the redex for $\rightarrow_{\text{f\_obs,CT}}$. We distinguish two situations.

3.1.1. $\rightarrow_{\text{f\_obs,CT}}$ deletes a summand different from "nil". The case in which the deleted summand is "nil" is trivial.

Given $t \equiv E_1 + \cdots + E_n$, suppose without loss of generality that a top level summand E_i, which is also reducible by $\rightarrow_{\text{CR}}$, is deleted by $\rightarrow_{\text{f\_obs,CT}}$. By definition of $\rightarrow_{\text{f\_obs,CT}}$ there exists a summand E_k in t which contains some derivative E', which is equivalent to E_i. Since $\rightarrow_{\text{f\_obs,CT}}$ does not make use of $\rightarrow_{\text{CR}}$, E' can also be rewritten by $\rightarrow_{\text{CR}}$ obtaining a summand E'_k. On the other side, once E_i has been rewritten by $\rightarrow_{\text{CR}}$ obtaining E'_i, it is sufficient to apply $\rightarrow_{\text{CR}}$ on E' and then $\rightarrow_{\text{f\_obs,CT}}$ to delete E'_i.

$$
\begin{array}{ccc}
t \equiv E_1 + \cdots + E_i + \cdots + E_k[E'] + \cdots + E_n & & \\
\downarrow_{\text{f\_obs,CT}} & & \downarrow_{\text{CR}} \text{ on } E_i \\
E_1 + \cdots + E_{i-1} + E_{i+1} + \cdots + E_k[E'] + \cdots + E_n & & E_1 + \cdots + E'_i + \cdots + E_k[E'] + \cdots + E_n \\
\downarrow_{\text{CR}} \text{ on } E' & & \downarrow_{\text{CR}} \text{ on } E' \\
E_1 + \cdots + E_{i-1} + E_{i+1} + \cdots + E'_k + \cdots + E_n & \leftarrow_{\text{f\_obs,CT}} & E_1 + \cdots + E'_i + \cdots + E'_k + \cdots + E_n \\
 & & \text{on } E'_i \text{ and } E'_k
\end{array}
$$

Note that the two rewritings by $\rightarrow_{\text{CR}}$ on the right-hand side, on E_i and E' respectively, can be applied in any order.

Moreover, if the redex for $\rightarrow_{\text{CR}}$ is contained in a summand different from those involved in the reduction (E_i, E_k), this can be seen as a situation of independent redexes. We have a similar situation when the redex for $\rightarrow_{\text{CR}}$ occurs in the context of E' in E_k and is not involved in the reduction (E_i, E_k).

3.1.2. $\rightarrow_{\text{f\_obs,CT}}$ deletes an internal action.

This situation can be simply depicted by the following diagram:

$$
\begin{array}{ccc}
t[\mu.\tau.E] & & E \text{ contains a redex for } \rightarrow_{\text{CR}} \\
\downarrow_{\text{f\_obs,CT}} & \downarrow_{\text{CR}} & \\
t[\mu.E] & t[\mu.\tau.E'] & \\
\downarrow_{\text{CR}} & \downarrow_{\text{f\_obs,CT}} & \\
t[\mu.E'] & &
\end{array}
$$

3.2. The redex for $\rightarrow_{\text{f\_obs,CT}}$ is a subterm of the redex for $\rightarrow_{\text{CR}}$.
This situation is independent of the kind of reduction performed by $\rightarrow_{\text{f\_obs,CT}}$. The rule $\rightarrow_{\text{CR}}$ is applied on a recursive subterm $\text{rec}\,X.E[F[\text{rec}\,Y.F']]$. The redex for $\rightarrow_{\text{f\_obs,CT}}$ can occur in the external context E of $\text{rec}\,Y.F'$ or in F'. If it occurs in E, $\rightarrow_{\text{CR}}$ recognizes this reduction but does not apply it (see remarks in Section 3). The term $t[\text{rec}\,X.E\{X/F[\text{rec}\,Y.F']\}]$ resulting from the application of $\rightarrow_{\text{CR}}$ can still be reduced by $\rightarrow_{\text{f\_obs,CT}}$, thus obtaining $t[\text{rec}\,X.E''\{X/F[\text{rec}\,Y.F']\}]$. On the other side, $\rightarrow_{\text{f\_obs,CT}}$ reduces E to E''. The resulting term is still reducible by $\rightarrow_{\text{CR}}$, thus closing the diagram.

$$\begin{array}{ccc} & t[\text{rec}\,X.E[F[\text{rec}\,Y.F']]] & \\ & \downarrow_{\text{f\_obs,CT}} \quad \downarrow_{\text{CR}} & \\ t[\text{rec}\,X.E''[\text{rec}\,Y.F]] & & t[\text{rec}\,X.E'\{X/F[\text{rec}\,Y.F']\}] \\ \downarrow_{\text{CR}} & & \downarrow_{\text{f\_obs,CT}} \\ & t[\text{rec}\,X.E''\{X/F[\text{rec}\,Y.F']\}] & \end{array}$$

If the redex for $\rightarrow_{\text{f\_obs,CT}}$ occurs in F', the same arguments apply except that the last reduction by $\rightarrow_{\text{f\_obs,CT}}$ on $t[\text{rec}\,X.E'\{X/F[\text{rec}\,Y.F']\}]$ is not necessary.

$$\begin{array}{ccc} & t[\text{rec}\,X.E[F[\text{rec}\,Y.F']]] & \\ & \downarrow_{\text{f\_obs,CT}} \quad \downarrow_{\text{CR}} & \\ t[\text{rec}\,X.E[F''[\text{rec}\,Y.F'']]] & & t[\text{rec}\,X.E'[X/F[\text{rec}\,Y.F']]] \\ \downarrow_{\text{CR}} & & \\ t[\text{rec}\,X.E'\{X/F''[\text{rec}\,Y.F'']\}] & & \end{array}$$

and

$$t[\text{rec}\,X.E'\{X/F''[\text{rec}\,Y.F'']\}] = t[\text{rec}\,X.E'\{X/F[\text{rec}\,Y.F']\}].$$

4. The term t can be rewritten by $\rightarrow_{\text{f\_obs,CT}}$ and by $\rightarrow_{\text{A}_\text{p}}$.

4.1. The redex for $\rightarrow_{\text{A}_\text{p}}$ is a subterm of the redex for $\rightarrow_{\text{f\_obs,CT}}$. The same arguments as in case *3.1* apply, with $\rightarrow_{\text{CR}}$ replaced by $\rightarrow_{\text{A}_\text{p}}$.

4.2. The redex for $\rightarrow_{\text{f\_obs,CT}}$ is a subterm of the redex for $\rightarrow_{\text{A}_\text{p}}$. The following diagram illustrates this situation, for the cases in which $E \neq \tau.E'$:

$$\begin{array}{ccc} t[\text{rec}\,X.\tau.E] & & E \text{ contains a redex for } \rightarrow_{\text{f\_obs,CT}} \\ \downarrow_{\text{f\_obs,CT}} \quad \downarrow_{\text{A}_\text{p}} & & \\ t[\text{rec}\,X.\tau.E'] \quad t[\tau.\text{rec}\,X.E\{\tau.X/X\}] & & \\ \downarrow_{\text{A}_\text{p}} \quad \downarrow_{\text{f\_obs,CT}} & & E\{\tau.X/X\} \text{ is still reducible by } \rightarrow_{\text{f\_obs,CT}} \\ t[\tau.\text{rec}\,X.E'\{\tau.X/X\}] & & \end{array}$$

In the case in which $E \equiv \tau.E'$, the redex for $\rightarrow_{\text{f\_obs,CT}}$ is $\tau.\tau.E'$, then the right-hand part of the above diagram is modified as follows: $t[\text{rec}\,X.\tau.E] \rightarrow_{\text{A}_\text{p}} t[\tau.\text{rec}\,X.E\{\tau.X/X\}] \rightarrow_{\text{A}_\text{p}} t[\tau.\tau.\text{rec}\,X.E'\{\tau.\tau.X/X\}] \rightarrow^*_{\text{f\_obs,CT}} t[\tau.\text{rec}\,X.E'\{\tau.X/X\}]$.

5. The term t can be rewritten by $\rightarrow_{A_p}$ and $\rightarrow_{CR}$.

5.1. The redex for $\rightarrow_{CR}$ is a proper subterm of the redex for $\rightarrow_{A_p}$. This situation can be simply depicted by the following diagram:

$$
\begin{array}{ccc}
 & t[\mathrm{rec}\,Z.\tau.(E''[\mathrm{rec}\,X.E'[F[\mathrm{rec}\,Y.F']]])] & \\
 & \downarrow_{A_p} \qquad\qquad \downarrow_{CR} & \\
t[\tau.\mathrm{rec}\,Z.E'[\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F]]\{\tau.Z/Z\}] & & t[\mathrm{rec}\,Z.\tau.(E''[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F']\}])] \\
 & \downarrow_{CR} \qquad\qquad \downarrow_{A_p} & \\
 & t[\tau.\mathrm{rec}\,Z.E'[\mathrm{rec}\,X.E\{X/\mathrm{rec}\,Y.F\}]\{\tau.Z/Z\}] &
\end{array}
$$

5.2. The redex for $\rightarrow_{A_p}$ is a proper subterm of the redex for $\rightarrow_{CR}$.

Let $\mathrm{rec}\,X.E'[F[\mathrm{rec}\,Y.F']]$ be a redex for $\rightarrow_{CR}$ in t, such that the external context of $\mathrm{rec}\,Y.F'$ in E and/or F' contain a redex for $\rightarrow_{A_p}$. If $\rightarrow_{A_p}$ is first applied on any redex in the external context of $\mathrm{rec}\,Y.F'$ or in F', the resulting term is still reducible by $\rightarrow_{CR}$. On the other side, once $\rightarrow_{CR}$ has been applied, the resulting term can still be reducible by $\rightarrow_{A_p}$ if a redex for $\rightarrow_{A_p}$ is in the external context of $\mathrm{rec}\,Y.F'$, while it is not reducible if the redex for $\rightarrow_{A_p}$ is only in F'.

Let $\rightarrow_{A_p}$ be applicable on the external context of $\mathrm{rec}\,Y.F'$ in E'; we then have the following diagram:

$$
\begin{array}{ccc}
 & t[\mathrm{rec}\,X.E'[F[\mathrm{rec}\,Y.F']]] & \\
 & \downarrow_{A_p} \qquad\qquad \downarrow_{CR} & \\
t[\mathrm{rec}\,X.E'F[[\mathrm{rec}\,Y.F']]] & & t[\mathrm{rec}\,X.E'[F\{X/\mathrm{rec}\,Y.F']\}] \\
 & \downarrow_{CR} \qquad\qquad \downarrow_{A_p} & \\
 & t[\mathrm{rec}\,X.E''\{X/F[\mathrm{rec}\,Y.F']\}] &
\end{array}
$$

Let $\rightarrow_{A_p}$ be applicable on F'; we then have the following diagram:

$$
\begin{array}{ccc}
 & t[\mathrm{rec}\,X.E'[F[\mathrm{rec}\,Y.F']]] & \\
 & \downarrow_{A_p} \qquad\qquad \downarrow_{CR} & \\
t[\mathrm{rec}\,X.E'[F[\mathrm{rec}\,Y.F'']]] & & t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F']\}] \\
 & \downarrow_{CR} & \\
 & t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F'']\}] &
\end{array}
$$

and $t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F'']\}] = t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F']\}]$ trivially holds.

5.3. The same subterm is a redex for both $\rightarrow_{A_p}$ and $\rightarrow_{CR}$.

Let $\rightarrow_{A_p}$ be applicable on $\mathrm{rec}\,X.E$, i.e. $E \equiv \tau.E'$ for some E'. This implies that $\mathrm{rec}\,Y.F$ has to be a redex for $\rightarrow_{A_p}$, i.e. $F' \equiv \tau.F''$ for some F'', otherwise $\rightarrow_{CR}$ would not be applicable on $\mathrm{rec}\,X.E[F[\mathrm{rec}\,Y.F']]$. The diagram is the following:

$$t[\mathrm{rec}\,X.\tau.(E''[F[\mathrm{rec}\,Y.\tau.F'']])]$$

$$\downarrow_{A_p} \qquad\qquad \downarrow_{CR}$$

$$t[\tau.(\mathrm{rec}\,X.E''[F[\mathrm{rec}\,Y.\tau.F'']]\{\tau.X/X\})] \qquad t[\mathrm{rec}\,X.\tau.(E'\{X/F[\mathrm{rec}\,Y.\tau.F'']\})]$$

(now $\rightarrow_{CR}$ is not applicable) $\downarrow_{A_p}$ $\qquad\qquad \downarrow_{A_p}$

$$t[\tau.(\mathrm{rec}\,X.E''[F[\tau.(\mathrm{rec}\,Y.F''\{\tau.Y/Y\})]]\{\tau.X/X\})] \qquad t[\tau.(\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.\tau.F'']\}\{\tau.X/X\})]$$

(now $\rightarrow_{CR}$ is again applicable) $\downarrow_{CR}$

$$t[\tau.(\mathrm{rec}\,X.E'\{X/F[\tau.(\mathrm{rec}\,Y.F''\{\tau.Y/Y\})]\}]\{\tau.X/X\})]$$

where $t[\tau.(\mathrm{rec}\,X.E'\{X/F[\tau.(\mathrm{rec}\,Y.F''\{\tau.Y/Y\})]\}]\{\tau.X/X\})]$ and $t[\tau.(\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.\tau.F'']\}\{\tau.X/X\})]$ are equal. Note that the two rewritings by $\rightarrow_{A_p}$ on the left can be applied in any order.

6. The term t can be rewritten by $\rightarrow_{CR}$.

This is the case when a redex for $\rightarrow_{CR}$ is contained in another redex for $\rightarrow_{CR}$, as in $t[\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F'[\mathrm{rec}\,Z.G']]]$ such that both $\mathrm{rec}\,X.E[\mathrm{rec}\,Y.F'[\mathrm{rec}\,Z.G']]$ (both X w.r.t. Y and X w.r.t. Z) and $\mathrm{rec}\,Y.F'[\mathrm{rec}\,Z.G']$ are redexes for $\rightarrow_{CR}$. Applying $\rightarrow_{CR}$ on the outermost redex for X w.r.t. Y results in replacing X for $\mathrm{rec}\,Y.F'[\mathrm{rec}\,Z.G']$, thus losing the previously possible reduction on it. On the other side, applying $\rightarrow_{CR}$ on the innermost redex (Y w.r.t. Z) results in a term which is still reducible by $\rightarrow_{CR}$:

$$t[\mathrm{rec}\,X.E[F[\mathrm{rec}\,Y.F'[G[\mathrm{rec}\,Z.G']]]]]$$

(on the outermost redex) $\downarrow_{CR}$ $\qquad\qquad \downarrow_{CR}$ (on the innermost redex)

$$t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F'[\mathrm{rec}\,Z.G']]\}] \qquad t[\mathrm{rec}\,X.E'[F[\mathrm{rec}\,Y.F'\{Y/G[\mathrm{rec}\,Z.G']\}]]]$$

$$\downarrow_{CR}$$

$$t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F'\{Y/G[\mathrm{rec}\,Z.G']\}]\}]$$

and

$$t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F'[\mathrm{rec}\,Z.G']\}]=t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F'\{Y/G[\mathrm{rec}\,Z.G']\}]\}].$$

Applying $\rightarrow_{CR}$ on the outermost redex for X w.r.t. Z results in replacing X for $G[\mathrm{rec}\,Z.G']$, obtaining a term which is still reducible by $\rightarrow_{CR}$ (X w.r.t. Y). On the other side, applying $\rightarrow_{CR}$ on the innermost redex (Y w.r.t. Z) results in a term, which is still reducible by $\rightarrow_{CR}$ (X w.r.t. Y):

$$t[\mathrm{rec}\,X.E[F[\mathrm{rec}\,Y.F'[G[\mathrm{rec}\,Z.G']]]]]$$

$$\downarrow_{CR} \qquad\qquad \downarrow_{CR}$$

$$t[\mathrm{rec}\,X.E'[F[\mathrm{rec}\,Y.F'\{X/G[\mathrm{rec}\,Z.G']\}]]] \qquad t[\mathrm{rec}\,X.E'[F[\mathrm{rec}\,Y.F'\{Y/G[\mathrm{rec}\,Z.G']\}]]]$$

$$\downarrow_{CR} \qquad\qquad \downarrow_{CR}$$

$$t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F'\{X/G[\mathrm{rec}\,Z.G']\}]\}]=t[\mathrm{rec}\,X.E'\{X/F[\mathrm{rec}\,Y.F'\{Y/G[\mathrm{rec}\,Z.G']\}]\}]$$

7. The term t can be rewritten by $\rightarrow_{A_p}$.

This can only be the case of a redex for $\rightarrow_{A_p}$ which contains another redex for $\rightarrow_{A_p}$, as in the following diagram:

$$t[\mathrm{rec}\,X.\tau.(E[\mathrm{rec}\,Y.\tau.F])]$$

(on the outermost redex) $\downarrow_{A_p}$ $\qquad$ $\downarrow_{A_p}$ (on the innermost redex)

$$t[\tau.\mathrm{rec}\,X.E[\mathrm{rec}\,Y.\tau.F]\{\tau.X/X\}] \qquad t[\mathrm{rec}\,X.\tau.(E[\tau.\mathrm{rec}\,Y.F\{\tau.Y/Y\}])]$$

$$\downarrow_{A_p} \qquad\qquad \downarrow_{A_p}$$

$$t[\tau.\mathrm{rec}\,X.E[\tau.\mathrm{rec}\,Y.F\{\tau.Y/Y\}]\{\tau.X/X\}]$$

References

[1] J.A. Bergstra and J.W. Klop, A complete inference system for regular processes with silent moves, in: *Proc. Logic Colloquium 86* (North Holland, Amsterdam, 1988) 21–81.

[2] B. Courcelle, G. Kahn and J. Vuillemin, Algorithmes d'equivalence et de reduction a des expressions minimales dans une classe d'équations recursives simples, in: *Proc. 2nd Internat. Colloq. on Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol. 14 (Springer, Berlin, 1974) 200–213.

[3] R. De Nicola and M. Hennessy, Testing equivalences for processes, *Theoret. Comput. Sci.* **34** (1984) 83–133.

[4] R. De Nicola, P. Inverardi and M. Nesi, Using the axiomatic presentation of behavioural equivalences for manipulating CCS specifications, in: *Proc. Workshop on Automatic Verification Methods for Finite State Systems*, Lecture Notes in Computer Science, Vol. 407 (Springer, Berlin, 1990) 54–67.

[5] N. Dershowitz and J.-P. Jouannaud, Rewrite systems, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. B: Formal Models and Semantics* (Elsevier Amsterdam, 1990) 243–320.

[6] N. Dershowitz and J.-P. Jouannaud, Notations for rewriting, *Bull. European Assoc. Theoret. Comput. Sci.* **43** (1991) 162–172.

[7] N. Dershowitz, S. Kaplan and D.A. Plaisted, Rewrite, Rewrite, Rewrite, Rewrite, Rewrite ..., *Theoret. Comput. Sci.* **83** (1) (1991) 71–96.

[8] M. Hennessy and R. Milner, Algebraic laws for nondeterminism and concurrency, *J. Assoc. Comput. Mach.* **32** (1) (1985) 137–161.

[9] H. Hussmann, Nondeterministic Algebraic Specifications, Ph.D. Thesis, University of Passau, English Literal Translation as TUM-19104, March 1991.

[10] P. Inverardi and M. Nesi (1990), A rewriting strategy to verify observational congruence, *Inform. Process. Lett.* **35** (1990) 191–199.

[11] P. Inverardi and M. Nesi, Infinite normal forms for non-linear term rewriting systems, in: *Proc. 16th Internat. Symp. on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, Vol. 520 (Springer, Berlin, 1991) 231–239; full version to appear in *Theoret. Comput. Sci.*.

[12] P. Inverardi and C. Priami, Evaluation of tools for the analysis of communicating systems, *Bull. European Assoc. Theoret. Comput. Sci.* **45** (1991) 158–185.

[13] E. Madelaine, Verification tools from the CONCUR project, *Bull. European Assoc. Theoret. Comput. Sci.* **47** (1992) 110–128.

[14] J. Meseguer, Conditioned rewriting logic as a unified model of concurrency, *Theoret. Comput. Sci.* **96** (1) (1992) 73–156.

[15] R. Milner, *A Calculus of Communicating Systems*, Lecture Notes in Computer Science, Vol. 92 (Springer, Berlin, 1980).

[16] R. Milner, Lectures on a calculus for communicating systems, in: M. Broy. ed., *Proc. Control Flow and Data Flow: Concepts of Distributed Programming*, NATO ASI Series, Vol. F14 (Springer, Berlin, 1985) 205–228.

[17] R. Milner, A complete axiomatization for observational congruence of finite-state behaviours, *Inform. and Comput.* **81** (1989) 227–247.
[18] R. Milner, *Communication and Concurrency* (Prentice-Hall, London, 1989).
[19] *Proc. of the 3rd Workshop on Computer Aided Verification*, Lecture Notes in Computer Science, Vol. 575 (Springer, Berlin, 1992).
[20] D. Taubner, A note on the notation of recursion in process algebras, *Inform. Process. Lett.* **37** (1991) 299–303.
[21] R.J. van Glabbeek and W.P. Weijland, Branching time and abstraction in bisimulation semantics, in: *Proc. IFIP 11th World Computer Congress*, San Francisco (1989).
[22] Z. Ariola and J.W. Klop, Cyclic lambda rewriting, in: *Proc. LICS 94*, Paris (IEEE, 1994) 416–425.
[23] Z. Ariola and J.W. Klop, Equational term rewriting, Tech. Report, CWI, Amsterdam, to appear in *Fund. Inform.*