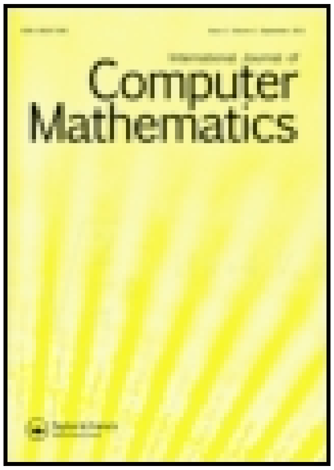

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

A note on causal trees and their applications to CCS

Stefano Basagni [a]
[a] Erik Jonsson School of Engineering and Computer Science , The University of Texas at Dallas , Richardson, TX 75080 E-mail:
Published online: 20 Mar 2007.

Intern. J. Computer Math., Vol. 71, pp. 137–159
Reprints available directly from the publisher
Photocopying permitted by license only

A NOTE ON CAUSAL TREES AND THEIR APPLICATIONS TO CCS

STEFANO BASAGNI*

Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, 2601 N. Floyd Rd., Richardson, TX 75080

(Received 19 March 1998)

Causal Trees are a variant of Milner's Synchronization Trees which aims at reconciling two antagonistic views of semantics for concurrent systems: the interleaving models and the truly concurrent ones. The original model of Causal Trees provides us with an interleaving description of a concurrent system which faithfully expresses *causality* by enriching the action labels of a synchronization tree. These enriched labels supply indication of the *observable* causes of *observable* actions. In this note we revise the original model of Causal Trees, so that *every* action label bears the casual indication, and not only the observable actions. This permits to inherit *all* the results of Milner's original theory in a natural way.

Keywords: Semantics of concurrent systems, interleaving models, truly concurrent models, causal trees, CCS

C. R. Categories: D.1.3, D.2.1, D.3.1, F.1.2, F.3.2

1. INTRODUCTION

In this note, we are interested in revising the origin model of *Causal Trees* as introduced in [5] so that they fit completely Milner's original theory on CCS and Synchronization Trees. Since many papers and books have been published so far on this seminal theory for representing concurrent systems, and since giving a survey on the theory itself is beyond the scope of this note, we refer the reader to basic publications like [8, 5, 10] for further details.

It is well known that Synchronization Trees [8] do not convey enough information to express the *causal* relations among the actions of a

*e-mail: basagni@utdallas.edu

concurrent system. In order to overcome this problem, in [5] Darondeau and Degano presented a variant of Mimer's model in which the labels of the *observable* actions carry additional information about causes: a label is now a pair consisting of (the name of) an action and a structure that indicates the actions which caused the current one. The causes are divided into *direct* and *inherited* ones and are encoded as integers which act as backward pointers to the earlier arcs (*i.e.*, actions) which caused the action related to the current arc.

The aim of this note is to simplify that original idea in two main directions:

- We let also the *invisible* action bears the casual information;
- We redefine the original causal transition system by simplifying the causal part of the label to a set of integers, following [6]. In this way no distinction is made between direct and inherited causes.

The first choice is needed to follow as closely as possible Milner's original work. Indeed, in [8], in the definition of the Strong Bisimulation relation, the invisible action τ is considered as every other action and so, in the causal setting, it has to bear its casual information. This easily allows us to obtain a "natural" definition for the Causal Strong Bisimulation relation (Section 2) and to define an operational semantics for the "causal" calculus (Section 3) which really shapes as Milner's, up to causal information. The second improvement simplifies the treatment of causes in order to clearly express their upgrading when applying the basic operators (Section 4).

The example depicted in Figure 1 shows how the two CCS terms $t_1 = \alpha.\beta.\mathbf{nil} + \beta.\alpha.\mathbf{nil}$ and $t_2 = \alpha.\mathbf{nil} \| \beta.\mathbf{nil}$ – which would be undistinguished in the model of Synchronization Trees and *also* in the original model of

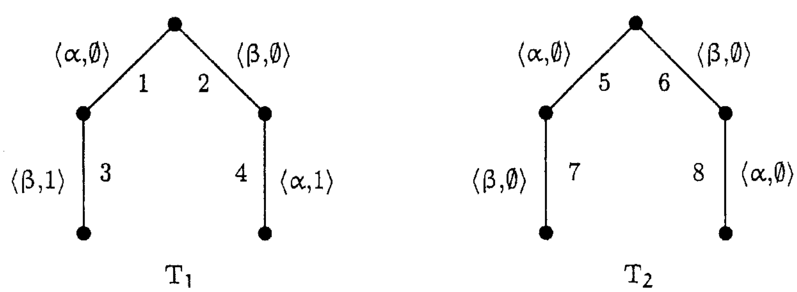

FIGURE 1 The causal trees T_i, originated by the CCS terms t_i $(i = 1, 2)$.

Causal Trees in the case that $\alpha = \tau$ or $\beta = \tau$ – have different representations in our model of Causal Trees. For the sake of readability, we have numbered the arcs and we have omitted the brackets when the causes are singletons. Arc 1 has label $\langle \alpha, \emptyset \rangle$, thus it has no cause in the tree (just as for arcs 2, 5 and 6). More interesting, the same is true for arcs 7 and 8: since they have the empty set as causes, the corresponding events do not depend on their immediate predecessors. On the contrary, arcs 3 and 4 do depend on 1 and 2, respectively, since they "point back" to them. Thus, the different causal relations originating from the above processes are completely reflected in the corresponding causal trees.

In this note we define a semantics for CCS following the paradigm described in [7]. Specifically:

1. The evolution of a system represented by a CCS term is described in a syntax driven way (Section 3). Each *computation, i.e.*, each path of the transition system, is assigned a tree.
2. The arcs of the trees are labeled by an *observation* which takes care of causality (Causal Trees).
3. Causal trees are compared *via* bisimulations which are now defined according to the new labeling (Section 6).

Furthermore, in Section 4, we introduce a new algebra of causal trees ($\mathcal{CT}$) and an interpretation for CCS which will allow us to consider $\mathcal{CT}$ as a convenient semantic domain. In this sense, we meet the requirement 1′) of [7], that asks for the definition of a causal denotational model for CCS. The "spirit" of Causal Trees relies on the interpretation of the parallel command which yields an expansion law which is interleaving in its shape, though causal in its essence, thus reconciling the two classical antagonistic views of semantics for concurrent systems.

In Section 5, $\mathcal{CT}$ is proved *fully adequate* with respect to the operational setting. This result allows us to closely follow the classical theory [9, 10] in achieving a sound and complete system of equational axioms for causal observational congruence over (finite) CCS programs (Section 6).

As a final note, we point out that the new model of Causal Tress introduced in this note can be considered a truly concurrent model for concurrent systems, despite its "interleaving structure." Indeed, in [1] the equivalence of $\mathcal{CT}$ with three truly concurrent operational models for CCS (namely, the model of Flow Event Structures [3], the Flow Nets model [2], and the model of Proved Transition Systems [4], which are compared in [4]) is established. This is naturally obtained due to the new definition of causal trees given in this note.

2. CAUSAL TREES

We start by defining the class of Causal Trees. In what follows, $\mathbf{N}^+$ will be the set of non-zero natural numbers and K will range over the finite sets of $\mathcal{P}(\mathbf{N}^+)$. We let $\mathbf{N} = \mathbf{N}^+ \vee \{0\}$ and assume as given a fixed set of actions $\mathcal{A}$ made up of *names*, *conames* and the *silent action* τ, as in standard CCS (*i.e.*, $\mathcal{A} = \Delta \cup \bar{\Delta} \cup \{\tau\}$).

DEFINITION 2.1 (Causal Trees) Let $\mathcal{L} = \{\langle \mu, K \rangle | \mu \in \mathcal{A}\}$ and $\mathcal{ST}(\mathcal{A})$ be the set of synchronization trees labeled on $\mathcal{A}$. A causal tree $T \in \mathcal{CT}(\mathcal{A})$ is the synchronization tree $T \in \mathcal{ST}(\mathcal{L})$. The empty tree is **NIL**. ◇

If n and n' are nodes of a causal tree T, the notation $n \xrightarrow{z} n'$ means that there exists an arc labelled by $z = \langle \mu, K \rangle$ from n to n'. Thus we can introduce a notion of equivalence over Causal Trees [6].

DEFINITION 2.2 (Causal Strong Bisimulation) Two causal trees T and T' are causally strongly bisimilar if and only if there exists a relation $\sim_s$ on the nodes of T and T' such that:

i. if r and r' are the roots of T and T', then $r \sim_s r'$;
ii. $n \sim_s m$ if and only if
   1. when $n \xrightarrow{z} n'$ then there exists a node m' such that $m \xrightarrow{z} m'$ and $n' \sim_s m'$;
   2. symmetric of 1 ◇

As usual, we will consider the maximal causal strong bisimulation only, also denoted by $\sim_s$.

Causal strong bisimulation, $\sim_s$, on finite causal trees is a congruence and its inducing axiomatization is completely standard, providing that x, y and z are in $\mathcal{CT}(\mathcal{A})$. Indeed, the four axioms listed below reflect the tree structure of computations and are straightforwardly label independent.

$$\text{(A1)}\ x + y = y + x; \quad \text{(A2)}\ x + (y + z) = (x + y) + z;$$
$$\text{(A3)}\ x + x = x; \qquad \text{(A4)}\ x + \mathbf{NIL} = x.$$

Clearly, causal strong bisimulation allows one to make less identifications than the interleaving strong bisimulation [8], because the latter ignores causes. Indeed, terms t_1 and t_2 in the Introduction are strongly bisimilar but they are not causally bisimilar (the two notions only coincide when the considered trees represent sequential non-deterministic processes). *It is worth noticing that the same is true if at least one of the actions is the silent* *action* τ. This was not the case for the model discussed in [5], in which the two processes were causally identified (see also [6]).

It is easy to define a "cause erasing" morphism between causal trees and synchronization trees, $\Psi : \mathcal{CT}(\mathcal{A}) \rightarrow \mathcal{ST}(\mathcal{A})$. The *kernel* of this mapping operates on labels leaving untouched the structure of trees, and will be the function $\psi : \mathcal{L} \rightarrow \mathcal{A}$ such that $\psi(\langle\mu,K\rangle) = \mu$.

If we consider causal trees up to causal strong bisimulation, we can denote them as terms of a language with operations $\langle\mu, K\rangle\cdot$ of prefixing and $+$ of non-deterministic choice (with the empty tree $\mathbf{NIL} \equiv \sum_{i\in\emptyset}\langle\mu_i, K_i\rangle\cdot T_i$ as neutral element). Thus, we obtain *normal forms* for causal trees, which have the following pattern:

$$T \equiv \sum_{i\in I}\langle\mu_i,\ K_i\rangle\ \cdot\ T_i.$$

A causal tree bears in itself (in each of its paths) all the information needed to derive a partial ordering of events. An event e, labeled by μ, will be generated in correspondence to every arc e, labeled by $\langle\mu, K\rangle$, of the given path. The event e will be greater than all the events corresponding to arcs pointed back by the pointers in K. Thus, every arc of a causal tree may be interpreted as the incremental description of a partially ordered multiset of actions.

Example Consider the causal tree T in Figure 2(a) originated by the CCS process t defined by $\alpha.(\beta.\delta.\gamma.\mathbf{nil}\|\rho.\bar{\delta}.\mathbf{nil})\backslash\delta\ +\ \zeta.\sigma.\vartheta.\mathbf{nil}$ according to the CCS algebra $\mathcal{CT}$ for Causal Trees (see below). Paths p_1, p_2 and p_3 generate the partial ordering depicted in Figure 2(b), where events are identified with their labels. Branches p_1 and p_2 give rise to the two isomorphic partial orderings in the left side. Indeed, consider the left path made of arcs from 1 to 5. It originates five events labeled by α, β, ρ, τ and γ partially ordered by (the reflexive and transitive closure of) $1 \leq 2$ (because the label of 2 is $\langle\beta, 1\rangle$ and arc 1 immediately precedes 2 in the given path), $1 \leq 3$, $1, 2, 3 \leq 4$ and $1, 2, 3, 4 \leq 5$. Isomorphically for the path p_2. The right hand side of Figure 2(b) is self-explanatory. $\diamond$

A tree with arcs (or nodes) labeled by partial orderings can be thus immediately recovered from a causal tree, though such a tree would provide us with an *integral* description of a concurrent system, while a causal tree gives a *differential* one, the latter being more economic. This approach allows us to define truly concurrent bisimulations in the standard way (Section 6).

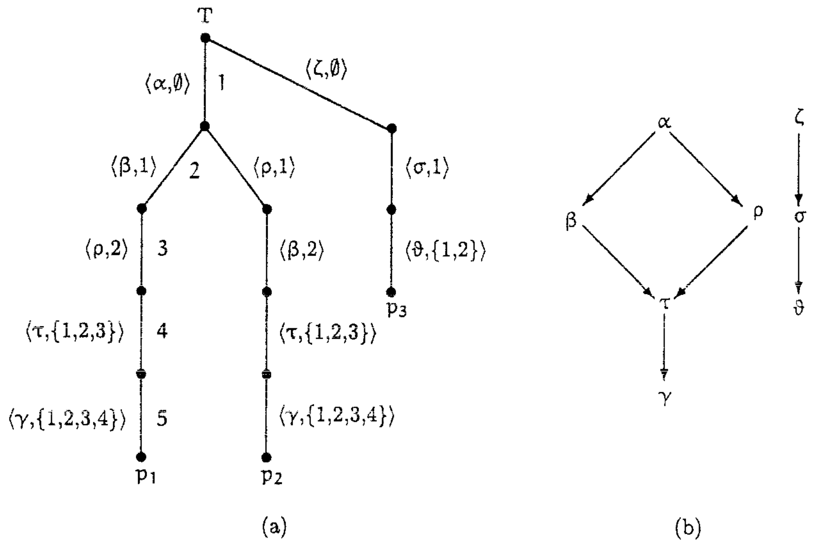

FIGURE 2 The causal tree T originated by the process t and the partial ordering represented by T.

3. A CAUSAL CALCULUS OF COMMUNICATING SYSTEMS

In this section we aim at defining causal trees induced by CCS programs *via* the method of *Structured Operational Semantics* [11]. To this purpose we embed CCS into a wider set of terms: CCCS (a Causal CCS, C^3S for short). The transition system defined on C^3S is a direct extension of the original system for CCS: the definition of CCS is retrieved from the definition of C^3S by merely erasing all indications about causes.

DEFINITION 3.1 Let $\Sigma = \bigcup_{i \in \mathbf{N}} \Sigma_i$ be the signature of CCS, where

$$\begin{aligned}
\Sigma_0 &= \{\mathbf{nil}\};\\
\Sigma_1 &= \{\mu.|\mu \in \mathcal{A}\} \cup \{\backslash\alpha|\alpha \in \Delta\} \cup \{\beta/\alpha|\alpha,\, \beta \in \Delta\};\\
\Sigma_2 &= \{+,\, \|\};\\
\Sigma_n &= \emptyset, \quad n \geq 3.
\end{aligned} \qquad \diamond$$

Each Σ_i, contains functional symbols having *arity* i. Let REC(Σ, χ) be the set of recursive Σ-terms over χ (a set of variables) which satisfy the Greibach condition of well-guardedness (with respect to guarding operators μ). Terms have possible forms $x(\in \chi)$ or **rec** $x.t$ or $f(t_1,\ldots,t_n$ for $f \in \Sigma_n$, $(n \geq 0)$. The

CCS programs are the elements of CREC (Σ, χ), the subset of the closed terms in REC(Σ, χ).

A generic C^{3}S term is created from a CCS term by means of the binary (infix) operator $\Rightarrow$ which prefixes terms $t \in$ CREC(Σ, χ) by a finite set K of natural numbers whose intended meaning is to indicate, at each step of a C^{3}S derivation, the activating causes of all the active terms and subterms, given by backward references to the past of the derivation. By CCREC(Σ, χ) we will indicate the set of CCS terms t prefixed by a set of causes K by means of operator $\Rightarrow: K \Rightarrow t$. These will be the generators of C^{3}S.

DEFINITION 3.2 The C^{3}S signature is given by $\Sigma^* = \bigcup_{i \in N} \Sigma_i^*$, where

$$\begin{aligned}
\Sigma_0^* &= \mathrm{CCREC}(\Sigma, \chi); \\
\Sigma_1^* &= \{\backslash\alpha | \alpha \in \Delta\} \cup \{\beta/\alpha | \alpha, \beta \in \Delta\}; \\
\Sigma_2^* &= \{+, \|\}, \text{ and} \\
\Sigma_n^* &= \emptyset, \quad n \geq 3.
\end{aligned} \qquad \diamond$$

Thus, C^{3}S is the family of $\Sigma \backslash (\{\mathbf{nil}\} \cup \{\mu. | \mu \in \mathcal{A}\})$-terms over generators $K \Rightarrow t$. (From the definition of the signature of classical CCS we have dropped the prefix operator and the inactive process **nil**.) A typical C^{3}S term, with causes $K_i (i = 1,2)$, attached to all outermost occurrences of guarding operators and recursion symbols, is $K_1 \Rightarrow \lambda.\mathbf{nil} \| K_2 \Rightarrow \mathbf{rec}\, x.t$. We assume henceforth that operator $\Rightarrow$ distributes over all operators in $\Sigma \backslash (\{\mathbf{nil}\} \cup \{\mu. | \mu \in \mathcal{A}\})$, so that C^{3}S terms are reducible to that canonical form.

A natural way to embed CCS in C^{3}S is by prefixing a generic term with the empty set, thus indicating that the actions of t have no activating causes, $\emptyset \Rightarrow t$. The transition rules for C^{3}S define formally this intuition, and show exactly how, in the various cases, the sets of causes have to be updated. For such an upgrade we introduce the followaing operators:

$$\delta: \mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+), \qquad \delta(K) = \{k + 1 | k \in K\};$$

$$\eta: \mathcal{P}(\mathbf{N}^+) \times \mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+), \quad \eta(H, K) = \begin{cases} H \cup K & \text{if } 1 \in K, \\ K & \text{otherwise.} \end{cases}$$

The operator $\delta(K)$ simply increases by one every element of a nonempty set K. The operator $\eta(H, K)$ joins the sets H and K only if $1 \in K$. We extend δ and μ on C^{3}S terms as follows:

$$\delta(K \Rightarrow t) \equiv (\delta(K) \Rightarrow t);$$
$$\eta(H, (K \Rightarrow t)) \equiv (\eta(K, H) \Rightarrow t).$$

The C^3S labelled transition system relation, defined in the S.O.S. style [11], is denoted by $\xrightarrow{z}$, where $z = \langle \mu, K \rangle$ is a label from the set $\mathcal{L}$. For such a relation we understand that the restriction and relabeling operators act only on the first component of the label $\langle \mu, K \rangle$, as in standard CCS, *e.g.* $\langle \alpha, K \rangle \backslash \beta$ is defined as $\langle \alpha, K \rangle$ only when $\alpha \neq \beta$.

In what follows, e, f, e' and f' will range over C^3S terms and t will be a generic CCS term.

DEFINITION 3.3 (C^3S transitions)

$$\mathit{Act} \quad K \Rightarrow \mu.t \xrightarrow{\mu,K} (\{1\} \cup \delta(K) \Rightarrow t)$$

$$\mathit{Sum}1 \ \frac{e \xrightarrow{z} e'}{e + f \xrightarrow{z} e'} \qquad \mathit{Sum}2 \ \frac{f \xrightarrow{z} f'}{e + f \xrightarrow{z} f'}$$

$$\mathit{Res} \ \frac{e \xrightarrow{z} e'}{e\backslash\alpha \xrightarrow{z\backslash\alpha} e'\backslash\alpha},\ z\backslash\alpha \text{ defined} \qquad \mathit{Rel} \frac{e \xrightarrow{z} e'}{e[\beta/\alpha] \xrightarrow{z[\beta/\alpha]} e'[\beta/\alpha]}$$

$$\mathit{Asyn}1 \ \frac{e \xrightarrow{z} e'}{e \| f \xrightarrow{z} e' \| \delta(f)} \qquad \mathit{Asyn}2 \ \frac{f \xrightarrow{z} f'}{e \| f \xrightarrow{z} \delta(e) \| f'}$$

$$\mathit{Syn\ rule} \ \frac{e \xrightarrow{\lambda,K} e' f \xrightarrow{\bar{\lambda},K} f'}{e \| f \xrightarrow{\tau,K \cup K'} \eta(\delta(K'), e') \| \eta(\delta(K), f')}$$

$$\mathit{Rec} \ \frac{(K \Rightarrow (t[\mathbf{rec}\, x.t/x])) \xrightarrow{z} e}{(K \Rightarrow (\mathbf{rec}\, x.t)) \xrightarrow{z} e}$$

◇

Some comments are in order.

- Axiom *Act* allows for the autonomous firing of a guard μ of a CCS term $\mu.t$: the direct cause of activation (of the actions) of the residual term t will be 1 (referring to μ). This is obtained by adding the singleton $\{1\}$ to the set of the activating causes of t (*i.e.*, the action that directly causes all the possible actions of the subterm t is only a step behind in the "execution"). The *hereditary* causes, *i.e.*, the references to actions that are "causes of the direct cause", are all incremented by 1 (we have to do a further step in the past of the derivation to retrieve those causes, after μ occurred): this is the task of δ which is applied to the set of activating causes of $\mu.t$.
- Rule *Asyn*1 allows a C^3S term e, which acts in parallel with a term f, to autonomously evolve in a term e' by an action labelled by $\langle \mu, K \rangle$. *The whole term* $e \| f$ will evolve with the same action in a term, $e' \| \delta(f)$, in

CCS programs are the elements of CREC (Σ, χ), the subset of the closed terms in REC(Σ, χ).

A generic C^{3}S term is created from a CCS term by means of the binary (infix) operator $\Rightarrow$ which prefixes terms $t \in$ CREC(Σ, χ) by a finite set K of natural numbers whose intended meaning is to indicate, at each step of a C^{3}S derivation, the activating causes of all the active terms and subterms, given by backward references to the past of the derivation. By CCREC(Σ, χ) we will indicate the set of CCS terms t prefixed by a set of causes K by means of operator $\Rightarrow$:$K \Rightarrow t$. These will be the generators of C^{3}S.

DEFINITION 3.2 The C^{3}S signature is given by $\Sigma^* = \bigcup_{i \in N} \Sigma_i^*$, where

$$\begin{aligned}
\Sigma_0^* &= \text{CCREC}(\Sigma, \chi);\\
\Sigma_1^* &= \{\backslash\alpha | \alpha \in \Delta\} \cup \{\beta/\alpha | \alpha, \beta \in \Delta\};\\
\Sigma_2^* &= \{+, \|\}, \text{ and}\\
\Sigma_n^* &= \emptyset, \quad n \geq 3.
\end{aligned}$$

$\diamond$

Thus, C^{3}S is the family of $\Sigma\backslash(\{\mathbf{nil}\} \cup \{\mu.|\mu \in \mathcal{A}\})$-terms over generators $K \Rightarrow t$. (From the definition of the signature of classical CCS we have dropped the prefix operator and the inactive process **nil**.) A typical C^{3}S term, with causes $K_i (i = 1,2)$, attached to all outermost occurrences of guarding operators and recursion symbols, is $K_1 \Rightarrow \lambda.\mathbf{nil} \| K_2 \Rightarrow \mathbf{rec}\, x.t$. We assume henceforth that operator $\Rightarrow$ distributes over all operators in $\Sigma\backslash(\{\mathbf{nil}\} \cup \{\mu.|\mu \in \mathcal{A}\})$, so that C^{3}S terms are reducible to that canonical form.

A natural way to embed CCS in C^{3}S is by prefixing a generic term with the empty set, thus indicating that the actions of t have no activating causes, $\emptyset \Rightarrow t$. The transition rules for C^{3}S define formally this intuition, and show exactly how, in the various cases, the sets of causes have to be updated. For such an upgrade we introduce the followaing operators:

$$\delta: \mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+), \qquad \delta(K) = \{k + 1 | k \in K\};$$

$$\eta: \mathcal{P}(\mathbf{N}^+) \times \mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+), \quad \eta(H, K) = \begin{cases} H \cup K & \text{if } 1 \in K,\\ K & \text{otherwise.}\end{cases}$$

The operator $\delta(K)$ simply increases by one every element of a nonempty set K. The operator $\eta(H, K)$ joins the sets H and K only if $1 \in K$. We extend δ and μ on C^{3}S terms as follows:

$$\delta(K \Rightarrow t) \equiv (\delta(K) \Rightarrow t);$$
$$\eta(H, (K \Rightarrow t)) \equiv (\eta(K, H) \Rightarrow t).$$

The C^{3}S labelled transition system relation, defined in the S.O.S. style [11], is denoted by $\xrightarrow{z}$, where $z = \langle \mu, K \rangle$ is a label from the set $\mathcal{L}$. For such a relation we understand that the restriction and relabeling operators act only on the first component of the label $\langle \mu, K \rangle$, as in standard CCS, *e.g.* $\langle \alpha, K \rangle \backslash \beta$ is defined as $\langle \alpha, K \rangle$ only when $\alpha \neq \beta$.

In what follows, e, f, e' and f' will range over C^{3}S terms and t will be a generic CCS term.

Definition 3.3 (C^{3}S transitions)

$$Act \quad K \Rightarrow \mu.t \xrightarrow{\mu,K} (\{1\} \cup \delta(K) \Rightarrow t)$$

$$Sum1 \; \frac{e \xrightarrow{z} e'}{e + f \xrightarrow{z} e'} \qquad Sum2 \; \frac{f \xrightarrow{z} f'}{e + f \xrightarrow{z} f'}$$

$$Res \; \frac{e \xrightarrow{z} e'}{e\backslash\alpha \xrightarrow{z\backslash\alpha} e'\backslash\alpha}, \; z\backslash\alpha \text{ defined} \qquad Rel \frac{e \xrightarrow{z} e'}{e[\beta/\alpha] \xrightarrow{z[\beta/\alpha]} e'[\beta/\alpha]}$$

$$Asyn1 \; \frac{e \xrightarrow{z} e'}{e \| f \xrightarrow{z} e' \| \delta(f)} \qquad Asyn2 \; \frac{f \xrightarrow{z} f'}{e \| f \xrightarrow{z} \delta(e) \| f'}$$

$$Syn\ rule \; \frac{e \xrightarrow{\lambda,K} e' f \xrightarrow{\bar{\lambda},K} f'}{e \| f \xrightarrow{\tau, K \cup K'} \eta(\delta(K'), e' \| \eta(\delta(K), f')}$$

$$Rec \; \frac{(K \Rightarrow (t[\mathbf{rec}\, x.t/x])) \xrightarrow{z} e}{(K \Rightarrow (\mathbf{rec}\, x.t)) \xrightarrow{z} e}$$

◇

Some comments are in order.

- Axiom *Act* allows for the autonomous firing of a guard μ of a CCS term $\mu.t$: the direct cause of activation (of the actions) of the residual term t will be 1 (referring to μ). This is obtained by adding the singleton $\{1\}$ to the set of the activating causes of t (*i.e.*, the action that directly causes all the possible actions of the subterm t is only a step behind in the "execution"). The *hereditary* causes, *i.e.*, the references to actions that are "causes of the direct cause", are all incremented by 1 (we have to do a further step in the past of the derivation to retrieve those causes, after μ occurred): this is the task of δ which is applied to the set of activating causes of $\mu.t$.
- Rule *Asyn*1 allows a C^{3}S term e, which acts in parallel with a term f, to autonomously evolve in a term e' by an action labelled by $\langle \mu, K \rangle$. *The whole term* $e \| f$ will evolve with the same action in a term, $e' \| \delta(f)$, in

which f has all its causes incremented by 1, as it has "lost a tick." Indeed, the action performed by e and labeled by $\langle \mu, K \rangle$:

- does not cause any of the actions that f can fire (and then the action label has to be skipped by the backward references in f), and
- increases by 1 the number of "back steps" required to retrieve the actions that are causes of the possible transitions of f.

Notice that, when $K = \emptyset$, the definition of the δ operator is such that K is still $\emptyset$. This corresponds to the intuition that if all possible actions of a (sub)term are independent of any other action, they remain autonomous when such an action occurs and then they continue to have no activating causes ($K = \emptyset$). Rule *Asyn2* is symmetric.

– In the case of a synchronization (*Syn rule*) the causes of the invisible action will be the union of the causes of the terms involved in the communication. We must take particular care to realize the inheritance of causes: the causes K of an action performed by e and labeled by $\langle \lambda, K \rangle$ (see *Syn rule* in the previous definition), increased by 1 to keep track of the τ, must be merged with the causes of the action labeled by $\langle \bar{\lambda}, K' \rangle$. Henceforth, if there is a parallel command in f then $\delta(K)$ will be added only to the set of causes of the subterm of f which actually evolved with the action labeled by $\langle \bar{\lambda}, K' \rangle$ (this set will be $\delta(K') \cup \{1\}$). This is reflected by the operator η which permits such a kind of "cross inheritance" of causes. The argument on f is symmetric.

We are now able to establish in an obvious way a correspondence between C^3S and CCS derivations. It may be observed that C^3S transitions $e \xrightarrow{\mu, K} e'$ are connected to CCS transitions $t \xrightarrow{\mu} t'$ by the following implications:

$$\frac{e \xrightarrow{\mu, K} e'}{\Psi(e) \xrightarrow{\mu} \Psi(e')} \qquad \frac{t \xrightarrow{\mu} t'}{\exists e, e' : \Psi(e) = t,\ \Psi(e') = t' \wedge \exists K : e \xrightarrow{\mu, K} e'}$$

in which the cause erasing function $\Psi : C^3S \rightarrow CCS$ is easily defined starting from $\Psi(K \Rightarrow t) = t$ and proceeding homomorphically by induction on $\Sigma \backslash (\{\mathbf{nil}\} \cup \{\mu. | \mu \in \mathcal{A}\})$.

For e in C^3S, the operational meaning of e is standard. It is the causal tree $[e]_{\mathrm{op}}$ obtained by unfolding from root e the transition system $\{e' \xrightarrow{z} e'' | z \in \mathcal{L}\}$, written as

$$[e]_{\mathrm{op}} = \sum_{i \in I} \langle \mu_i, K_i \rangle \cdot [e_i]_{\mathrm{op}}$$

if and only if $e \xrightarrow{\mu_i, K_i} e_i$.

For example:

$$[K \Rightarrow \mathbf{nil}]_{\mathrm{op}} = \mathbf{NIL} \quad \text{and}$$
$$[K \Rightarrow (\alpha.\beta.\mathbf{nil} + \beta.\alpha.\mathbf{nil})]_{\mathrm{op}} = \langle \alpha, K \rangle \cdot \langle \beta, \{1\} \cup \delta(K) \rangle \cdot \mathbf{NIL} + \langle \beta, K \rangle \cdot \langle \alpha, \{1\} \cup \delta(K) \rangle \cdot \mathbf{NIL}.$$

For t in CCS, the corresponding causal tree will be $[\emptyset \Rightarrow t]_{\mathrm{op}}$.

In order to define the causal strong equivalence we obviously extend Milner's approach so as to consider causes. As for the interleaving case, in the causal strong bisimulation the invisible actions (labeled by $\langle \tau, K \rangle$) are considered in the same way as the others ($\langle \lambda, K \rangle$, $\lambda \neq \tau$).

DEFINITION 3.4 (Causal strong bisimulation) Two C^3S terms e and f are causally strongly bisimilar, written $e \sim_s f$, if and only if for each $z \in \mathcal{L}$,

i. whenever $e \xrightarrow{z} e'$ then there exists f' such that $f \xrightarrow{z} f'$ and $e' \sim_s f'$,
ii. whenever $f \xrightarrow{z} f'$ then there exists e' such that $e \xrightarrow{z} e'$ and $e' \sim_s f'$. $\diamond$

It is worth noticing that by "erasing causes" (*i.e.*, applying the function Ψ to the derivations of the previous definition) we will obtain the definition of the interleaving case.

Causal strong equivalence will be the maximal causal strong bisimulation, also denoted by $\sim_s$. Two CCS terms will be causally strongly bisimilar if the corresponding C^3S terms are such:

DEFINITION 3.5 Let t_1 and t_2 be two CCS terms. Then

$$t_1 \sim_s t_2 \text{ if and only if } (\emptyset \Rightarrow t_1) \sim_s (\emptyset \Rightarrow t_2). \qquad \diamond$$

Once provided the right methods to treat causes, causal strong equivalence can be proved to satisfy all the properties that hold for interleaving case (*e.g.*, the property of being a *congruence, etc.*). Tools and detailed discussion are in [1].

4. $\mathcal{CT}$: A NEW ALGEBRA OF CAUSAL TREES

4.1. Combinatory Operators

In order to upgrade the set $\mathcal{CT}(\mathcal{A})$ of causal trees into an interpretation for C^iS(i = 2, 3) programs we introduce *via* axiomatic definition three indexed families of basic combinatory operators on causes and causal trees. These auxiliary operators will be used to upgrade, in an inductive way, the causes after an algebraic operation has been performed.

The first operator serves primarily to manipulate causes in prefixing trees by actions. Every arc at depth p of a causal tree T prefixed by a new arc will be caused by the newly added one and therefore the set K of its causes has to be augmented of a pointer (the non-zero natural p) to the new arc. This results in $K \cup \{p\}$. The family of operators $\triangleleft . \triangleright_n : \mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+)$, indexed on $\mathbf{N}$, will take care of such an updating of causes and is defined as the unique family of unary operators on $\mathcal{CT}(\mathcal{A})$ satisfying the following well-guarded recursive formula:

$$\triangleleft H \triangleright_n T \equiv \Sigma_{i \in I} \langle \mu_i, \triangleleft H \triangleright_n K_i \rangle \cdot \triangleleft H \triangleright_{n+1} T_i,$$

where

$$\triangleleft H \triangleright_n K = \begin{cases} K \cup H & \text{if } n = 0, \\ \triangleleft \delta(H) \triangleright_{n-1} & \text{otherwise.} \end{cases}$$

$\triangleleft H \triangleright_n K$ can be operationally interpreted as increasing the elements of H by n and taking the union of the resulting set with K.

The parallel composition of two trees is a bit involved. As for the expansion law on synchronization trees, it can be divided into three parts: the first (the second, respectively) when an arc from the first (the second, respectively) tree is taken, the third when a communication occurs. The structure of the resulting causal tree is again easy to define, but updating the causes requires attention. In the first case, the second tree has "lost one tick," therefore 1 is to be added to its causes (similarly for the second case). However not all the pointers must be incremented, but only those that point to arcs already "consumed in the inductive step;" the causes "inside" the tree must instead be left untouched. As an example, take an arc at depth n: its causes greater than or equal to n will become $n+1$ for they refer to arcs belonging to the prefix of the result, while the others remain as they are for they point to arcs in the residual part of the tree. This task is performed by the auxiliary operator $[n] : \mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+)$, defined as follows:

$$[n]T \equiv \Sigma_{i \in I} \langle \mu_i, [n]K_i \rangle \cdot [n+1]T_i,$$

where

$$[n]K = \{k + 1 | k \in K, k \geq n\} \cup \{k | k \in K, k < n\}.$$

When a communication occurs, its causes should be the union of the causes of both the communicating arcs, say a and b. Recursively, all the arcs depending on either a or b, should contain the causes of both (properly upgraded). The auxiliary operator $\lceil K/n \rfloor : \mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+)$, implements such a "cross inheritance" of causes:

$$\lceil K/n \rfloor T \equiv \Sigma_{i \in I} \langle \mu_i, \lceil K/n \rfloor K_i \rangle \cdot \lceil \delta(K)/n + 1 \rfloor T_i,$$

where

$$\lceil K/n \rfloor H = \begin{cases} K \cup H & \text{if } n \in H, \\ H & \text{otherwise.} \end{cases}$$

An example of the application of the above operators is depicted in Figure 3.

4.2. The Algebra $\mathcal{CT}$: Axioms and Morphisms

We now turn the set $\mathcal{CT}(\mathcal{A})$ of causal trees into an interpretation of C^{3}S and CCS terms. The algebra of causal trees, $\mathcal{CT}$, is the unique model of the following axioms on carrier $\mathcal{CT}(\mathcal{A})$, with $+$ interpreted as sum of trees:

Init $K \Rightarrow_{\mathcal{CT}} T = \lhd K \rhd_0 T$;
Nil $\mathbf{nil}_{\mathcal{CT}} = \mathbf{NIL}$;

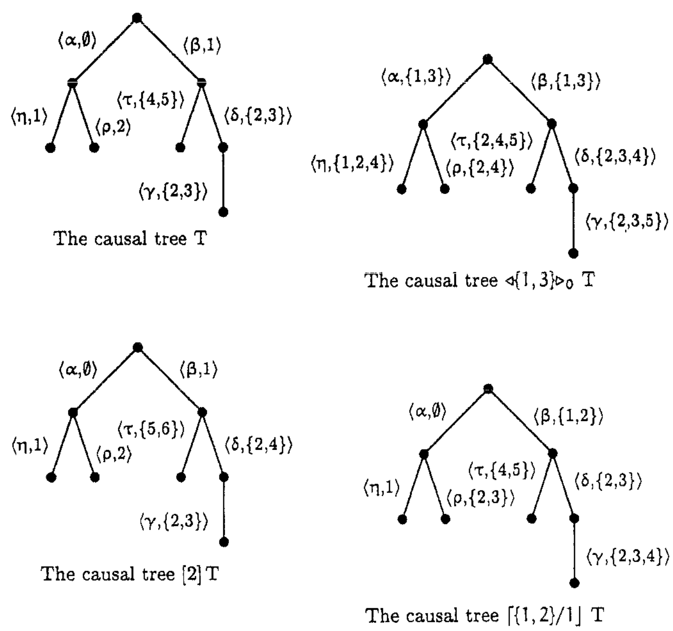

FIGURE 3 The application of the auxiliary operators to the causal tree T.

Pref $\mu_{.CT} T = \langle \mu, \emptyset \rangle \cdot \triangleleft\{1\}\triangleright_0 T$;
Sum $T +_{CT} U = T + U$;
Relab $T[\beta/_{CT}\alpha] = \sum_{i \in I} \langle \mu_i[\beta/\alpha], K_i \rangle \cdot T_i[\beta/_{CT}\alpha]$;
Restrict $T \backslash_{CT} \alpha = \sum_{\mu_i \neq \alpha, \bar{\alpha}} \langle \mu_i, K_i \rangle \cdot T_i \backslash_{CT} \alpha$;
Interleave

$$T \|_{CT} U = \sum_{i \in I} \langle \mu_i, K_i \rangle \cdot (T_i \|_{CT} [1] U) + \sum_{m \in M} \langle \nu_m, K_m \rangle \cdot ([1] T \|_{CT} U_m)$$
$$+ \sum_{\lambda_i = \bar{\rho}_m} \langle \tau, K_i \cup K_m \rangle \cdot (\lceil \delta(K_m)/1 \rfloor T_i \|_{CT} \lceil \delta(K_i)/1 \rfloor U_m),$$

where $U = \sum_{m \in M} \langle \nu_m, K_m \rangle \cdot U_m$.

Some comments are in order.

- The first axiom is used to add the set K, properly upgraded, to the causes of each arc of the causal tree T.
- The third axiom concerns prefixing: by the operator $\triangleleft \cdot \triangleright_0$ applied to $\{1\}$ we add the pointer to the new arc to the sets of causes of all arcs of T. The new arc is now the first of T and it "has no cause."
- Axiom *Interleave* merges two trees by interleaving their actions (the first two sumniands) or by synchronizing them when complementary (last summand). In the first case if the action is μ (respectively ν) then the causal tree U (respectively, T) is delayed by operator [1] which increments by 1 all backward references pointing outside U (respectively T). If complementary actions are synchronized $\lambda_i = \bar{\rho}_m$, resulting in τ, its causes result from the union of the causes of λ_i, and $\bar{\rho}_m$ and by the $\lceil K/n \rfloor$ operator the causes of $\bar{\rho}_m(\lambda_i)$, properly updated, are passed down to the descendant of $\lambda_i(\bar{\rho}_m)$. This axiom may be seen as the causal counterpart of Milner's expansion theorem.

We now interpret C^3S terms in $\mathcal{CT}$.

DEFINITION 4.1

$$[K \Rightarrow t]_{CT} = K \Rightarrow_{CT} [\![t]\!];$$
$$[e' + e'']_{CT} = [e']_{CT} +_{CT} [e'']_{CT};$$
$$[e[\beta/\alpha]]_{CT} = [e]_{CT}[\beta/_{CT}\alpha];$$
$$[e \backslash \alpha]_{CT} = [e]_{CT} \backslash_{CT} \alpha;$$
$$[e' \| e'']_{CT} = [e']_{CT} \|_{CT} [e'']_{CT},$$

where $[\![\ ]\!]$: $\langle \text{CCS}, \Sigma_{\text{CCS}} \rangle \to \langle \mathcal{CT}, \Sigma_{\mathcal{CT}} \rangle$ is the Σ-homomorphism defined as follows:

$$
\begin{aligned}
&[\![\mathbf{nil}]\!] = \mathbf{nil}_{\mathcal{CT}};\\
&[\![\mu, T]\!] = \mu._{\mathcal{CT}}[\![t]\!];\\
&[\![t[\beta/\alpha]]\!] = [\![t]\!][\beta/_{\mathcal{CT}}\alpha];\\
&[\![t\backslash\alpha]\!] = [\![t]\!]\backslash_{\mathcal{CT}}\alpha;\\
&[\![t' + t'']\!] = [\![t']\!] +_{\mathcal{CT}} [\![t'']\!];\\
&[\![t' \| t'']\!] = [\![t']\!] \|_{\mathcal{CT}} [\![t'']\!].
\end{aligned}
\qquad \diamond
$$

The algebra $\langle \mathcal{CT}, \Sigma_{\mathcal{CT}} \rangle$ may be used as an intrepretation for CCS letting **rec** $x.t$ be interpreted as the fixed point at x of the functional interpretation $[\![t]\!]$ of t. Indeed, it will be:

$$
\begin{aligned}
\mathbf{rec}\, x.t &= t[\mathbf{rec}\, x.t/x], \quad \text{hence}\\
[\![\mathbf{rec}\, x.t]\!] &= [\![t[\mathbf{rec}\, x.t/x]]\!].
\end{aligned}
$$

The existence and uniqueness of fixed points are guaranteed by the assumption of the well-guardedness of recursive definitions and by the metric continuity of all operators in $\Sigma \cup \{\Rightarrow\}$. (Those operators are distance preserving, due to the form of the defining axioms, letting the distance between pairs of trees be the maximal distance between trees.)

Since it is easy to show that for each t in CCS and for the corresponding C^3S term $\emptyset \Rightarrow t$ the two mappings $[\![\ ]\!]$ and $[\ ]_{\mathcal{CT}}$ coincide, in the sequel we denote the both of them with $[\![\,]\!]_{\mathcal{CT}}$.

5. FULL ADEQUACY

Here the equality is established between the operational meaning $[e]_{\text{op}}$ and the algebraic meaning $[e]_{\mathcal{CT}}$ of an arbitrary C^3S expression, *i.e.*, the relation $[e]_{\text{op}} = [e]_{\mathcal{CT}}$ which expresses the *full adequacy* of the model $\mathcal{CT}$.

THEOREM 5.1 (*Full adequacy*) *The algebraic model $\mathcal{CT}$ is fully adequate for C^3S.* $\diamond$

A quick comparison between the algebraic axioms for $\mathcal{CT}$ and the logical axioms set for C^3S transitions shows that in order to prove the above theorem all we have to establish is the following series of propositions which imply the existence of normal forms $\sum_{i \in I} z_i \cdot [e_i]_{\mathcal{CT}}$ for causal trees.

From Propositions 5.1 to 5.5 below, both transitions $e \xrightarrow{z} e'$ (between terms) and transitions $[e]_{CT} \xrightarrow{z} [e']_{CT}$ (between trees, defined in Section 2) obey the rules stated for C³S in Section 3. The propositions rely upon the combinatory laws of Appendix A and their proofs may be found in Appendix B.

PROPOSITION 5.1 *For each* $(\textbf{rec}\ x.t) \in CREC(\Sigma, \chi)$ *and for each finite set* K *in* $\mathcal{P}(\mathbf{N}^+)$,

$$[K \Rightarrow \textbf{rec}\ x.t]_{CT} = [K \Rightarrow t[\textbf{rec}\ x.t/x]]_{CT}. \qquad \diamond$$

PROPOSITION 5.2 *For each* $t \in CREC(\Sigma, \chi)$ *and for each finite set* $K \in \mathcal{P}(\mathbf{N}^+)$,

$$[K \Rightarrow \mu.t]_{CT} = \langle \mu, K \rangle \cdot [\delta(K) \cup \{1\} \Rightarrow t]_{CT}. \qquad \diamond$$

PROPOSITION 5.3 *For each* $t, t', t'' \in CREC(\Sigma, \chi)$ *and for each finite set* $K \in \mathcal{P}(\mathbf{N}^+)$,

i. $[K \Rightarrow t' + t'']_{CT} = [K \Rightarrow t']_{CT} +_{CT} [K \Rightarrow t'']_{CT}$;
ii. $[K \Rightarrow (t[\beta/\alpha])]_{CT} = [(K \Rightarrow t)[\beta/\alpha]]_{CT}$;
iii. $[K \Rightarrow (t\backslash\alpha)]_{CT} = [(K \Rightarrow t)\backslash\alpha]_{CT}$;
iv. $[K \Rightarrow t'\|t'']_{CT} = [K \Rightarrow t']_{CT}\|_{CT}[K \Rightarrow t'']_{CT}$. $\diamond$

PROPOSITION 5.4 *For each C³S term* e, $[1]\ [e]_{CT} = [\delta(e)]_{CT}$. $\diamond$

PROPOSITION 5.5 *For each C³S term* e *and for each finite set* $H \in \mathcal{P}(\mathbf{N}^+)$ *and* $1 \in H$,

$$\lceil H/1 \rfloor [e]_{CT} = [\eta(H, e)]_{CT}. \qquad \diamond$$

6. CAUSAL WEAK BISIMULATION

We start by defining the causal weak bisimulation between C³S terms (and thus, by way of extension, on CCS terms) and then we will define it on causal trees to indicate a *sound* and *complete* system of equational axioms for causal CCS.

To correctly define the causal weak bisimulation relation we have the problem of the "cleaning" of the sets of causes of the $\langle \lambda, K \rangle$ transitions from the pointers to the τ actions which precede them. To this aim we define the new operator "flexa," $\dagger$, and an extension of the transition relation such that when a $\langle\tau, K\rangle$ transition occurs the sets of causes of the following transitions are not modified.

We require for the operator $\dagger$: $\mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+)$ to distribute over all operations in $\Sigma \backslash (\{\mathbf{nil}\} \cup \{\mu. | \mu \in \mathcal{A}\})$, and we have:

$$\dagger\,(K \Rightarrow t) \equiv (\dagger(K) \Rightarrow t),$$
$$\dagger\,K = \begin{cases} \emptyset & \text{if } K = \{1\}, \\ \{k-1 | k \in K,\, k > 1\} & \text{otherwise.} \end{cases}$$

Let $\mathcal{L}^{DD} = \{\langle\lambda, K\rangle | \lambda \in \Delta \cup \bar{\Delta} \wedge K \subset \mathbf{N}^+, K \text{ finite}\} \cup \{\tau\}$ be the set of labels where only the non-τ symbols bear indication of their observable causes (see [5]).

The transition relation $\stackrel{\omega}{\longmapsto}$ where $\omega \in \mathcal{L}^{DD}$, is defined from $\xrightarrow{\mu, K}$ by the following two rules:

$$\frac{e \xrightarrow{\lambda, K} e'}{e \stackrel{\lambda, K}{\longmapsto} e'} \qquad \frac{e \xrightarrow{\tau, K} e'}{e \stackrel{\tau}{\longmapsto} \dagger(e')}$$

In order to cope with this new setting we need some definitions, which are standard (see [10]).

Definition 6.1 (Sequences of internal actions) Given e and e' in $\mathrm{C^3S}$, we write $e \Rightarrow e'$ if there exists a sequences of n τ-transitions, $n \geq 0$ such that e becomes e' without taking care of the causes of τs, namely there exists e_i, $1 \leq i \leq n$, such that

$$e \equiv e_1 \stackrel{\tau}{\longmapsto} e_2 \stackrel{\tau}{\longmapsto} e_3 \cdots \stackrel{\tau}{\longmapsto} e_n \equiv e'. \qquad \diamond$$

We will write $e \stackrel{\omega}{\Longrightarrow} e'$, $\omega \in \mathcal{L}^{DD}$, if there exist e_1 and e_2 such that

$$\text{either } e \Rightarrow e_1 \stackrel{\lambda, K}{\longmapsto} e_2 \Rightarrow e' \text{ or } e \Rightarrow e_1 \stackrel{\tau}{\longmapsto} \dagger(e_2) \Rightarrow e'.$$

$e \stackrel{\tau}{\Longrightarrow} e'$ will be the same of $e \Rightarrow e'$ and, if $n = 0$, $e \equiv e'$.

Definition 6.2 (Causal weak bisimulation) Given e and f in $\mathrm{C^3S}$, we have $e \sim_W f$ if for each $\omega \in \mathcal{L}^{DD}$,

i. whenever $e \stackrel{\omega}{\Longrightarrow} e'$ then there exists f' such that $f \stackrel{\omega}{\Longrightarrow} f'$ and $e' \sim_W f'$.
ii. whenever $f \stackrel{\omega}{\Longrightarrow} f'$ then there exists e' such that $e \stackrel{\omega}{\Longrightarrow} e'$ and $e' \sim_W f'$.

$\diamond$

Two CCS terms will be causally weakly equivalent if the corresponding C^3S terms are such.

Causal weak bisimulation satisfies all the properties that hold for the interleaving case, *e.g.*, propositions from Sections 5.2 and 5.3 in [10] are still valid.

DEFINITION 6.3 Given e and f in C^3S, let $e \sim_C f$ if for each $\omega \in \mathcal{L}^{DD}$,

i. whenever $e \overset{\omega}{\longmapsto} e'$ then there exists f' such that $f \overset{\omega}{\longrightarrow} f'$ and $e' \sim_W f'$,
ii. whenever $f \overset{\omega}{\longmapsto} f'$ then there exists e' such that $e \overset{\omega}{\longrightarrow} e'$ and $e' \sim_W f'$. ◇

As done before, if t_1 and t_2 are CCS terms then $t_1 \sim_C t_2$ iff $(\emptyset \Rightarrow t_1 \sim_C (\emptyset \Rightarrow t_2)$.

Following the corresponding proof in [10], it is immediate to prove that $\sim_C$ is a congruence with respect to CCS operators, and that it is completely characterized by (the transposition to terms of) axioms (Al) $\div$ (A4) listed in Section 2 and the τ-laws:

$$\text{(A5)} \quad \mu.\tau.x = \mu.x;$$
$$\text{(A6)} \; x + \tau.x = \tau.x;$$
$$\text{(A7)} \; \mu.(x + \tau.y) + \mu.y = \mu.(x + \tau.y),$$

where x, y and z are CCS terms and $\mu \in \mathcal{A}$.

Besides, it is easy to establish the following:

FACT 6.1 *The causal observational congruence, $\sim_C$, is finer than $\sim$ (Milner's congruence).* ◇

By way of conclusion, we mention that the correspondence between the transitions of C^3S terms $e \overset{z}{\longrightarrow} e'$ and the tree-transitions $[e]_{CT} \overset{z}{\longrightarrow} [e']_{CT}$ due to the full adequacy of the model $\mathcal{CT}$, permits to define the causal weak bisimulation relation on causal trees following the pattern of Definition 6.2. Similarly to what we did with terms, we need a new tree-transition relation which permits to correctly upgrade the sets of causes of those arcs who are preceded by τ-arcs. A family of auxiliary operators is introduced which removes from the sets of causes of all arcs of a sub-tree prefixed by a τ-arc the pointer to that arc. Thus, we have a relation which is an extension of the tree-transition relation defined in Section 5. Formal definitions can be found in Appendix C.

The new definition of causal congruence introduced here, allows us to obtain a complete system of equational axioms for causal observational congruence over finite, *i.e.*, non recursive, CCS programs t identified with the corresponding C^3S expressions $\emptyset \Rightarrow t$. The first seven axioms are copies of Milner's axioms (A1) $\div$ (A7) indicated above, where label μ ranges over $\mathcal{L}$. The remaining axioms are *Init*, *Nil*, *Pref*, *Sum*, *Relab*, *Restr*, *Interleave* and the defining equations for combinators $\lhd \cdot \rhd$, $[n]$ and $\lceil K/n \rfloor$ operating on causal trees specified by sum expressions. The correctness of the resultant axiomatization derives from the fact that axioms (A1) $\div$ (A4) are clearly valid and that the τ-laws may be proved valid for causal trees, letting μ range over $\mathcal{L}^{DD}$. Completeness directly emerges from a remake of the original proof of completeness for axioms (A1) $\div$ (A7) in [9, 10], provided that axioms *Init*, *Nil*, *Pref*, *Sum*, *Relab*, *Restr* and *Interleave* are used to derive from any non recursive C^3S expression an equivalent expression on combinators **NIL**, $\langle \mu, K \rangle \cdot$ and $+$ (a "normal form" for causal trees).

References

[1] Basagni, S. (1995). *Causal Trees: An Application to CCS*, Tech. Rep. RI-DSI-146-9, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano.

[2] Boudol, G. (1990). Flow Event Structure and Flow Net, In: *Semantics of Systems of Concurrent Processes* (Guessarian, I., Ed.), Springer-Verlag, *L.N.C.S.*, **469**, 62–95.

[3] Boudol, G. and Castellani, I., Permutation of Transition: an Event Structure Semantics for CCS and SCCS, in [12], pp. 411–427.

[4] Boudol, G. and Castellani, I. (1990). Three Equivalent Semantics for CCS, In: *Semantics of Systems of Concurrent Processes* (Guessarian, I., Ed.), Springer-Verlag, *L.N.C.S.*, **469**, 96–141.

[5] Darondeau, Ph. and Degano, P. (1989). Casual Trees, *Proc. 16th ICALP* (Ausiello, G., Dezani-Ciancaglini, M. and Ronchi Della Rocca, S., Eds.), Stresa, 11–15 July, 1989, Springer-Verlag, *L.N.C.S.*, **372**, 234–248.

[6] Darondeau, Ph. and Degano, P. (1990). Casual Trees Interleaving+Causality, In: *Semantics of Systems of Concurrent Processes* (Guessarian, I., Ed.), Springer-Verlag, *L.N.C.S.*, **469**, 239–255.

[7] Degano, P. and Priami, C. (1992). Proved Trees, *Proc. 19th ICALP* (Kuich, W., Ed.), Wien, July 13–17, 1992, Spriger-Verlag, *L.N.C.S.*, **623**, 629–640.

[8] Milner, R. (1980). *A Calculus of Communicating Systems*, Springer-Verlag, *L.N.C.S.*, **92**.

[9] Milner, R. (1984). Notes on a Calculus for Communicating Systems, In: *Control Flow and Data Flow: Concepts of Distributed Programming* (Broy, M., Ed.), *NATO ASI Series F*, **14**, Springer-Verlag, Heidelberg, pp. 205–228.

[10] Milner, R. (1989). *Communication and Concurrency*, Prentice Hall International.

[11] Plotkin, G. (1981). A Structural Approach to Operational Semantics, DAIMI Report FN-19, Department of Computer Science, Aarhus.

[12] Proc REX School/Workshop on *Linear Time, Branching Time and Partial Ordering in Logics and Models for Concurrency* (De Bakker, J. W., De Roever, J.-P. and Rozenberg, G., Eds.), 1988, Springer-Verlag, *L.N.C.S.*, **354** (1989).

A. SOME ALGEBRAIC LAWS

Here we state some algebraic laws satisfied by the operators introduced in Section 4.1. These laws will be useful in the proofs of the propositions of Section 5.

Before we need the following definition:

DEFINITION A.1 (Boundedness) A finite set $K \in \mathcal{P}(\mathbf{N}^+)$ is i-bounded if for each $k \in K$ is $k < i$. Then a causal tree is bounded if for each initial path labeled with

$$\langle \mu_1, K_1 \rangle \cdot \langle \mu_2, K_2 \rangle \cdot \cdots \cdot \langle \mu_n, K_n \rangle$$

every set K_i, is i-bounded. $\diamond$

Notice that if $i = 1$, being $k < 1$ and $k \in \mathbf{N}^+$, then it will be $K_1 = \emptyset$. The previous definition is aimed at ensuring that all the causes of an arc are in the tree – this constraint avoids *dangling references*, *i.e.*, references to arcs not in the tree. The definition also introduces a "normalization" constraint on causal trees in order to have a meaningful notion of equivalence. Each law is shaped as an indexed family of equations over causal trees.

LEMMA A.1 *Let T be a causal tree and let K, H in $\mathcal{P}(\mathbf{N}^+)$ be finite. Then*

$$\triangleleft K \triangleright_n \ \triangleleft H \triangleright_n T = \triangleleft K \cup H \triangleright_n T. \qquad \diamond$$

Let, in the following five lemmas, T and U be bounded trees in $\mathcal{CT}$, and let K and H be finite in $\mathcal{P}(\mathbf{N}^+)$.

LEMMA A.2 *Let $n, j \in \mathbf{N}^+$, $n \geq j$. Then* $[j] \triangleleft K \triangleright_{n-1} T = \triangleleft K \triangleright_n T$. $\diamond$

LEMMA A.3 *Let $n \in \mathbf{N}^+$. Then* $\triangleleft K \triangleright_n (T \|_{\mathcal{CT}} U) = \triangleleft K \triangleright_n T \|_{\mathcal{CT}} \triangleleft K \triangleright_n U$. $\diamond$

LEMMA A.4 *Let $n \in \mathbf{N}^+$. Then*

$$[n](T \|_{\mathcal{CT}} U) = [n]T \|_{\mathcal{CT}} [n]U. \qquad \diamond$$

LEMMA A.5 *Let $n \in \mathbf{N}^+$. Then*

$$\lceil H/n \rfloor \triangleleft K \triangleright_0 T = \begin{cases} \triangleleft K \triangleright_0 T & \text{if } n \notin K, \\ \triangleleft H \triangleright_0 (\triangleleft K \triangleright_0 T) & \text{if } n \in K. \end{cases} \qquad \diamond$$

LEMMA A.6 *Let $n \in \mathbf{N}^+$. Then*

$$\lceil \triangleleft K \triangleright_n /n \rfloor (T \|_{\mathcal{CT}} U) = \lceil \triangleleft K \triangleright_n /n \rfloor T \|_{\mathcal{CT}} \lceil \triangleleft K \triangleright_n /n \rfloor U. \qquad \diamond$$

($\triangleleft K \triangleright_n$ abbreviates $\triangleleft K \triangleright_n \emptyset$, *i.e.*, the application of δ to the set K n times.)

The proofs of the above lemmas are easily obtained by induction on the depth of a (bounded) causal tree. We do not include them here, because they are tedious and long. They can be found in [1]-Appendix A.

B. FULL ADEQUACY: THE PROOF

We now establish Propositions 5.1 – 5.5. We need the laws of Appendix A, the axioms of Section 4.2 and the following lemmas:

LEMMA B.1 *If T and U are bounded causal trees then $(T \|_{CT} U)$ is a bounded causal tree.*

Proof Follows directly from the *Interleave* axiom for $\mathcal{CT}$. ◇

LEMMA B.2 *For each t in CCS, $[t]_{CT}$ is bounded and equals $[\emptyset \Rightarrow t]_{CT}$.*

Proof $[t]_{CT}$ is a bounded causal tree. Indeed, by contradiction, let t have minimal complexity in the set of terms for which $[t]_{CT}$ is not bounded. From the algebraic axioms of $\mathcal{CT}$, t must have either the form $\mu.t'$ or the form $t' \| t''$. If $t = \mu.t'$ and $[t']_{CT}$ is bounded then $[t]_{CT}$ is bounded (axiom *Pref*). If $t = t' \| t''$ and both $[t']_{CT}$ and $[t'']_{CT}$ are bounded then $[t]_{CT}$ is bounded (Lemma B. 1). Hence t cannot be minimal and we have the desired contradiction.

For the second assertion we have $[\emptyset \Rightarrow t]_{CT} = \emptyset \Rightarrow_{CT} [t]_{CT} = \lhd \emptyset \rhd_0 [t]_{CT} = [t]_{CT}$ applying axiom *Init*. ◇

PROPOSITION 5.1 *For each $(\textbf{rec}\ x.t) \in CREC(\Sigma, \chi)$ and each finite set $K \in \mathcal{P}(\mathbf{N}^+)$,*

$$[k \Rightarrow \textbf{\textit{rec}}\, x.t]_{CT} = [K \Rightarrow t[\textbf{\textit{rec}}\, x.t/x]]_{CT}.$$

Proof Since **rec** is interpreted in $\langle \mathcal{CT}, \Sigma_{CT} \rangle$ as a fixed point combinator we have $[K \Rightarrow \textbf{rec}\, x.t]_{CT} = \lhd K \rhd_0 [\textbf{rec}\, x.t]_{CT} = \lhd K \rhd_0 [t[\textbf{rec}\, x.t/x]]_{CT} = [K \Rightarrow t[\textbf{rec}\, x.t/x]]_{CT}$ by applying axiom *Init*. ◇

PROPOSITION 5.2 *For each $t \in CREC(\Sigma, \chi)$ and each finite set $K \in \mathcal{P}(\mathbf{N}^+)$,*

$$[K \Rightarrow \mu.t]_{CT} = \langle \mu, K \rangle \cdot [\delta(K) \cup \{1\} \Rightarrow t]_{CT}.$$

Proof By axiom *Init* we have:

$$
\begin{array}{ll}
[K \Rightarrow \mu.t]_{CT} = & \\
\lhd K \rhd_0 [\mu.t]_{CT} = & \text{by axiom } \mathit{Pref} \\
\lhd K \rhd_0 \langle \mu, \emptyset \rangle \cdot \lhd\{1\}\rhd_0 [t]_{CT} = & \text{by definition of } \lhd K \rhd_n \\
\langle \mu, K \rangle \cdot \lhd(K) \rhd_1 \lhd\{1\}\rhd_0 [t]_{CT} = & \text{by definition of } \lhd K \rhd_n \\
\langle \mu, K \rangle \cdot \lhd\delta(K) \rhd_0 \lhd\{1\}\rhd_0 [t]_{CT} = & \text{by Lemma A.1} \\
\langle \mu, K \rangle \cdot \lhd\delta(K) \cup \{1\}\rhd_0 [t]_{CT} = & \text{by axiom } \mathit{Init} \\
\langle \mu, K \rangle \cdot [\delta(K) \cup \{1\} \Rightarrow t]_{CT}. &
\end{array}
$$
◇

PROPOSITION 5.3 *For each* t, t', $t'' \in CREC(\Sigma, \chi)$ *and each finite set* $K \in \mathcal{P}(\mathbf{N}^+)$,

i. $[K \Rightarrow t' + t'']_{CT} = [K \Rightarrow t']_{CT} +_{CT} [K \Rightarrow t'']_{CT}$;
ii. $[K \Rightarrow (t[\beta/\alpha])]_{CT} = [(K \Rightarrow t)[\beta/\alpha]]_{CT}$;
iii. $[K \Rightarrow (t \backslash \alpha)]_{CT} = [(K \Rightarrow t) \backslash \alpha]_{CT}$;
iv. $[K \Rightarrow t' \| t'']_{CT} = [(K \Rightarrow t']_{CT} \|_{CT} [K \Rightarrow t'']_{CT}$;

Proof We shall prove only the last assertion (the first three being trivial consequences of the definition of $\lhd K \rhd_n$ and of its distributivity over $+$, $[\alpha/\beta]$ and $\backslash\alpha$).

iv. Applying the *Init* axiom we have:

$$
\begin{array}{ll}
[K \Rightarrow t' \| t'']_{CT} = & \\
\lhd K \rhd_0 [t' \| t'']_{CT} = & \\
\lhd K \rhd_0 \lhd [t']_{CT} \|_{CT} [t'']_{CT} = & \text{by Lemma B.2 and Lemma A.3} \\
\lhd K \rhd_0 [t']_{CT} \|_{CT} \lhd K \rhd_0 [t'']_{CT} = & \text{by axiom } \mathit{Init} \\
[K \Rightarrow t']_{CT} \|_{CT} [K \Rightarrow t'']_{CT} &
\end{array}
$$
◇

PROPOSITION 5.4 *For each* C^3S *term* e, $[1]\ [e]_{CT} = [\delta(e)]_{CT}$.

Proof If e is an expression of the form $K \Rightarrow t$ then what we want to prove is $[1]\ [K \Rightarrow t]_{CT} = [\delta(K) \Rightarrow t]_{CT}$.

Applying the *Init* axiom we have:

$$
\begin{array}{ll}
[1][K \Rightarrow t]_{CT} = & \\
[1] \lhd K \rhd_0 [t]_{CT} = & \text{by Lemma B.2 and Lemma A.2} \\
\lhd K \rhd_1 [t]_{CT} = & \text{by definition of } \lhd K \rhd_n \\
\lhd \delta(K) \rhd_0 [t]_{CT} = & \text{by the } \mathit{Init} \text{ axiom} \\
[\delta(K) \Rightarrow t]_{CT}. &
\end{array}
$$
◇

There remains to check the case when e is an expression of the form $e' \| e''$. We want that $[1][e]_{CT} = [1][e' \| e'']_{CT} = [\delta(e' \| e'')]_{CT} = [\delta(e)]_{CT}$. We proceed by induction on (sub)expressions. By definition:

$$\begin{array}{ll}
[1][e]_{CT} = & \\
[1]([e']_{CT} \|_{CT} [e'']_{CT}) = & \text{by Lemma A.4} \\
[1][e']_{CT} \|_{CT} [1][e'']_{CT} = & \text{by inductive hypothesis} \\
[\delta(e')]_{CT} \|_{CT} [\delta(e'')]_{CT} = & \text{by definition} \\
[\delta(e') \| \delta(e'')]_{CT} = & \text{because of the distributivity of } \delta \text{ over} \| \\
[\delta(e' \| e'']_{CT} = & \\
[\delta(e)]_{CT}. & \diamond
\end{array}$$

Proposition 5.5 *For each C^3S term e and for each finite set $H \in \mathcal{P}(\mathbf{N}^+)$ and $1 \in H$,*

$$\lceil H/1 \rfloor [e]_{CT} = [\eta(H, e)]_{CT}.$$

Proof This proof is similar to that of Proposition 5.4, and relies on Lemmas B.2, A.5 and A.6. $\diamond$

C. CAUSAL WEAK BISIMULATION ON CAUSAL TREES: FORMAL DEFINITIONS

Here we extend the tree-transition relation introduced in Section 5 in order to define causal weak bisimulation on causal trees. Starting from $\xrightarrow{z}$, $z \in \mathcal{L}$, we deduce the relation $\overset{\omega}{\longmapsto}$, $\omega \in \mathcal{L}^{DD}$, by the following two rules:

$$\frac{T \xrightarrow{\lambda, K} T'}{T \overset{\lambda K}{\longmapsto} T'} \qquad \frac{T \xrightarrow{\tau, K} T'}{T \overset{\tau}{\longmapsto} \triangleright T' \triangleleft^1}$$

where the family of operators $\triangleleft . \triangleright^n : \mathcal{P}(\mathbf{N}^+) \to \mathcal{P}(\mathbf{N}^+)$ is defined in the following way:

$$\triangleright \mathbf{NIL} \triangleleft^n \equiv \mathbf{NIL},$$
$$\triangleright T \triangleleft^n \equiv \Sigma_{i \in I} \langle \mu_i, \triangleright K_i \triangleleft^n \rangle \cdot \triangleright T_i \triangleleft^{n+1},$$

where

$$\triangleright K \triangleleft^n = \{k - 1 | k \in K, k > n\} \cup \{k | \in K, k < n\}.$$

($\triangleright K \triangleleft$ removes from the set K the integer n, if $n \in K$, and decreases the pointers to the arcs that precede the current τ arc, in order to skip it.)

Notation Given two causal trees, T and T', we write $T \Rightarrow T'$ to mean that T' is a sub-tree of T and that there exists a path in T made of n τ-arcs, $n \geq 0$, such that from the root of T we can reach the root of T' without taking care of the causes of the τ's, *i.e.*, there exist T_i, $1 \leq i \leq n$, such that:

$$T \equiv T_1 \stackrel{\tau}{\longmapsto} T_2 \stackrel{\tau}{\longmapsto} T_3 \cdots \stackrel{\tau}{\longmapsto} T_n \equiv T'.$$

By writing $T \stackrel{\omega}{\Longrightarrow} T'$, $\omega \in \mathcal{L}^{DD}$, we mean that there exists T_1 and T_2 such that $T \Rightarrow T_1 \stackrel{\omega}{\longmapsto} T_2 \Rightarrow T'$. $T \stackrel{\tau}{\longrightarrow} T'$ is equal to $T \Rightarrow T'$ and, if $n = 0$, $T \equiv T'$. $\diamond$

Definition C.1 (Causal weak bisimulation on Causal Trees) Given T_1 and T_2 in $\mathcal{CT}$, we say $T_1 \sim_W T_2$ if for each $\omega \in \mathcal{L}^{DD}$,

i. whenever $T_1 \stackrel{\omega}{\Longrightarrow} T'_1$ then there exists T'_2 such that $T_2 \stackrel{\omega}{\Longrightarrow} T'_2$ and $T'_1 \sim_W T'_2$,
ii. whenever $T_2 \stackrel{\omega}{\Longrightarrow} T'_2$ then there exists T'_1 such that $T_1 \stackrel{\omega}{\Longrightarrow} T'_1$ and $T'_1 \sim_W T'_2$, $\diamond$

Definition C.2 (Causal Observational Congruence) Given T_1 and T_2 in $\mathcal{CT}$, we say $T_1 \sim_C T_2$ if for each $\omega \in \mathcal{L}^{DD}$,

i. whenever $T_1 \stackrel{\omega}{\longmapsto} T'_1$ then there exists T'_2 such that $T_2 \stackrel{\omega}{\Longrightarrow} T'_2$ and $T'_1 \sim_W T'_2$,
ii. whenever $T_2 \stackrel{\omega}{\longmapsto} T'_2$ then there exists T'_1 such that $T_1 \stackrel{\omega}{\Longrightarrow} T'_1$ and $T'_1 \sim_W T'_2$. $\diamond$