Consiglio Nazionale delle Ricerche

ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE

PISA

A NEW OPERATIONAL SEMANTICS FOR CCS BASED ON CONDITION/EVENT SYSTEMS

P. Degano, R. De Nicola, U. Montanari

Nota interna B4-42
Agosto 1986

A New Operational Semantics for CCS Based on Condition/Event Systems

*Pierpaolo Degano *, Rocco De Nicola+ and Ugo Montanari **

* Dipartimento di Informatica, Università di Pisa, PISA
+ Istituto di Elaborazione dell'Informazione, C.N.R., PISA

Abstract. A new class of Petri Nets, called Augmented Condition/Event Systems is defined, by slightly relaxing the condition for enabling events. A Net from this class called Σ_{CCS} is used to give a new operational semantics to Milner's Calculus for Communicating Systems. The set of CCS agents together with the traditional, interleaving based, derivation relation is proved isomorphic to the case graph of Σ_{CCS} (when single transitions only are considered). Our achievement is twofold: first we provide CCS with a semantics which is able to describe concurrency and causal dependencies between the actions the various agents can perform; second, we guarantee an adequate linguistic level for the particular class of Petri Nets which can be defined through CCS operators.

Running Head: CCS as a Condition/Event System

1. Introduction

Petri Nets [5, 9] are one of the best known models of concurrent and communicating systems, and have a suggestive graphical representation. They can be viewed as a generalization of transition systems in which concurrency and causal independence between transitions is explicitly represented. A generally acknowledged inadequacy is their lack of compositionality and modularity, due to the absence of a linguistic level which is necessary to build bigger nets from smaller ones. Many different kinds of Petri Nets have been defined, and they can be divided into two families. The first class contains those more intensional which represent models of real systems, and may thus have cycles. The nets in the second class can be seen as behaviours of systems, with a partial ordering structure obtained by unfolding nets in the first class. Privileged representatives of the first class are Condition/Event (C/E) Systems which describe how the occurrence of *events* causes changes in local states, called *conditions*. Typical elements of the second class are Occurrence Nets. A brief summary of the relevant aspects of C/E Systems is given in Section 2.

We intend to study the relationships between C/E Systems and Milner's Calculus for Communicating Systems (CCS) [7]. This is a language for concurrent and communicating systems based on a small number of operators. CCS operational semantics is a transition system defined via a set of transition rules in a syntax-driven way, by using the SOS technique of [8]. As it stands, CCS has a so-called *interleaving* semantics where the "|" operator for parallel composition of processes is not primitive: given any finite process containing |, another process always exists without | which exhibits the same behaviour. A brief summary of CCS is given in Section 2.

In this paper, we present a new operational semantics for CCS which considers the parallel operator as "first class" and associates a Condition/Event System to CCS. Our achievement is twofold: first we provide CCS with a semantics which is able to describe concurrency and causal dependencies between the actions the various agent can perform; second, we guarantee an adequate linguistic level for the particular class of Petri Nets which can be defined through CCS operators. Our results should provide a framework for evaluating CCS's expressive power, and for understanding the relationships between the two theories, so that analytic concepts and techniques might be transferred form one theory to the other.

There have been many attempts to use Petri Nets to give an operational semantics to CCS without value passing ("pure" CCS). To the best of our knowledge, no definitive result has yet been achieved: either only subsets of CCS have been considered or the interleaving semantics of the resulting net is not the standard CCS one [1, 6, 10, 11].

Goltz and Mycroft [6] give a denotational semantics of CCS in terms of Occurrence Nets and an operational semantics in terms of Place/Transitions Nets. In the former case, since Occurrence Nets are used, the operational intuition of intensional nets and some of the advantages of their graphical representation are lost. In the latter, only a CCS subcalculus which does not contain the restriction operator is considered. Moreover the semantics they give is not in complete agreement with the interleaving semantics; in particular, it does not respect all the causal dependencies between the actions performed by the agents (further details are given in the conclusions).

Winskel [10] also proposes a partial ordering denotational semantics for CCS which is based on event structures, a domain very close to Occurrence Nets and thus extensional as well. However, he simply claims that the interleaved semantics agrees with Milner's synchronization trees semantics but does not give any formal statement of this fact. In [11] a categorical interpretation of Place/Transition Nets is proposed and various operators are defined on them to provide an adequate linguistic level. Unfortunately, the sum operator defined there is not in full agreement neither with that defined in [7] nor with operational intuitions about nondeterministic choice; this means that his category of nets cannot be used as a semantic domain for CCS.

De Cindio et al. [1] map CCS into a subclass of Petri Nets (Superposed Automata Nets) the elements of which are systems composed of interacting sequential automata. However, they restrict CCS syntax to forbid the generation of unboundedly many instances of a recursive system.

Here, for the first time, we map the whole "pure" CCS into a minor extension of C/E Systems which we call Augmented Condition/Events (A-C/E) Systems. Our extension relaxes simplicity and permits self-loops. Indeed, we have taken care of maintaining all the relevant properties of C/E Systems, *in primis* the capability of clearly expressing concurrency and of distinguishing it from nondeterminism. The epistemological consequences of this different notion of event have not been studied here. Our semantics carries more information than Milner's, namely that about concurrency, since the latter can be recovered by considering the interleaving part of the case graph of the former.

Simplicity (i.e. an event is characterized by its pre- and post conditions) has been relaxed since situations of pure nondeterminism can be expressed in CCS, e.g. by the CCS agent $\alpha E + \beta E$, which may evolve to E in two ways, firing either an event observed as α or as β. Afterwards, we show that simplicity can be kept, although a more complicated and less suggestive net is obtained.

The other extension is actually needed to enable a transition for which some preconditions coincide with some postconditions. This is because self-loops arise naturally when dealing with recursive CCS agents, e.g. rec x. αx. The conditions for enabling events and the property of contact-freeness have been accordingly modified, still maintaining the properties that an event whose preconditions hold is always enabled and that *no token is ever lost.* Thus, the formal troubles due to contact situations are avoided, and we argue that all the results proved for contact-free C/E Systems hold for contact-free A-C/E Systems too. An extension similar to our is proposed in [11] although no justification is given for it there.

In order to define the new CCS operational semantics, the SOS approach is taken once again, but a different notion of derivation relation is defined, which we call *partial ordering derivation.* It relates sequential processes of agents, rather than their whole global states. Sequential processes are obtained by decomposing CCS agents, and the partial ordering derivation relation describes both the actions sequential processes may perform and their effects. The A-C/E system, called Σ_{CCS}, is obtained straightforwardly from the partial ordering derivation. More precisely, sequential processes are conditions; decompositions of CCS agents are cases; and elements of the partial ordering derivation relation are events.

The soundness of the new semantics (with respect to the interleaving operational semantics of [7]) relies on the fact that the transition graph of CCS (in which nodes are agents and arcs are derivations) is isomorphic to the *interleaving* case graph of Σ_{CCS} (the case graph with the arcs labelled by a single event only).

2. Background

Here, we shortly introduce the relevant definitions about Petri Condition/Event (C/E) Systems [5, 9] and Milner's Calculus for Communicating Systems [7].

A net is a triple $<B, E; F>$, where

- B is the set of conditions;
- E is the set of events;
- $B \cap E = \Phi$;
- $F \subseteq (B \times E) \cup (E \times B)$.

Given a net $N = <B, E; F>$, let $x, y \in B \cup E$,

- $\cdot x$ denotes $\{y \mid yFx\}$, called **preset**, and $x\cdot$ denotes $\{y \mid xFy\}$, called **postset**;
- x is **isolated** if $\cdot x \cup x\cdot = \Phi$;
- N is **simple** if, whenever $\cdot x = \cdot y$ and $x\cdot = y\cdot$, $x = y$;
- a subset $c \subseteq B$ is called **case**;
- $e \in E$ is enabled by c iff c is a case and $\cdot e \subseteq c$ and $e\cdot \subseteq B - c$ (- denotes set difference).

Furthermore, given $c_1, c_2 \subseteq B$ and $G \subseteq E$, the **step** $c_1\ [G>\ c_2$, is defined if

- $\forall e \in G$, e is c_1-enabled;
- $\forall e_1, e_2 \in G$, $e_1 \neq e_2$, $\cdot e_1 \cap \cdot e_2 = e_1\cdot \cap e_2\cdot = \Phi$;
- $c_2 = (c_1 - \cdot G) \cup G\cdot$ (where $\cdot$ is extended on sets).

A Condition/Event system (C/E system) is a quadruple $\Sigma = <B, E; F, C>$, where

- $<B, E; F>$ is a simple net with no isolated element and $B \cup E \neq \Phi$;
- $C \subseteq 2^B$ is an equivalence class of the reachability relation $R = (r \cup r^{-1})^*$, being $r \subseteq 2^B \times 2^B$, where $c_1\ r\ c_2$ iff $\exists G \subseteq E$ such that $c_1\ [G>\ c_2$. C is called the case class;
- $\forall e \in E$, $\exists c$ such that e is c-enabled.

A contact-free C/E system is a C/E system $<B, E; F, C>$ in which $\forall e \in E$, $\forall c \in C$

- $\cdot e \subseteq c$ implies $e\cdot \subseteq B - c$;
- $e\cdot \subseteq c$ implies $\cdot e \subseteq B - c$.

Given a C/E system $\Sigma = <B, E; F, C>$, let $P = \{(c_1, G, c_2) \mid c_1 [G> c_2$ is a step of $\Sigma\}$. Then, the graph $\Phi = (C, P)$ is called the case graph of Σ.

We present now a brief survey of CCS's syntax and semantics. First, we shall recall the main operators of the calculus, then we will present the traditional interleaving semantics of [7].

If $\Delta = \{\alpha, \beta, \gamma \dots\}$ is a fixed set, $\Delta^- = \{\alpha^- \mid \alpha \in \Delta\}$, $\Lambda = \Delta \cup \Delta^-$ (ranged over by λ) denotes the set of *visible actions*; τ is a distinguished *invisible action* not in Λ and $\Lambda \cup \{\tau\}$ is ranged over by μ, then the CCS agents are closed terms (i.e. terms without free variables) generated by the following BNF-like grammar:

$E ::= x \mid NIL \mid \mu E \mid E\backslash\alpha \mid E[\phi] \mid E + E \mid E|E \mid rec\ x.\ E,$

where x is a variable and ϕ is a permutation of $\Lambda \cup \{\tau\}$ which preserves τ and the operation $^-$ of complementation.

CCS interleaving operational semantics is based on a labelled transition system, the transition relation of which is defined by a set of inference rules over agents. A relation $—\mu\rightarrow$, called *derivation relation*, is defined, with the intuition that agent E_1 may evolve to become agent E_2 either by reacting to a λ-stimulus from its environment ($E_1 —\lambda\rightarrow E_2$) or by performing an internal action which is independent of the environment ($E_1 —\tau\rightarrow E_2$).

Milner's derivation relation $E_1 —\mu\rightarrow E_2$ is defined as the least relation satisfying the following axiom and inference rules:

Act) $\mu E —\mu\rightarrow E$

Res) $E_1 —\mu\rightarrow E_2$ implies $E_1\backslash\alpha —\mu\rightarrow E_2\backslash\alpha$, $\mu \notin \{\alpha, \alpha^-\}$

Rel) $E_1 —\mu\rightarrow E_2$ implies $E_1[\phi] —\phi(\mu)\rightarrow E_2[\phi]$

Sum) $E_1 —\mu\rightarrow E_2$ implies $E_1 + E —\mu\rightarrow E_2$ and $E + E_1 —\mu\rightarrow E_2$

Com) $E_1 —\mu\rightarrow E_2$ implies $E_1|E —\mu\rightarrow E_2|E$ and $E|E_1 —\mu\rightarrow E|E_2$

$E_1 —\lambda\rightarrow E_2$ and $E'_1 —\lambda^-\rightarrow E'_2$ implies $E_1|E'_1 —\tau\rightarrow E_2|E'_2$

Rec) $E_1[rec\ x.\ E_1/x] —\mu\rightarrow E_2$ implies $rec\ x.\ E_1 —\mu\rightarrow E_2$.

This relation completely specifies CCS operational semantics which, given an agent, determines the actions or the sequences of actions it may perform, and the new agent which is obtained as a result. We aim at defining a Petri Net which, besides the above informations, carries also informations about the causal dependencies and the possible concurrency among the actions performed by every CCS agent.

3. Augmented Condition/Event Systems

In this section we introduce the class of Petri Nets which we will use in the rest of the paper to describe the semantics of CCS agents. We define the class of Augmented Condition/Event (A-C/E) Systems by slightly relaxing the condition for enabling events to deal with self-loops and by removing the simplicity condition to obtain simpler nets.

We drop the condition of simplicity on A-C/E systems since we want to straightforwardly represent situations of pure nondeterminism, such as the one expressed in CCS by the agent $\alpha E + \beta E$, which may evolve to E in two ways, firing an event either observed as α or as β. Afterwards, we will show that simplicity can be kept, but at the price of obtaining a more complicated and less suggestive net.

The actual extension to C/E Systems consists of relaxing the conditions under which a transition may occur. This is mainly due to the fact that we want to consider as enabled an event for which the preset is not disjoint from the postset. Indeed, loops arise when considering CCS agents involving recursion, e.g. rec x. αx. The property of contact-freeness has been accordingly modified, still avoiding those cases where the preset of an event holds, but the event is not enabled, and avoiding the loss of tokens when playing the token game.

Definition 3.1.

Given a net N = <B, E; F> and a case c,

- an event e is **a-enabled** by c if and only if ${}^{\bullet}e \subseteq c$ and $e^{\bullet} \subseteq {}^{\bullet}e \cup B\text{-}c$.

Given $c_1, c_2 \subseteq B$ and $G \subseteq E$, the a-step c_1 [[G> c_2, is defined if

- $\forall e \in G$, e is a-enabled by c_1;
- $\forall e_1, e_2 \in G$, $e_1 \neq e_2$, $\cdot e_1 \cap \cdot e_2 = e_1\cdot \cap e_2\cdot = \Phi$;
- $c_2 = (c_1 - \cdot G) \cup G\cdot$. ♦

Definition 3.2.

An Augmented Condition/Event system (A-C/E system) is a quadruple Σ = <B, E; F, C>, where

- <B, E; F> is a net with no isolated element and $B \cup E \neq \Phi$;
- $C \subseteq 2^B$ is an equivalence class of the reachability relation $R=(r \cup r^{-1})^*$, being $r \subseteq 2^B \times 2^B$, where c_1 r c_2 iff $\exists G \subseteq E$ such that c_1 [[G> c_2;
- $\forall e \in E$, $\exists c$ such that e is a-enabled by c. ♦

Definition 3.3.

An a-contact-free A-C/E system is an A-C/E system <B, E; F, C> in which $\forall e \in E$, $\forall c \in C$

- $\cdot e \subseteq c$ implies $e\cdot \subseteq \cdot e \cup B - c$;
- $e\cdot \subseteq c$ implies $\cdot e \subseteq e\cdot \cup B - c$. ♦

Property 3.1.

Given an a-contact-free A-C/E system <B, E; F, C> and a case c in C, an event is a-enabled iff its preset is in c.

The following theorem characterizes a-contact-freeness.

Definition 3.4.

A case c of a net N = <B, E; F> is **critical** if there exists an event e such that either

$\cdot e \subseteq c$ and $(c - \cdot e) \cap e\cdot \neq \Phi$ or $e\cdot \subseteq c$ and $(c - e\cdot) \cap \cdot e \neq \Phi$ ♦

Note that in a critical case the preset of an event holds, which, if fired, would cause the "loss of some token". Actually, the case reached after firing would be the union of $c - \cdot e$ and $e\cdot$ and, if the two sets were intersecting, some token would be lost. Alternatively, the dual property might hold.

Theorem 3.1. (*a system is a-contact-free iff no token is lost in the token game*)

An A-C/E system $\Sigma = \langle B, E; F, C\rangle$ is a-contact-free iff no case in C is critical.

Proof. Immediate. ♦

The if part of the above theorem is not true for standard C/E Systems, as proved by the C/E system in Fig. 3.1. which, being non contact-free, has no critical cases.

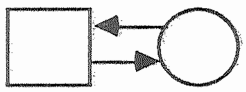

Fig. 3.1. A non contact-free C/E system which is an a-contact-free A-C/E system.

The following theorem relates C/E Systems with A-C/E Systems.

Theorem 3.2.

The set of (contact-free) C/E Systems is a proper subset of (a-contact-free) A-C/E Systems.

Proof. Containement is obvious. For proper inclusion see the net in Fig. 3.1. ♦

4. CCS as an Augmented C/E System

In this section we present a new set of inference rules for CCS which are a simpler version of those presented in [2, 3, 4], and which relate parts of CCS agents, rather than whole global states. CCS agents are decomposed into sets of sequential processes, called grapes, and the new transition relation not only tells the actions an agent E may perform, but also tells those sequential processes of E which move when a transition occurs. The new transition relation will be used as the starting point for defining the new semantics of CCS in terms of an A-C/E System. The transitions have the form $I_1 -\mu\rightarrow I_2$ and their intuitive meaning is that the set of grapes I_1 may become the set I_2 by performing the action μ. The new axioms and inference rules are in direct correspondence with those of Section 2. Before introducing the new transition relation we need few definitions.

Definition 4.1. (*defining CCS sequential processes*)

A grape is a term defined by the following BNF-like grammar

$$G ::= E \mid id|G \mid G|id \mid G\backslash\alpha \mid G[\phi]$$

where E, $\backslash\alpha$ and $[\phi]$ have the standard CCS meaning. ♦

Intuitively speaking, a grape represents a subagent of a CCS agent, together with its access path. A CCS agent can be decomposed by function dec into a set of grapes.

Definition 4.2. (*decomposing CCS agents into sequential processes*)

The function dec decomposes a CCS agent into a set of grapes and is defined by structural induction as follows:

$$\mathrm{dec}(x) = \{x\}$$

$$\mathrm{dec}(NIL) = \{NIL\}$$

$$\mathrm{dec}(\mu E) = \{\mu E\}$$

$$\mathrm{dec}(E\backslash\alpha) = \mathrm{dec}(E)\backslash\alpha$$

$$\mathrm{dec}(E[\phi]) = \mathrm{dec}(E)[\phi]$$

$$\mathrm{dec}(E_1+E_2) = \{E_1+E_2\}$$

$$\mathrm{dec}(E_1|E_2) = \mathrm{dec}(E_1)|id \cup id|\mathrm{dec}(E_2)$$

$$\mathrm{dec}(rec\ x.\ E) = \{rec\ x.\ E\}.$$ ♦

We understand constructors as extended to operate on sets, e.g. $I\backslash\alpha=\{g\backslash\alpha \mid g\in I\}$. Note that the decomposition stops when an action, a sum or a recursion is encountered, since these are considered as atomic sequential processes.

Example 4.1.

$\mathrm{dec}((((recx.\ \alpha x+\beta x) \mid recx.\ \alpha x+\gamma x) \mid recx.\ \bar{\alpha} x)\backslash\alpha) =$

$\{(((recx.\ \alpha x+\beta x)|id)|id)\backslash\alpha,\ ((id|recx.\ \alpha x+\gamma x)|id)\backslash\alpha,\ (id|recx.\ \bar{\alpha} x)\backslash\alpha\}$ ♦

Definition 4.3.

A set I of grapes is **complete** if there exists a CCS agent E such that dec(E) = I. ◆

Property 4.1. *(complete sets of grapes are isomorphic to CCS agents)*

Function dec is injective and thus defines a bijection between CCS agents and complete sets of grapes.

Proof. Immediate by structural induction. ◆

Note that the inverse function of dec is standard unification, provided that distinct variables are substituted for each occurrence of id, and $\{\mu E\}$, $\{E_1+E_2\}$ and {rec x. E} are considered atomic. In other words, the most general unifier of a complete set of grapes I is the CCS agent of which I is the decomposition. Note also that complete sets of grapes can be used to represent the global states of the system.

Definition 4.4. *(partial ordering derivation relation)*

The **partial ordering derivation relation** $I_1 \text{—}\mu\text{→} I_2$ is defined as the least relation satisfying the following axiom and inference rules

act) $\{\mu E\} \text{—}\mu\text{→} \mathrm{dec}(E)$

res) $I_1 \text{—}\mu\text{→} I_2$ implies $I_1\backslash\alpha \text{—}\mu\text{→} I_2\backslash\alpha$, $\mu \notin \{\alpha, \alpha^-\}$

rel) $I_1 \text{—}\mu\text{→} I_2$ implies $I_1[\phi] \text{—}\phi(\mu)\text{→} I_2[\phi]$

sum) $(\mathrm{dec}(E_1) - I_3) \text{—}\mu\text{→} I_2$ implies $\{E_1+E\} \text{—}\mu\text{→} I_2 \cup I_3$ and $\{E+E_1\} \text{—}\mu\text{→} I_2 \cup I_3$

com) $I_1 \text{—}\mu\text{→} I_2$ implies $I_1|\mathrm{id} \text{—}\mu\text{→} I_2|\mathrm{id}$ and $\mathrm{id}|I_1 \text{—}\mu\text{→} \mathrm{id}|I_2$

$I_1 \text{—}\lambda\text{→} I_2$ and $I'_1 \text{—}\lambda^-\text{→} I'_2$ implies $I_1|\mathrm{id} \cup \mathrm{id}|I'_1 \text{—}\tau\text{→} I_2|\mathrm{id} \cup \mathrm{id}|I'_2$

rec) $(\mathrm{dec}(E_1[\mathrm{rec}\ x.\ E_1/x]) - I_3) \text{—}\mu\text{→} I_2$ implies $\{\mathrm{rec}\ x.\ E_1\} \text{—}\mu\text{→} I_2 \cup I_3$. ◆

We can now comment about our axiom and rules. In axiom act), a single grape is rewritten as a set of grapes, since the firing of the action makes explicit the (possible) parallelism of E. The rules

res) and **rel)** and the first two rules for **com)** simply say that if a set of grapes I_1 can be rewritten as I_2 via μ, then we can prefix the access paths of the grapes in both I_1 and I_2 with any of the path constructors $.\backslash\alpha$, $.[\phi]$, .lid and idl., and still obtain a derivation, labelled say by μ'. Clearly, when dealing with restriction $\mu' = \mu$, but the inference is possible only if $\mu \notin \{\alpha,\alpha^-\}$; in **rel)** $\mu' = \phi(\mu)$ and in the first two rules of **com)** $\mu' = \mu$. The second rule for **com)** is just the synchronization rule. A move generated by the rule **sum)** can be understood as consisting of two steps. Starting from the singleton $\{E_1+E\}$ or $\{E+E_1\}$ a first step discards alternative E and decomposes E_1 into dec(E_1) and a second step (the condition of the inference rule) rewrites a subset of dec(E_1) as I_2, while the rest of dec(E_1), say I_3, is rewritten unchanged. The net effect of the two steps, however, is to rewrite the singleton $\{E_1+E\}$ or $\{E+E_1\}$ into the set $I_2 \cup I_3$. The rule for **rec)** determines the moves of a recursively defined agent via the transitions of the set of grapes obtained by decomposing the result of an unwinding of the recursive definition: if a subset of dec(E_1[rec x. E_1/x]) may be rewritten via an action as I_2, then {rec x. E_1} may be rewritten via the same action as $I_2 \cup I_3$, where I_3 is the subset of dec(E_1[rec x. E_1/x]) which does not contribute to the action.

Theorem 4.1. (*asynchrony of partial ordering derivation relation*)
If $I_1 —\mu\rightarrow I_2$ is a derivation, then there exists a set of grapes I_3, with $I_3 \cap I_1 = \Phi$, such that $I_1 \cup I_3$ is a complete set of grapes. Furthermore, for every such I_3 we have also that $I_3 \cap I_2 = \Phi$ and that $I_2 \cup I_3$ is a complete set of grapes.
Proof. Immediate by induction on the structure of the proof of the derivation. ♦

The theorem above amounts to saying that the initial set of grapes of a derivation can always be seen as a part of a global state. Moreover, derivations are independent from grapes which are concurrent with the rewritten ones, but inactive. This result throws light on the asynchrony of our derivation relation.

Theorem 4.2. (*correspondence between Milner's and partial ordering derivations*)
We have a Milner's derivation $E_1 —\mu\rightarrow E_2$ iff there exist a set of grapes I_3 and a triple $I_1 —\mu\rightarrow I_2$ in the partial ordering derivation relation such that
dec(E_1) = $I_1 \cup I_3$ and dec(E_2) = $I_2 \cup I_3$.

Proof. Given a derivation, use the structure of its deduction to obtain the other derivation. ♦

We can now define the A-C/E System corresponding to CCS.

Definition 4.5. *(from CCS to the corresponding a-contact-free A-C/E system)*

Let Σ_{CCS} = <B, E; F, C>, where

- B is the set of all grapes;
- E is the set of the triples $I_1 —\mu\rightarrow I_2$ in the partial ordering derivation relation;
- I_1 F ($I_1 —\mu\rightarrow I_2$) and ($I_1 —\mu\rightarrow I_2$) F I_2, for all $I_1 —\mu\rightarrow I_2$ in E;
- C is the class of all complete sets of grapes. ♦

Property 4.2.

C is an equivalence class of the reachability relation.

Proof. Given two cases dec(E_1) and dec(E_2), we have dec(E_1) r^{-1} dec($\alpha E_1 + \beta E_2$) r dec(E_2). ♦

Property 4.3.

The A-C/E system Σ_{CCS} is a-contact-free.

Proof. By Theorem 4.1., if $I_3 \cup I_1$ is complete then $I_3 \cap I_2$ is empty; the claim follows trivially. ♦

Example 4.2. Let us consider the CCS agent of Example 4.1.

E = (((recx. αx+βx) | recx. αx+γx) | recx. $\bar{\alpha}$x)\α.

Fig. 4.1 shows the sub-system of Σ_{CCS} corresponding to E, containing only the cases c with dec(E) r* c, the conditions in such cases and the events enabled by them. It has one case

$c_0 = \{b_0, b_1, b_2\}$, where

b_0 = (((recx.αx+βx)|id)|id)\α;

b_1 = ((id|recx. αx+γx)|id)\α;

b_2 = (id|recx. $\bar{\alpha}$x)\α;

the following four transitions

$e_0 : \{b_0, b_2\} —\tau\rightarrow \{b_0, b_2\}$;

$e_1 : \{b_0\} —\beta\rightarrow \{b_0\}$;

$e_2 : \{b_1\} —\gamma\rightarrow \{b_1\}$;

$e_3 : \{b_1, b_2\} —\tau\rightarrow \{b_1, b_2\}$;

and the relation F such that

$^\bullet e_0 = e_0{}^\bullet = \{b_0, b_2\};$ $^\bullet e_1 = e_1{}^\bullet = \{b_0\};$

$^\bullet e_2 = e_2{}^\bullet = \{b_1\};$ $^\bullet e_3 = e_3{}^\bullet = \{b_1, b_2\}.$ ♦

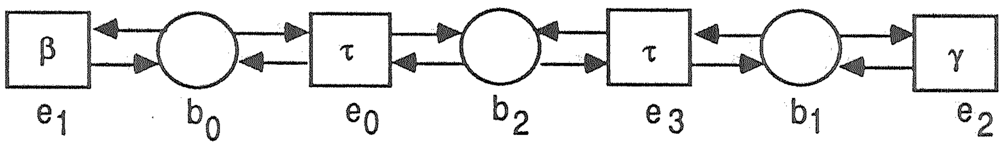

Fig. 4.1. The fragment of Σ_{CCS}, as constructed in Example 4.2., which corresponds to the agent E of Example 4.1. For clarity sake, we have labelled the events with the actions they contain.

We now show that the interleaving semantics of Σ_{CCS} coincides with the standard semantics of CCS.

Definition 4.6. (*interleaving case graph*)

Let $P_{abstr} = \{(c_1, \mu, c_2) \mid c_1 \,[[\, I_1 \!\!-\!\!\mu\!\!\rightarrow\! I_2 > c_2$ is a step of $\Sigma_{CCS}\}$.

Then, the graph $\Phi = (C, P_{abstr})$ is the **interleaving** case graph of the A-C/E system Σ_{CCS}. ♦

Note that passing from Σ_{CCS} to Φ corresponds to an abstraction step, since more than one event is mapped on the same arc. For instance, the two events e_0 and e_3, both labelled by τ, are mapped into the single arc labelled by τ of the graph in Fig. 4.2.a).

Definition 4.7. (*CCS transition graph*)

Let E be the set of CCS agents and T be Milner's derivation relation.

Then, the graph $\Psi = (E, T)$ is the **transition graph** of CCS. ♦

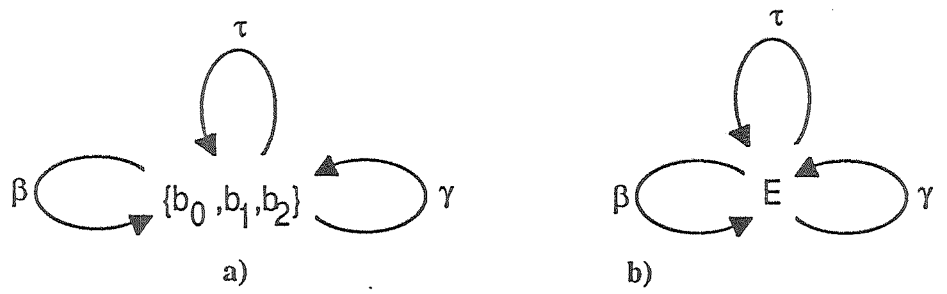

Fig.4.2. The part relative to the agent E of Example 4.2. of the interleaving case graph of Σ_{CCS} (in a) and of the transition graph of CCS (in b) .

Theorem 4.3. (*interleaving Σ_{CCS} is CCS*)

The interleaving case graph Φ of Σ_{CCS} is isomorphic to the transition graph Ψ of CCS.

Proof. The one-to-one correspondence between the nodes of Φ and Ψ follows from Property 4.1., so we will write hereto c_i for dec(E_i) and viceversa.

If there is an arc (c_1, μ, c_2) in Φ, there exist then both a set of grapes I_3 and a triple I_1—μ→I_2 in the partial ordering derivation relation such that $I_1 \cup I_3 = c_1$ and $I_2 \cup I_3 = c_2$. No matter how we choose I_1, I_2 and I_3, by Theorem 4.2. there exists a unique E_1—μ→E_2 in Milner's derivation relation, hence the arc (E_1, μ, E_2) in Ψ corresponds to the arc (c_1, μ, c_2) in Φ. The proof of the converse is similar. ♦

Corollary 4.1.

The trees obtained by unfolding the interleaving case graph of Σ_{CCS} are isomorphic to the synchronization trees of CCS. ♦

The above construction is suggestive and straightforward, but violates the simplicity requirement for C/E Systems. Simplicity is felt as an important property within the theory of Petri Nets, being imposed by the extensionality requirement. The following construction, slightly more cumbersome, generates a system which is simple.

Definition 4.8. (*a simple a-contact-free A-C/E system for CCS*)

Let S_{CCS} = <B, E; F, C>, where

- $B = G \cup \Lambda$, where G is the set of all grapes;
- E is the set of the triples I_1—μ→I_2 in the partial ordering derivation relation;
- ($I_1 \cup \{\mu\}$) F (I_1—μ→I_2) and (I_1—μ→I_2) F ($I_2 \cup \{\mu\}$), for all I_1—μ→I_2 in E;
- $C = \{I \cup \Lambda \mid I$ is a complete set of grapes$\}$. ♦

Property 4.4.

The A-C/E system S_{CCS} is simple and a-contact-free.

Fig. 4.3. shows the fragment of S_{CCS} corresponding to the Examples 4.1. and 4.2. (omitting the conditions for the actions not included in the sort of agent E).

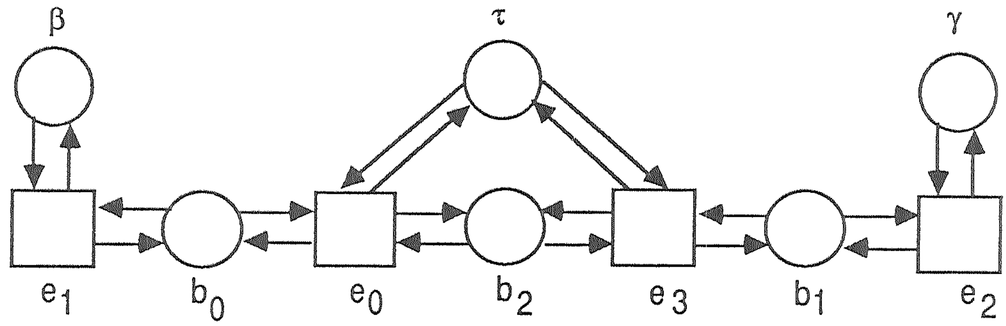

Fig. 4.3. The fragment of S_{CCS} corresponding to the Examples 4.1. and 4.2. .

It is immediate to extend Theorem 4.3. and Corollary 4.1. to the simple and a-contact-free A-C/E system S_{CCS}.

5. Conclusions

We have introduced Augmented Condition/Event Systems by slightly relaxing the classical simplicity and enabling conditions for Condition/Event Systems. A Net from this class, called Σ_{CCS}, is used to give a new operational semantics to Milner's Calculus for Communicating Systems. Indeed, full CCS has been mapped on the a-contact-free Augmented Condition/Event system Σ_{CCS} and also on the simple a-contact-free Augmented Condition/Event system S_{CCS}. The transition system of CCS has been proved isomorphic to the interleaving case graphs of Σ_{CCS} and S_{CCS}.

We argue that our extensions do not affect the results proven for standard Condition/Event Systems, thus non-sequential processes defined for Condition/Event Systems [9] can be immediately defined also for CCS.

The translation is in one direction only, mainly because synchronization in CCS involves only two agents, whereas in Petri Nets the preconditions of an event may even be an infinite set. CCS synchronization algebra needs extensions in order to cope with this problem.

The authors have tackled the problem of partial ordering semantics in a number of papers; particularly relevant to the work presented here are papers [2, 3, 4]. In [2] a partial ordering semantics for CCS is given in terms of Concurrent Histories, a sequential rewriting system previously developed by two of the authors. In [4], a new model for non-sequential computations, called Distributed Transition Systems, in which states are sets of processes and transitions specify which processes stay idle, is introduced and both Condition/Event and Place/Transition Petri Nets and CCS are modelled in terms of DTS. The model for CCS developed in [4] is used in [3] as the basis for defining new equivalence relations over the set of CCS agents and for studying the relationships between interleaving and partial ordering observational semantics of the language.

In this paper we let Condition/Event Systems play the role of transition systems and are thus able to contrast Petri Nets and CCS directly. As in the other papers, the technique used to define CCS agents transitions is based on Plotkin's SOS. However, the derivation rules are simpler. Indeed, there are just two sets of processes and no explicit reference to idle processes is made.

In the introduction we discussed other attempts to give a semantics for CCS in terms of Petri Nets. The most complete of these attempts is that of [6], which, in denotational style, defines an operation on nets in correspondence with every CCS operator and associates a net to every CCS agent. Unfortunately, as already mentioned, the resulting semantics is not in full agreement with CCS original interleaving semantics. In particular, the new semantics does not respect all the original causal constraints between the actions performed by CCS agents. Indeed, there exist agents whose corresponding Occurrence Nets permit the firing of an action before its causes have fired. For example, consider the following agent:

$$E = (\beta\delta\gamma\, NIL \mid \alpha\bar{\delta}\, NIL)\backslash\delta \mid \bar{\alpha}\, NIL$$

According to the original interleaving based semantics of [7], we have that E can perform an action γ only after it has performed an action β; now the corresponding net according to [6], reported in Fig. 5.1., allows a γ to be performed after two invisible actions regardless of β.

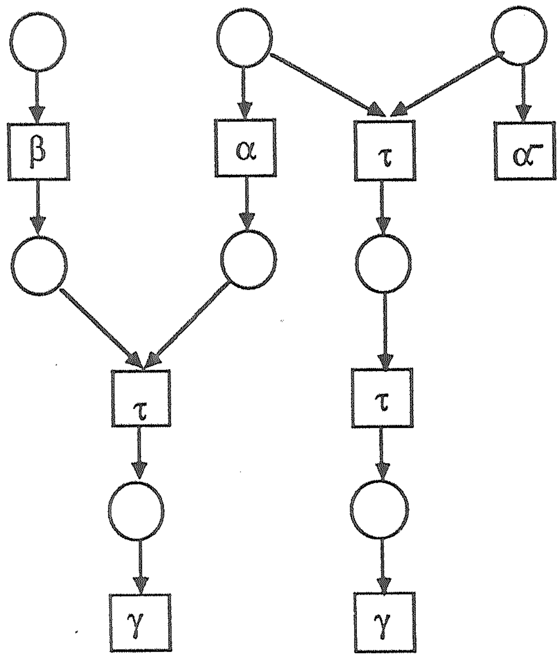

Figure 5.1. The occurrence net associated to $(\beta\delta\gamma\,NIL|\,\alpha\delta^{-}NIL\,)\backslash\delta\,|\,\alpha^{-}NIL$ by Goltz and Mycroft

The main reason for the loss of causal dependencies seems to be the way they describe the effect of synchronizing two single transitions in a net, see Definition (6) of Section 4.1 in [6]. Whenever a net contains two complementary actions , say α and α^{-}, they add to it newly generated copies of the subnets "behind" α and α^{-} together with a new transition labelled by τ whose preconditions are the preconditions of α and α^{-} and whose postconditions are the initial conditions of the new copies. In this way, the relationships between the transitions of the new copies of the subnets "behind" α and α^{-} and the transitions of the original nets which are not related to α or $\alpha^{-}$are lost.

Acknowledgements.

The construction of the simple net for CCS come out after pleasant and friendly discussions with Pippo Torrigiani. Tommaso Bolognesi pointed out inadequacies of the definition of the parallel operator in [6].

References

1. De Cindio, F., De Michelis, G., Pomello, L. and Simone C.: Milner's Communicating Systems and Petri Nets. In: *Selected papers from the 3^{rd} European Workshop on Applications and Theory of Petri Nets*, Informatik Fachberichte 66, pp. 40-59, Springer-Verlag, 1983.

2. Degano, P., De Nicola, R. and Montanari, U.: Partial Ordering Derivations for CCS. Proc.5^{th} Int. Conf. on Fundamentals of Computation Theory (L. Budach, ed.), LNCS 199, pp. 520-523, Springer-Verlag, 1985.

3. Degano, P., De Nicola, R. and Montanari, U.: Observational Equivalences for Concurrency Models. To appear in Proc. IFIP Workshop on Formal Description of Programming Concepts, (M. Wirsing, ed.), North Holland, 1986

4. Degano, P. and Montanari, U.: Concurrent Histories: A Basis for Observing Distributed Systems. To appear in *Journal of Computer and System Sciences.*

5. Genrich, H.J., Lautenbach, K., and Thiagarajan, P.S.: Elements of General Net Theory. In: *Net Theory and Applications* (W. Brauer, ed.) LNCS 84, pp. 21-163, Springer-Verlag, 1980.

6. Goltz, U. and Mycroft, A.: On the Relationships of CCS and Petri Nets. Proc. 11^{th} ICALP (J. Paredaens, ed.), LNCS 172, pp. 196-208, Springer-Verlag,1984.

7. Milner, R.: *A Calculus of Communicating Systems.* LNCS 92, Springer-Verlag, 1980.

8. Plotkin, G.: A Structural Approach to Operational Semantics. Lecture Notes, Aarhus University, Department of Computer Science, 1981.

9. Reisig, W.: *Petri Nets: An Introduction*, EACTS Monographs on Theoretical Computer Science, Springer-Verlag, 1985.

10. Winskel, G.: Event Structure Semantics for CCS and Related Languages, Proc. 9^{th} ICALP (M. Nielsen and E. M. Schmidt, eds.), LNCS 140, pp. 561-567, Springer-Verlag, 1982.

11. Winskel, G.: A New Definition of Morphism on Petri Nets, Proc. 1^{st} Symposium on Theoretical Aspects of Computer Science, (M. Fontet and K. Melhorn, eds.), LNCS 166, pp. 140-150, Springer-Verlag, 1984.